LOCUS

COMPUTING

CORPORATION

PC-Interface User's Guide

PC-Interface User's Guide

GN00101-01.REV00
June 1989

Release 3.0

Locus Computing Corporation

GN00101-01.REV00, Initial Printing, June 1989

Locus Computing Corporation, 9800 La Cienega Blvd, Inglewood CA 90301-4440

Printed in U.S.A.

About This Book

This manual explains how to use your IBM or IBM-compatible personal computer with a network interface board or an asynchronous communications adapter and the PC-Interface software.

The PC-Interface software allows you to:

- Integrate your personal computer into an existing UNIX® host computer network.
- Run DOS applications using the file services and printers of one or more UNIX hosts as if the host were a local fixed disk.
- Execute UNIX commands from DOS on your personal computer.
- Conduct a standard UNIX session using your personal computer as a terminal.

Throughout this manual, the term "personal computer" refers to the IBM Personal Computer, the IBM Personal Computer XT, the IBM Personal Computer AT, the IBM PS/2, or other compatible computer. The term "DOS" refers to the Microsoft Disk Operating System program (version 3.10 or later) or the IBM Disk Operating System program (version 3.10 or later). The term "UNIX" refers to the UNIX System V operating system or other compatible operating system with TCP/IP networking service available through either a BSD socket interface or System V TLI. The term "LAN" means local area network and refers to a connection path from your personal computer to a host via Ethernet.

Helpful Features

A reference card is provided with this manual, summarizing the PC-Interface commands and the terminal emulation keyboard uses.

Who Should Use This Manual

This manual is for anyone using PC-Interface with a personal computer to take advantage of information or services that exist in a UNIX host environment.

You should have a general knowledge of MS-DOS, Version 3.10 or later. If you plan to use the PC-Interface terminal emulation feature, you should also be familiar with the UNIX system used at your site.

For information about the structure and administration of PC-Interface on the UNIX host, refer to the *PC-Interface Administrator's Guide*. For instructions on installing PC-Interface on the host, see the *PC-Interface Installation Guide*. Note that Appendix A of this manual contains instructions for installing PC-Interface on your personal computer.

Organization Of This Manual

This manual has six chapters and four appendices as follows:

Chapter 1. Introduction acquaints you with the PC-Interface software and the configuration requirements for your personal computer.

Chapter 2. Using Host File Services tells you how to establish communications with your UNIX host and how to run DOS applications using the host as a large fixed disk.

Chapter 3. Using UNIX From DOS describes how to execute UNIX commands from your personal computer running DOS.

Chapter 4. Using Your Computer as a UNIX Terminal describes the terminal emulation features of PC-Interface.

Chapter 5. Tailoring PC-Interface gives you some tips and ideas for modifying your PC-Interface installation to better suit the way you use both the PC-Interface program and DOS applications over PC-Interface.

Chapter 6. Command Summary provides a quick reference for all the PC-Interface commands discussed elsewhere in this manual.

Appendix A. Installing PC-Interface describes installing an asynchronous communications adapter or a network interface board in your personal computer, and gives step-by-step instructions on how to combine the PC-Interface software with DOS to create a working version of PC-Interface.

Appendix B. Messages explains all the messages you might receive from PC-Interface.

Appendix C. Using a Dial-Up Modem tells you how to establish a host connection using a dial-up modem.

Appendix D. Tips on Using DOS Applications offers hints on running DOS applications and commands that behave differently in the PC-Interface environment than they do in a pure DOS environment.

Related Publications

1. Other PC-Interface publications:
   - *PC-Interface Administrator's Guide*
   - *PC-Interface Host Installation Guide*
2. Personal computer hardware publications:
   - Personal Computer *Guide to Operations*, IBM Corporation.
   - Personal Computer *Quick Reference*, IBM Corporation.
   - Personal System/2 *Guide to Operations*, IBM Corporation.
   - Personal System/2 *Quick Reference*, IBM Corporation.
3. Personal computer software publications:
   - Personal Computer *Disk Operating System*, IBM Corporation.
   - *DOS Technical Reference*, IBM Corporation.

Contents

Chapter 1. Introduction

Chapter 2. Using Host File Services

Chapter 4. Using Your Computer As A UNIX Terminal

Chapter 5. Tailoring PC-Interface

Chapter 6. Command Summary

Chapter 1. Introduction

PC-Interface

The PC-Interface software package allows you to bridge the gap between your personal computer and UNIX hosts. With PC-Interface and LAN or asynchronous connections to UNIX hosts, you can run DOS applications using data from the UNIX file system. You can store files and even DOS applications on a UNIX host and access them just as if they were on a local fixed disk. By taking advantage of the increased storage capacity of the UNIX file system, your personal computer can operate as though it had multiple fixed disks.

When you use the host file services of PC-Interface, the file security capabilities of UNIX allow 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.

PC-Interface can also emulate a VT220 terminal, allowing you to conduct a regular UNIX session from your personal computer.

For all its power, PC-Interface is remarkably simple to use. DOS users need to know nothing about UNIX to use host file services. Just treat the UNIX system as an extra disk drive connected directly to your personal computer. Users familiar with UNIX in addition to DOS can run UNIX commands from DOS. These users can also 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 mail message.

Minimum Requirements

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

Personal Computer

- An IBM personal computer (PC, XT, or AT), IBM PS/2, or other compatible personal computer.
- 256K bytes of memory.
- One 360K-byte (double-sided) diskette drive or one 720K-byte microdiskette drive.
- A monochrome display with a monochrome display adapter or a color display with a color/graphics monitor adapter.
- PC-DOS or MS-DOS, Version 3.10 or later.
- An asynchronous communications adapter or a network interface board. See the *Release Notes* accompanying this manual for a list of network interface boards supported by your release of PC-Interface.

Host

Your personal computer must be attached via a LAN or RS-232 connection to a host system running the UNIX System V operating system or other compatible operating system. The

UNIX modules for PC-Interface must already be installed on this host.

Installing The Adapter

Install your network interface board or asynchronous communications adapter according to the manufacturer's instructions. Check Appendix A and the *Release Notes* accompanying this manual for information about configuring jumper settings.

Installing PC-Interface

The pocket at the back of this book contains your PC-Interface distribution diskettes. To use PC-Interface, you must either create a working diskette that combines the PC-Interface programs with the DOS files needed to "boot" your personal computer, or else install PC-Interface on a fixed-disk drive so that it is automatically initialized when you boot your personal computer.

The PC-Interface distribution diskette contains an installation program called INSTALL that simplifies the installation process. It allows you to install either the minimum "basic" system or the complete PC-Interface system. Before reading further, you should follow the instructions in Appendix A, Installing PC-Interface.

Chapters 2 through 4 of this manual assume that you have installed PC-Interface according to the instructions in Appendix A.

Copy Protection

The PC-Interface software is copy-protected. Although you may make copies for your personal use, you cannot use two copies at the same time. When the UNIX host detects two simultaneous sessions from the same distribution diskette, it disables the session on the personal computer of the user who logged in last. This may happen some time after the second user has logged in and begun working.

Compatibility

Most popular DOS application software products operate with PC-Interface, including the following:

- BASICA, Version 3.3
- dBase III Plus LAN PACK
- Desqview, Version 2.0
- Framework II
- Framework III
- Lotus 1-2-3
- MS C, Version 5.1
- MS Windows
- MS Word
- PC Excel
- Sidekick
- Word Perfect
- Wordstar

Other DOS application software products should operate with PC-Interface if they do not:

- Program the 8259 interrupt controller in a way that interferes with PC-Interface's use of that controller.

- 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 disk.

- 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.

Chapter 2. Using Host File Services

Introduction

PC-Interface lets you interact with a UNIX host in the following three ways:

- Your personal computer can use the file services of the host while running DOS applications as if the host were a colocated fixed disk.

- Your personal computer can use the PC-Interface ON utilities to access the processing capabilities of the UNIX host directly from the DOS environment.

- Your personal computer can emulate a VT220 terminal, enabling you to conduct a standard UNIX host session.

You can work entirely in one mode or the other or you can switch back and forth between them.

This chapter tells you how to use host file services, including:

- Beginning and ending a host file services session.

- Handling differences between DOS and UNIX file-naming conventions.

- Converting text files between DOS and UNIX formats.

- Using UNIX file permission modes to protect your files in a multiuser environment.

- Executing UNIX processes from DOS.

- Printing files on either a remote or local printer.

Chapter 3 describes using the ON utilities.

Chapter 4 describes using terminal emulation.

The procedures described throughout this manual and the examples shown assume that PC-Interface has been installed for your personal computer as described in Appendix A.

Beginning A File Services Session

The following procedure starts a file services session. This allows you to run DOS applications, using and storing files on the host as if it were a local fixed disk.

1. If you are using a diskette, insert your PC-Interface working diskette in drive A: and close the drive door. If you are using a fixed disk, leave the drive A: door open.

2. Switch ON your personal computer, if you have not already done so.

   A series of messages display, including the following:

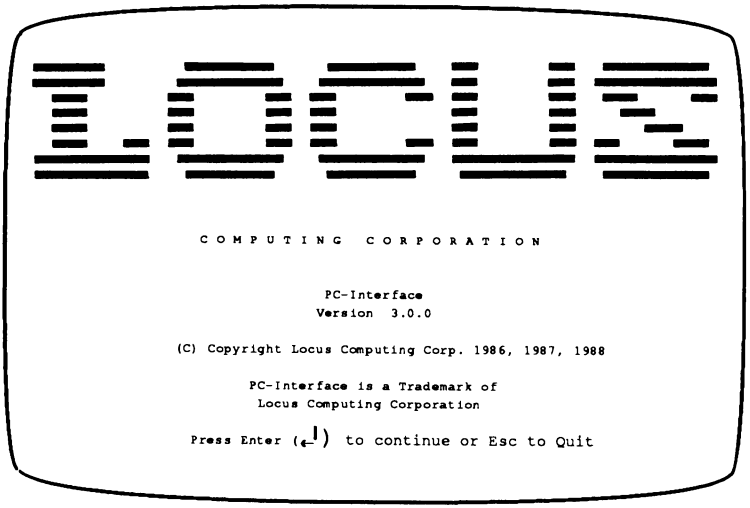
   ```
   PC-Interface Release 3.0 (Serial #x.xxxx) Initialized
   ```

   Following this message, you may see the commands from the AUTOEXEC.BAT file as they are executed.

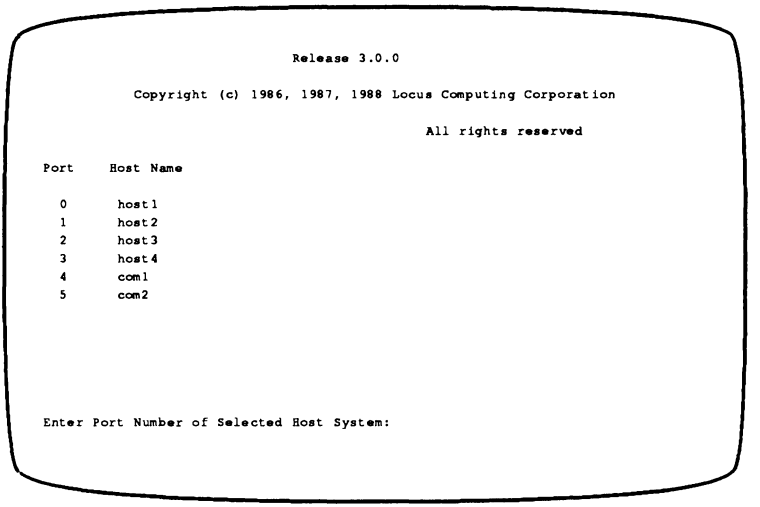
3. In response to the system prompt (A> if you are using a diskette; C> if you are using a fixed disk), type:

```
login
```

Press Enter (↵). The PC-Interface logo screen appears:



4. Press Enter (↵) to continue. The PC-Interface host menu screen appears:



The PC-Interface host menu screen shows you the UNIX host connection paths available to you. When your personal computer is connected to one or more hosts over a LAN, the host names appear on the menu. In addition, PC-Interface supports host connections through one or two serial ports. When your personal computer is equipped with serial ports, the words **com1** or **com2** appear on the menu depending on which serial ports are available.

You may have one or more choices on your menu screen. Depending on the size of your network, there may be more hosts than can be displayed on the screen. In that case, you can establish a session with a host that is not shown (as long as the host is part of your network) by using the command-line login procedure. See "Logging In From The Command Line," later in this chapter.

Logging In Over A LAN Connection Path

Use the following procedure to choose a UNIX host connection over a LAN:

1. In response to the host-selection prompt, type the port number that appears on the menu next to the desired host name and press Enter (↵).[1]

   A connection-attempt message briefly replaces the host-selection prompt, and is replaced by the following prompt:

   ```
   Username:
   ```

2. Type your UNIX user name and press Enter (↵).

   PC-Interface prompts you for your password as follows:

   ```
   Password:
   ```

3. Type your UNIX user password and press Enter (↵). The characters you type do not appear on the screen, thus protecting the security of your password.

   PC-Interface attempts to establish a connection to the UNIX host you selected. If your user name or password was entered incorrectly and PC-Interface cannot establish a connection, it displays the following message on your screen:

   ```
   Login incorrect
   Try Again? (y or n):
   ```

[1] PC-Interface continues to prompt you until you enter a valid port number. If the host you want is not available and you want to return to DOS, press and hold the CTRL key and type C.

When you type y, PC-Interface reprompts you for your user name and password. When you type n, PC-Interface returns you to DOS, and you receive the DOS operating system prompt. (**Note**: Do not press Enter after the y or n).

4. When the connection is established, the following appears on your screen:

```
Welcome to PC-Interface.
hostname File System Available as Drive D:
C>
```

The word *hostname* is replaced by the name of the host to which you have connected. Note that the drive specifiers you see might be different. PC-Interface associates the next available drive specifier supplied by DOS with the UNIX file system, treating it as a "virtual" drive. If the highest drive letter used by your personal computer is C:, the drive specifier for the first UNIX file system is D:.

Logging In Over An RS-232 Connection Path

Use the following procedure to choose a UNIX host connection over an RS-232 connection path:

1. In response to the host-selection prompt, type the port number that appears on the menu next to the desired com port and press Enter (↵).

The following prompt replaces the host-selection prompt:

```
Enter Baud Rate (300,1200,2400,4800,9600):
```

2. Type the baud rate you want to use for transmission to the host and press Enter (↵).[1]

   PC-Interface prompts you to log in to your UNIX host. A five-line window appears at the bottom of your screen in which the UNIX host identifies itself and prompts you for your user name and password.

   If the window remains blank, try pressing Enter again. If the screen is still blank, the automatic baud rate of the host may not match the baud rate you selected. To change the host baud rate, press the F1 function key repeatedly until the host identification appears.

   When you have successfully logged in, you receive the UNIX prompt.

3. Press the F9 key.

   The window disappears and the following is displayed:

   ```
   Connection Attempt in Progress
   ```

4. When the connection is established, the following appears on your screen:

   ```
   Welcome to PC-Interface.
   hostname File System Available as Drive D:
   C>
   ```

The word *hostname* is replaced by the name of the host to which you have connected. Note that the drive specifiers you

[1] The most efficient baud rate for transmitting data to the UNIX host varies depending on your particular system configuration. In some networks, the fastest speed, 9600 baud, might cause errors in the host and result in performance degradation because of error recovery. If you are uncertain about which baud rate to use, consult your system administrator.

see might be different. PC-Interface associates the next available drive specifier supplied by DOS with the virtual drive. If the highest drive letter used by your personal computer is C:, the drive specifier for the first UNIX file system is D:.

Logging In To Additional Hosts

You can establish up to four virtual drives, each associated with a different host and each assigned a different drive letter.[1] For example, if your last local drive is the C: drive, you can have virtual drives from D: to G:.[2]

To log in to additional hosts and establish additional virtual drives, repeat the appropriate login procedure either specifying a new virtual drive letter or letting PC-Interface assign one for you. If you do not specify a virtual drive letter when you log in, the system searches for the next available letter and assigns it to the virtual drive.

[1]You can, in fact, configure PC-Interface to support a maximum of 16 virtual drives. For more information, see "Changing The Maximum Number Of Virtual Drives" in Chapter 5.

[2]These drive letters are reserved for virtual drives and should not be used with the SUBST command. If you want to use a drive letter with the SUBST command, use one that is higher than the highest drive letter for which your system is configured.

If you wish to specify a particular letter for a virtual drive rather than letting PC-Interface assign one for you, use the /D: option of the LOGIN command. For example:

```
login /d:g
```

establishes drive G: as the next virtual drive.

Note that you can only log in once to any given host. If you try to log in a second time to the same host, PC-Interface displays the following message:

```
login: hostname already logged in
```

You may continue to log in to different hosts up to the maximum you have configured for your system.

Logging In From The Command Line

To save time, you can bypass the PC-Interface logo screen and the host menu selection screen when you log in over a LAN.

Use the following procedure to log in from the command line:

1. In response to the system prompt, type:

   ```
   login host1 username
   ```

   where *host1* is the name of the host you want to log in to and *username* is your user name. Alternately, you can specify only the host name; PC-Interface prompts you for your user name.

   Press Enter (↵).

If the host is not available, PC-Interface returns the following message:

```
login: host1 not available.
```

Otherwise, PC-Interface prompts you for your password as follows:

```
Password:
```

2. Type your password and press Enter (↵). The characters you type do not appear on the screen, protecting the security of your password.

   PC-Interface attempts to establish a connection to the UNIX host you selected. If PC-Interface cannot establish a connection, it displays the following message on your screen:

   ```
   login: login failed.
   ```

3. When the connection is established, the following appears on your screen:

   hostname **File System Available as Drive D:**
   C>

The word *hostname* is replaced by the name of the host to which you have connected. Note that the drive specifiers you see might be different. PC-Interface associates the next available drive specifier supplied by DOS with the virtual drive. If the highest drive letter used by your personal computer is C:, the drive specifier for the first UNIX file system is D:.

Using The UNIX File System

Once you have completed the logging in process, you have access to the same UNIX files as you do when you access the UNIX host in any other way. These files are available to DOS on the virtual drive indicated in the connection message. The term "virtual drive" means that PC-Interface allows DOS to interface with the UNIX system as if the UNIX host were a local fixed disk drive even though it is not.

You 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. For example, to list a file called MEMO that is stored on the UNIX system, you might enter the following:

```
type d:memo
```

You can also change the default drive to the virtual drive, just as you would change to any physical DOS drive. If the virtual drive identifier specified in the connection message is drive D:, for example, enter:

```
d:
```

In general, you can use any DOS command or utility with files on the virtual drive. However, because UNIX handles all disk management functions for the virtual drive, the following DOS commands are unnecessary and cannot be used for the virtual drive:

CHKDSK
DISKCOMP
DISKCOPY
FDISK
FORMAT
SYS

If you try to issue one of these commands for the virtual drive or a file on the virtual drive, DOS returns a disk error.

You can copy files, including DOS commands or application programs, from diskette to the virtual drive following the same procedures you would to copy to a local fixed disk. You can create directories on the virtual drive, and you can set DOS search paths that specify directories on the virtual drive.

In short, you can do almost anything in your file services session you could if you were working on a stand-alone personal computer with a fixed disk.

In fact, as long as you only accessed files you had created on the virtual drive, you would never need to be aware of the UNIX system. But you would also be missing one of the important features of PC-Interface—the ability to take advantage of the UNIX environment.

When you use the host file services of PC-Interface, you can access other users' files (for which you have access permission), and you can make your files available to other users. You can use DOS applications to manipulate files created by a UNIX process or to create files for input to a UNIX process.

When you use PC-Interface in this way, you need to understand some UNIX features and a few basic differences between DOS and UNIX. The sections following "Ending a File Services Session" discuss some special considerations and the PC-Interface commands that allow you to perform UNIX-type functions from your host file services session. These topics include the following:

- Naming files
- Using DOS and UNIX text files
- UNIX file permissions

- Executing UNIX processes
- Printing with PC-Interface

Ending A File Services Session

Use the following procedure to end a host file services session on a virtual drive:

1. Change the current drive to the drive you started PC-Interface from. For example, if you started PC-Interface from a diskette, change to drive A: by typing:

   ```
   a:
   ```

   Press Enter (↵).

   If you started PC-Interface from a fixed disk, change the current drive to the fixed drive.

2. To log out of a specific virtual drive, type either:

   ```
   logout driveletter:
   ```

   or

   ```
   logout hostname
   ```

   where *driveletter* is the letter of the virtual drive and *hostname* is the name of the UNIX host from which you want to log out.

   To log out of *all* virtual drives at once, type:

   ```
   logout
   ```

PC-Interface disconnects you from the UNIX host or hosts and displays a message showing the drive letter and host name of each virtual drive from which you logged out.

Although you cannot access files on the UNIX file system of the logged-out host without starting another file services session, you can continue to use DOS and access files and programs located on diskettes, your personal computer's fixed disk, and virtual drives you are still logged into.

Note that any search paths previously set (for example, in AUTOEXEC.BAT at initialization) remain in effect. If these search paths reference a logged-out virtual drive, you may get unexpected error messages. For this reason, you should use the DOS PATH command to display your effective search path and delete any such references.

Naming Files

DOS And UNIX Differences

The rules for naming files and directories differ between DOS and UNIX with respect to case, size, and characters.

UNIX is case sensitive while DOS is not. Alphabetic characters in UNIX file and directory names are usually lowercase, but they can be any combination of upper- and lowercase. Whatever combination you enter is preserved. DOS, on the other hand, interprets all alphabetic characters in file and directory names as uppercase, whether you enter them in uppercase or lowercase. To UNIX, **chapter1** and **CHAPTER1** name two different files; DOS cannot

distinguish between the two forms, seeing them instead as the same name.

DOS limits file and directory names to eight characters plus an optional extension of up to three characters. UNIX allows names up to 14 characters. Although UNIX does not provide for file name extensions in the same sense as DOS, a UNIX name can contain a period anywhere in the name. Thus, while UNIX accepts any legal DOS name, DOS does not allow such perfectly good UNIX names as **messagetoall** or **chapter.seven**.

Both DOS and UNIX accept both alphabetic and nonalphabetic characters in file and directory names, but UNIX accepts more nonalphabetic characters than DOS. For example, control characters are valid characters in UNIX names but not in DOS names.

Creating File Names With PC-Interface

When you create files or directories during a file services session, your names must conform to standard DOS rules with respect to size and characters.

You can enter the name with either uppercase or lowercase alphabetic characters. When you create the file on a local diskette or fixed disk, DOS converts all alphabetic characters to uppercase as expected. When you create the file on the virtual drive, however, PC-Interface converts all alphabetic characters to lowercase. This prevents you from creating identical-except-for-case names (**chapter1** and CHAPTER1), which DOS cannot distinguish between.

Thus, any file you create with PC-Interface on the virtual drive can be accessed by either DOS or UNIX.

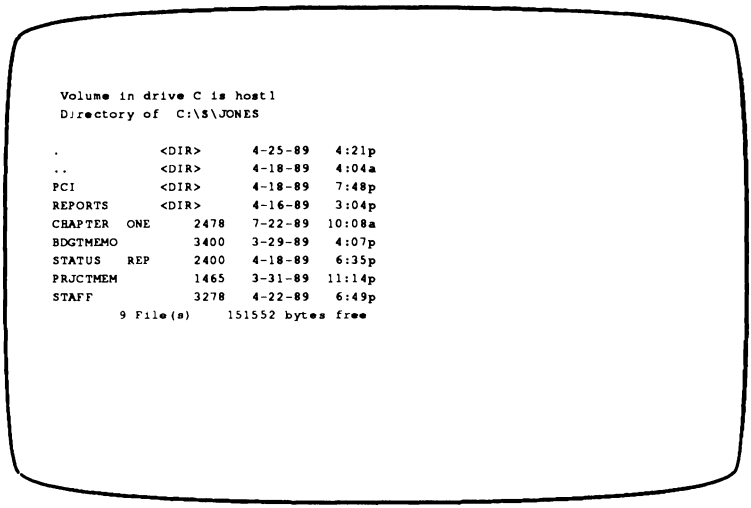
Accessing Files With PC-Interface

When you access files or directories during a file services session, you can enter the file name with either upper- or lowercase alphabetic characters. DOS converts all alphabetic characters to uppercase when searching a local disk; PC-Interface converts all alphabetic characters to lowercase when searching the virtual drive.

When file names are displayed, they always appear in uppercase. For example, entering the DOS DIR command to display the contents of a directory on the virtual drive yields the following display even though the actual UNIX names for these files are lowercase.[1]

[1] When you add or delete very small files on the virtual drive, you may not see a change in the number of bytes free displayed by DIR. The number of bytes free for the virtual drive is calculated based on UNIX blocks and, therefore, may be off by up to 4K as a result of round-off error.

Note also that on a virtual drive you may see different free space reported for different directories on the same drive. This is due to differences between the UNIX file system and the DOS file system. However, the free space reported is always correct for the current working directory.



```
 Volume in drive C is host1
 Directory of  C:\S\JONES

.            <DIR>       4-25-89   4:21p
..           <DIR>       4-18-89   4:04a
PCI          <DIR>       4-18-89   7:48p
REPORTS      <DIR>       4-16-89   3:04p
CHAPTER  ONE      2478   7-22-89  10:08a
BDGTMEMO          3400   3-29-89   4:07p
STATUS   REP      2400   4-18-89   6:35p
PRJCTMEM          1465   3-31-89  11:14p
STAFF             3278   4-22-89   6:49p
        9 File(s)     151552 bytes free
```

Notice that names that appear to DOS to have file name extensions, like **chapter.one**, are displayed in the standard DOS manner with the "extension" listed in a separate column.

In general, you can access any file or directory in the UNIX file system, whether it was created with PC-Interface or under UNIX. However, you must use a special mapped name for UNIX files or directories that do not conform to DOS rules. This includes names longer than DOS allows, names with more than three characters following a period, names with nonalphabetic characters that DOS does not recognize, and names with uppercase alphabetic characters.

When a UNIX name that does not conform to DOS rules is accessed by any DOS utility or application, PC-Interface translates, or maps, the name to a legal DOS name by appending a unique index consisting of an apostrophe

followed by one, two, or three characters.[1] If necessary, the UNIX file name is truncated before the index is appended. For example, a file called **messagetoall** might be mapped to the name MESS'BAQ. You can determine the mapped name by issuing the DOS DIR command. In the following example, the DIR display shows the mapped names for additional UNIX directories and files called **Mail**, **messagetoall**, **message.tobob**, and **message_toted**. Notice that the DOS DIR command shows only the mapped name, not the original UNIX name.

```
 Volume in drive C is host1
 Directory of  C:\S\JONES

.            <DIR>       4-25-89   4:21p
..           <DIR>       4-18-89   4:04a
PCI          <DIR>       4-18-89   7:48p
REPORT       <DIR>       4-16-89   3:04p
CHAPTER  ONE      2478   7-22-89  10:08a
BDGTMEMO          3400   3-29-89   4:07p
STATUS   REP      2400   4-18-89   6:35p
PRJCTMEM          1465   3-31-89  11:14p
STAFF             3278   4-22-89   6:49p
MAIL'FPE     <DIR>       6-19-89   8:50p
MESS'BAQ          2400   6-18-89   6:35p
MESS'BBF TOB       250   6-25-89   4:18p
MESS'BBS           598   6-30-89   6:01p
       13 File(s)     148304 bytes free
```

[1] On very large file systems, when all combinations of letters with an apostrophe have been used, the mapped name uses one of the following characters instead of the apostrophe:

- ~ (tilde)
- ^ (caret)
- [(left bracket)

To display the contents of **messagetoall** using the DOS TYPE command, you enter:

```
type mess'baq
```

Displaying UNIX-Style Directory Listings

Although you always use a file's mapped name with DOS commands, you sometimes want to know the original UNIX file or directory name. The PC-Interface UDIR ("UNIX directory") command displays the contents of a virtual drive directory in a form similar to what you would see in UNIX with the **ls -l** command, showing both the UNIX name and its corresponding mapped name.

For example, issuing the UDIR command for the directory \S\JONES on the virtual drive results in the following display:

```
Volume in drive C is host1
Directory of C:\S\JONES

.              .              jones  drwxr-xr-x  <DIR>       4-25-89   4:21p
..             ..             root   drwxr-xr-x  <DIR>       4-18-89   4:04a
pci            PCI            jones  drwxrwxr-x  <DIR>       4-18-89   7:48p
report         REPORT         jones  drwxrwxrwx  <DIR>       4-16-89   3:04p
chapter.one    CHAPTER.ONE    jones  -rwxr-----       2478  7-22-89  10:08a
bdgtmemo       BDGTMEMO       jones  -rwxrw-r--       3400  3-29-89   4:07p
status.rep     STATUS.REP     jones  -rwxr--r--       2400  4-18-89   6:35p
prjctmem       PRJCTMEM       jones  -rwxr--r--       1465  3-31-89  11:14p
staff          STAFF          jones  -r--r-----       3278  4-22-89   6:49p
Mail           MAIL'FPE       jones  drwxr-----  <DIR>       6-19-89   8:50p
messagetoall   MESS'BAQ       jones  -rw-r--r--       2400  6-18-89   6:35p
message.tobob  MESS'BBF.TOB   jones  -rw-r--r--        250  6-25-89   4:18p
message_toted  MESS'BBS       jones  -rw-r--r--        598  6-30-89   6:01p
      13 file(s)          148304 bytes free
```

Just as with the DOS DIR command, you can also specify specific file names or use the asterisk (*) or question mark (?) wild-card characters with UDIR. However, UDIR interprets these as DOS does, not as UNIX does.

For more on UDIR, refer to Chapter 6, Command Summary.

Changing File Names

Sometimes it is inconvenient to work with the mapped name. You might want to avoid constantly checking UDIR to determine mapped names for files whose UNIX names you know. Perhaps you regularly use DOS utilities or applications to create files for later use with UNIX processes, and you want to use names for these files that do not conform to DOS rules.

You can use the PC-Interface ON command with the UNIX **mv** command to rename files when the source or destination file names are not valid DOS names.

Suppose you have created several files on virtual drive D:—say, MESSAGE1, MESSAGE2, and MESSAGE3—during your file services session. If you would like these files to have more descriptive UNIX names, you can rename them by entering:

```
on d: mv message1 messagetobob
on d: mv message2 messagetojohn
on d: mv message3 messagetosteve
```

The UNIX **mv** command is similar to the DOS RENAME command, but provides you the additional ability to rename directories. For example, to avoid having to use the mapped name, which might be hard to remember or to type, you could

rename a UNIX directory called **directmail** with a valid DOS name by entering:

```
on d: mv directmail dmail
```

As illustrated, you always use the full UNIX name, not the mapped name, when you use the **mv** command with ON.

Finally, you can also use ON with **mv** to "move" a file from one directory to another on the virtual drive, as the following example shows:

```
on d: mv /s/jones/report/budget /s/jones/budgetmemo
```

Note that you must use UNIX–style path separators (/) when you use **mv** with ON.

For more information on the ON command, refer to Chapter 3, Using UNIX From DOS.

Using DOS And UNIX Text Files

DOS And UNIX Text Formats

DOS and UNIX differ from each other in the way they store text files. UNIX stores text lines as a sequence of characters terminated by a newline character. DOS, on the other hand, terminates text lines with a carriage-return character, followed by a line-feed character. A file created in one format appears corrupted when accessed by the other.

For example, suppose you have the following file in UNIX format:

```
Dear Bob:

This letter is in response to your inquiry of 25 February.
The information you requested is enclosed, along with other
materials that should help you evaluate our product line.
```

If you listed this file with the DOS TYPE command, you would see the following:

```
Dear Bob:

        This letter is in response to your inquiry of 25 February.
                                                                  T
he information you requested is enclosed, along with other
                                                          materials
that should help you to evaluate our product line.
```

Similarly, if you had created the original file in DOS format and then tried to edit it with the UNIX **vi** program, you would see the following:

```
Dear Bob:^M
^M
This letter is in response to your inquiry of 25 February.^M
The information you requested is enclosed, along with other^M
materials that should help you evaluate our product line.^M
^Z
```

Converting UNIX And DOS Files

You can use any file created during a PC-Interface file services session—your own or someone else's—just like any DOS file since these files are stored in DOS format even when they are created on the virtual drive.

However, to use a text file in UNIX format with DOS programs in your file services session, you must convert the file to DOS text format using the PC-Interface UNIX2DOS command. For example:

```
unix2dos  letter  ltr.dos
```

You can also convert the file and copy it from the virtual drive to a local drive on your personal computer in one step, as the following example illustrates:

```
unix2dos  d:bdgtmemo  c:budget
```

When you create text files during your file services session that you want to use later in a UNIX session, you can convert them to UNIX text format with the PC-Interface DOS2UNIX command. For example:

```
dos2unix  memo  memo.unx
```

When you enter the UNIX2DOS or DOS2UNIX command from PC-Interface, you use DOS file names. You cannot specify the same name for the source file and the target file, nor are you allowed to redirect your output back into the source file.

When you omit the second file name, these commands display the text file conversions on your screen but do not save them.

When you don't know the format of a text file, you should use the UNIX2DOS or DOS2UNIX commands to convert it to the format you need, just to be sure. The commands do not change anything when the file is already in the target format.

For more on DOS2UNIX and UNIX2DOS, refer to Chapter 6, Command Summary.

UNIX File Permissions

Unlike most DOS systems, the UNIX system can be accessed by more than one user. It therefore provides tools to prevent inspection, alteration, or execution of files by unauthorized users. The owner (usually the creator) of each UNIX file can grant or deny himself, other users in his user group, or all UNIX users permission to read, modify, or execute the file.

All the files and directories you create or access on the virtual drive during a PC-Interface file services session are protected by these permission assignments. The term "permission mode" (or just mode) refers to the specific assignments for a file or directory—that is, who can read it, who can write to it, and who can execute it.

Displaying File Modes

When you create a file or a directory on the virtual drive, it automatically inherits the default permission mode set by the UMASK environment variable.[1] You can check the current mode of any file or directory by issuing the UDIR command to display the file's permission mode. The third column of the display shows the owner, and the fourth column shows the

[1] For more information on setting UMASK environment variables, contact your system administrator.

current mode as a 10-character string, in the following form:

```
drwxrwxrwx
```

The first character indicates whether the entry is a file (-) or a directory (d).[1] The remaining nine characters indicate the read, write, and execute permissions in groups of three. That is, the first set of three characters represents the permissions for the file or directory owner, the second set represents the permissions for others in the owner's user group, and the last set represents the permissions for everyone else.

An **r** specifies permission to display or read the file; a **w** specifies permission to modify or write the file; and an **x** specifies permission to execute the file as a UNIX program. A hyphen (-) in any position indicates that the particular permission is denied.

Changing File Modes

Use the PC-Interface ON command with the UNIX **chmod** command to change the permission modes for your files and directories on virtual drives. For example, suppose you want to protect a file on virtual drive D: called BUDGET so that no one, not even you, can modify it. Issue the following command:

```
on  d:  chmod  a=r  budget
```

[1]The first character can also indicate a block device (b), a character device (c), a symbolic link (s), or a named pipe (p).

In this command, you first specify the virtual drive where the **chmod** command is to be run, followed by the group your change affects—**u** for you the owner, **g** for others in your user group, **o** for others on the system, and **a** for all three groups—followed by an equal (=) sign and the explicit permission modes you are assigning.

Alternately, you can indicate only the permissions you want to change without having to specify all the permission characters. To change the write permission so you can modify your BUDGET file, you could enter the following:

```
on  d:  chmod  u+w  budget
```

To give write and execute permission to others in your user group as well as everyone else on the system, you could enter the following:

```
on  d:  chmod  go+wx  budget
```

In this form of the command, you specify the groups your change affects, followed by a plus (+) or minus (-) sign, depending on whether you are granting or denying the permission, followed by the permissions you are changing.

Effect Of Directory Permissions On Files

You must have execute permission for a directory on the virtual drive in order to access any of the files in that directory. Without execute permission, you cannot copy files from or into the directory, make the directory your

working directory, or rename, inspect, or modify any file in the directory.

Therefore, to provide access to files, be sure you also assign the appropriate permissions to your directories. On the other hand, if you want to completely deny access to files in a directory without changing the mode for each file, you could remove execute permissions from the directory.

DOS Commands And File Permissions

Because DOS does not have the same system of file permissions, the interaction of DOS commands with files on the virtual drive can result in a few unexpected results.

When you execute DOS programs that you have stored on the virtual drive, DOS does not refer to the presence or absence of execute permission for the file. Rather DOS identifies executable files by their file name extensions—.COM, .EXE, or .BAT. Just be sure you have read permission for DOS executable files and that they are in directories with execute permission.

Sometimes the message DOS returns is affected by the file permissions. For example, when a DOS command you issue encounters a file for which you do not have read access, DOS returns a "File not found" message even though the file exists. Similarly, if you try to create a file in a directory for which you do not have write access, DOS returns a "File creation error" or "Access denied" message. The DOS COMP program, when used with a file on the virtual drive for which you do not have read permission, returns the error "Sharing violation."

DOS Applications And File Permissions

Certain DOS applications, when used with PC-Interface in a multiuser environment, will not protect your files from being simultaneously updated by you and another user with write permission; that is, they do not support DOS file sharing and record locking.

In these cases, you should carefully consider which combination of file and directory permissions give you the most appropriate protection. For example, to prevent a file from being simultaneously updated by someone else while you are working on it, you could temporarily remove all others' read and write permissions for that file. This would prevent anyone from even looking at the file until you were done.

File Sharing And Record Locking

When you use applications that support file sharing and record locking[1] file permissions are less important. PC-Interface supports standard DOS file-sharing and record-locking conventions. An application supporting file sharing (for example, a word-processing program) opens files in such a way that other applications cannot access the files until they are closed. This prevents the file from being modified by more than one user at a time. Similarly, an application supporting record locking, such as a database management program, opens files in such a way that no one else will be able to access a record while you are working on it.

[1] Most networked versions of DOS applications are of this type.

Note that not all applications support file sharing and record locking. The discussion below applies only to those that do.

When installing network versions of certain DOS applications, you are prompted to choose a network type. We recommend that you try "IBM PC Network" or another MS-NET compatible network type first, but you may have to experiment to find the selection that best supports file sharing and record locking.

Do not attempt to perform COPYs or BACKUPs on files on your virtual drive that are being accessed by another user. There may be locks on the files (or records), and the copy you make may not be complete.

If you find that a file appears to be locked when there are no other users logged in or that access is allowed to a file that should be locked, there may be a problem with file sharing or record locking, and you should notify your system administrator.

Executing UNIX Processes

If you are familiar with both DOS and UNIX, you might find occasions during a host file services session when you would like to use a UNIX program instead of a DOS command or application. The PC-Interface ON utility lets you issue a noninteractive UNIX command from within your host file services session without having to switch to terminal emulation.

For example, if you are working on virtual drive D: and you want to generate a list of users logged on to the

UNIX system and have the output displayed on your terminal, you would type:

```
on - who
```

When you do not want to wait for a UNIX command to finish before resuming your DOS work, you can redirect the output to a file and use the & option, as follows:

```
on - who > wholist &
```

To find out the status of a process for which you did not wait, type:

```
jobs
```

For more about ON, refer to Chapter 3, Using UNIX From DOS, and Chapter 6, Command Summary.

Printing With PC-Interface

During a PC-Interface host file services session, you can print files from both virtual drives and local drives. You can print jobs on either a local printer attached to your personal computer or a remote UNIX printer.

When you start a PC-Interface host file services session, all your printing is directed to your local printer by default. You can use any of the standard DOS printing methods:

- Pressing the Shift and print-screen (PRT SC)[1] keys to print current screen contents.
- Pressing the CTRL and PRT SC (or CTRL and P) keys together to print continuous screen contents.
- Issuing the DOS PRINT command and specifying the appropriate file name.
- Using the DOS COPY command to copy a file to the local printer device (PRN, LPT1, LPT2, or LPT3).
- Invoking the print function from within a DOS application.

A feature of PC-Interface is that in addition to printing from your local printer, you can send your print requests to a remote printer connected to the UNIX system. By specifying the print stream (LPT1, LPT2, or LPT3) and UNIX host in the PRINTER command, you can direct printing to any of three remote print streams.

Specify the UNIX host in one of the following ways:

- By host name.
- By the drive letter associated with the host.
- With a hyphen (-), meaning the host associated with the current working drive (that is, the virtual drive from which you issue the PRINTER command).

For example, if you were logged into a UNIX host called **frog** on virtual drive D:, you could use any of

[1] If you have an enhanced keyboard, use the print-screen (PRT SC) key without the Shift key.

the following commands to direct LPT1 print requests to **frog**'s system printer:

```
printer  lpt1  frog
printer  lpt1  d:
printer  lpt1  -
```

If you omit the print stream identifier, LPT1 is assumed.

When you specify remote printing with the PRINTER command, you can use the following print methods:

- Pressing the Shift and print-screen (PRT SC)[1] keys to print the current screen contents. When you use this method, printing is always directed to LPT1.
- Pressing the CTRL and PRT SC (or CTRL and P) keys to print continuous screen contents.
- Using the DOS COPY command to copy a file to one of the three remote print streams (LPT1, LPT2, or LPT3).
- Invoking the print function from within a DOS application.

Where and how each remote print stream prints depends on the UNIX print command associated with it. Under UNIX, each print command causes output to be sent to a particular printer. For example, a **troff** command might cause printing to be output to a laser printer, while an **lp** command might direct output to a line printer.

[1] If you have an enhanced keyboard, use the print-screen (PRT SC) key alone.

The PRINTER command allows you to specify the UNIX print command, and therefore the printer, for each of the three print streams. For example, if you enter:

```
printer  lpt3  -  troff
```

then all subsequent print requests directed to LPT3 are processed by the **troff** program on the current host and are printed on the printer associated with the **troff** command. When you use this method of printing, you can include any valid options for the UNIX print command as well as pipes and redirection. When the command string contains more than one word, it must be enclosed in quotation marks.

If you do not specify a print command for a particular print stream, PC-Interface sends print requests for that print stream to the system default print command for output to the default system printer. If you wish, you can establish your own defaults as described in Chapter 5, Tailoring PC-Interface.

Setting up your print streams as described above causes your jobs to print correctly most of the time. In the unlikely event that your print jobs are being broken up into pieces or truncated, increase the printer timeout by appending the following to the end of your print stream definition:

```
/t180
```

Refer to the manual page for PRINTER in Chapter 6, Command Summary, for more information about this and other options to the PRINTER command.

Switching Back To Local Printing

You can switch back to local printing at any time during your host file services session by typing:

```
printer  lptn  local
```

If you're not sure whether your printing is being directed to your local printer or to a remote printer, type:

```
printer
```

PC-Interface displays the current printer setting and options in effect for each of the three print streams.

For more on PRINTER, refer to Chapter 6, Command Summary.

Chapter 3. Using UNIX From DOS

Introduction

The PC-Interface ON utilities allow a personal computer user to execute a UNIX program on a specified host computer and view the output as if the program had actually run on the personal computer. The ON utilities comprise three DOS programs—ON, JOBS, and KILL.

ON can only be used to execute *noninteractive* UNIX commands—those that do not initiate a conversation with the user. Thus, ON is not the same as and does not replace a terminal emulation session, which is still necessary to run interactive UNIX programs such as text editors.

With ON you can:

- Execute noninteractive UNIX commands without having to start a terminal emulation session, for example, when terminal emulation is not available.
- Extend the functionality of DOS by allowing UNIX commands to be executed on DOS files as if they were DOS commands.

ON is significantly more efficient than terminal emulation over the network and uses fewer resources on the host computer. Using ON instead of terminal emulation lightens the burden on a heavily loaded host computer.

Using ON

There are two ways to use ON. The first is to enter ON followed by a host identifier and then the UNIX command you wish to execute. For example to run the **cal** command on the current host (-), type:

```
on - cal
```

The result is a calendar of the current month displayed on your screen. If you want a specific host to execute your UNIX command, you could type, for example:[1]

```
on host1 cal
```

where HOST1 is the name of the host, or

```
on d: cal
```

where D: is the virtual drive identifier.

UNIX commands can contain all options and arguments exactly as they would be typed at a UNIX prompt. You should specify UNIX file names with their UNIX names, not with their mapped names. You should also use UNIX-style path names within the UNIX command.

ON cannot execute multiple UNIX commands separated by semicolons. Use individual ON commands instead.

By default, ON executes a Bourne shell (**/bin/sh**) on the host computer. Other shells must be explicitly invoked.

ON automatically converts the text output of the UNIX command from UNIX format to DOS format. That is, the UNIX2DOS utility is built in.

If ON cannot execute the requested UNIX command, either because it cannot find a requested file or because you do not

[1]The host name must be typed exactly as shown on the login map. If you are connected to a host file service session over an RS-232 connection path, you would type **com***n* (where *n* represents the number of the serial port) rather than the name of the host.

have execute permission for a requested file, the following error message is returned:

unixcommand: **access denied or file not found**

where *unixcommand* is the name of the UNIX command that ON tried to run.

Renaming ON

What's In A Name?

The second way to use ON is to make a copy of the DOS file ON.EXE, giving it the name of a UNIX command. You can then execute the UNIX command on the current virtual drive by entering only the UNIX command name.

For example, if you make a copy of ON.EXE named LS.EXE, you can enter:

```
ls -l
```

from the DOS prompt and receive the resulting directory listing from the current UNIX host.

All options and arguments are entered following the name of a renamed copy of ON exactly as they would be at a UNIX prompt. This feature allows you to make copies of ON with the names of UNIX utilities and use them on DOS files as if they were DOS utilities.

Suppose you wanted to be able to search DOS files for particular character strings and to display the ends of DOS

files. First, you would create new "DOS utilities" equivalent to the UNIX **grep** and **tail** commands, as follows:

```
copy on.exe grep.exe
copy on.exe tail.exe
```

Then you could invoke these copies of ON just as you would the namesake UNIX utility. For example, these commands:

```
grep -n Wilson letter.txt
tail -23 letter.txt
```

would report every line in LETTER.TXT that contains the word "Wilson" and then display the last 23 lines of LETTER.TXT on your screen.

The following restrictions apply when making renamed copies of ON:

- The UNIX command that ON is renamed to resemble must contain only lowercase characters in its name.

- The name of the UNIX command cannot violate any of the DOS naming conventions. For example, you could not create an ON version of a shell script named **mycalendar** because its name contains too many letters.

  Commands that do not adhere to these restrictions can still be run by using the ON command explicitly.

  Note: Do not make copies of ON.EXE with the names JOBS or KILL. The ON utilities interpret JOBS and KILL as built-in commands. When you type them directly from your DOS prompt, the built-in versions are run. If you

want to execute UNIX versions of these commands from DOS, use the first form of the ON command, like this:

```
on - jobs
on - kill pid
```

- Do not make copies of ON.EXE with the names of DOS internal commands. The DOS shell interprets DOS internal commands, such as TYPE, as soon as they are entered. You therefore cannot run ON commands renamed with these names directly from the DOS prompt. To run a UNIX command with the same name as a DOS internal command, use the first form of the ON command, like this:

```
on - type cat
```

Linking ON On The Virtual Drive

You can use the DOS COPY command to create a library of renamed copies of ON and place them all in a directory in your DOS search path. The disadvantage of this method is that each copy takes up valuable disk space.

A better idea is to place ON in a publicly accessible directory on the virtual drive and use the UNIX **ln** command to create the library of copies. The advantage of using UNIX **ln** over DOS COPY is that **ln** allows a file to have more than one name without taking up extra disk space.

You can use ON itself to do the linking. Use this procedure:

1. First, make the directory that contains ON.EXE the current directory of your virtual drive.

2. Make a linked copy of ON named LN, as follows:

```
on - ln on.exe ln.exe
```

3. Use the new LN to make copies of ON for **grep** and **tail**:

```
ln on.exe grep.exe
ln on.exe tail.exe
```

You cannot use **ln** to link copies across UNIX file system boundaries or across virtual drive. See **cp** in your UNIX reference manual for more information on **ln**.

Setting Environment Variables

By default, ON uses the following UNIX environment in which to run UNIX commands:

- The current directory for ON-initiated UNIX commands is the current directory of the virtual drive you selected. For instance, if your virtual drive is D:, and the current directory of D: is \USR\HOMEDIR, then any UNIX program you invoke is run in D:\USR\HOMEDIR.

- The default UNIX search path is the following:

  the current directory of the virtual drive
  /bin
  /usr/bin

ON recognizes two DOS environment variables that allow you to alter the default UNIX environment:

1. The UPATH variable is used to specify an alternate UNIX search path. The format used is the same as for the UNIX

Bourne shell PATH variable, with path names separated by colons. For example, you might change the default UNIX search path as follows:

```
set UPATH=:/lbin:/usr/ucb:/bin:/etc:/usr/bin
```

Slashes (not backslashes) are used as path separators in the UPATH variable.

2. The EXPORT variable contains a list of other DOS environment variables that are to be placed in the UNIX environment. For example, you might add the following variables to your DOS environment:

```
set home=/usr/paula
set tz=PST8PDT
set lpdest=laser2
```

You can then export these DOS variables to the UNIX environment with this command:

```
set export=home tz lpdest
```

Because DOS environment variable names are always uppercase, only all uppercase variable names can be exported to the UNIX environment.

ON converts variable names to uppercase as they are exported. This means that everything to the left of the equals sign in the DOS SET command can be entered in either case. Thus, these are equivalent commands:

```
set lpdest=laser2
set LPDEST=laser2
```

By contrast, however, ON passes everything to the right of the equals sign literally. This means that the values of DOS variables that are exported to the UNIX environment must be

entered exactly as they would be at a UNIX prompt. For example, on a host with a UNIX printer named **laser2**, this command sets the LPDEST variable correctly:

```
set lpdest=laser2
```

This command is incorrect:

```
set lpdest=LASER2          (incorrect)
```

To view variables that have been exported to the UNIX environment, type:

```
on - set
```

Expanding DOS Environment Space

By default, DOS sets aside only 160 bytes of memory to contain all of the characters in all DOS environment variables. The PC-Interface user can quickly run out of DOS environment space simply by (1) setting a DOS search path containing long virtual drive path names, (2) setting a UPATH variable, and then (3) setting several more variables to export to the UNIX environment.

The following line, placed in your CONFIG.SYS file, reserves more environment space:

```
shell=driveletter:\command.com /e:nn /p
```

where *driveletter* is the drive ID of the fixed disk or diskette that you boot from, and where (for DOS 3.10) *nn* is an integer from 10 to 62 representing the number of 16-byte paragraphs of memory to reserve. For DOS 3.2 and above, *nn*—an integer ranging between 160 and 32,768—represents the number of bytes of memory to reserve.

The /E parameter of COMMAND.COM is present but undocumented in most versions of DOS 3.10. It is documented in IBM PC-DOS 3.2 and above.

Breaking Out Of ON

Unless ON is running as a detached task, as described in the next section, you can interrupt it by typing CTRL-C or CTRL-BREAK. The following prompt is displayed:

```
a - abort, c - continue, d - detach:
```

If you don't select one of the three options, the prompt is redisplayed.

Entering `a` kills the job, clears the job from the job table, and returns you to a DOS prompt. Entering `c` allows the process to continue as before in the DOS foreground. Entering `d` detaches the UNIX process from your terminal, returns your DOS prompt, and continues to run the specified program in the background under UNIX. The next section describes the job table and detached tasks.

Job Control

The ON utilities allow you to:

- Start one or more UNIX commands from your DOS prompt and run them in the background.
- Detach an ON command that is currently running in the foreground so that it continues to run in the background.

Your DOS prompt returns and you can issue other commands while your ON program continues.

- View a table showing the status of all currently executing and completed ON commands.
- View the output of detached commands, including commands that have finished executing, at any time.

The following sections describe the features for controlling jobs executed with ON.

The Job Table

PC-Interface maintains a job table that keeps track of detached UNIX processes initiated by ON—that is, processes that are placed in the background. The job table shows the status of up to ten detached ON commands.

The job table holds one entry per ON command. Each UNIX command, however, may create several subordinate processes. For example a single **spell** command takes only one space in the DOS job table, but appears in a UNIX **ps** (process status) list as four or five processes.

All UNIX systems limit the number of simultaneous processes each user can run. This limit is typically 20 processes per user. If several UNIX commands initiated with ON each create several processes, the limit for processes per user may be reached before the limit for entries in the job table.

Two Ways To Detach Tasks

You can detach UNIX tasks initiated with ON so they run in the UNIX background in either of two ways:

1. Add an ampersand (&) to the end of the ON command. For example:

   ```
   on - spell memo &
   ```

   Note that the ampersand must be preceded by a space. The following command is incorrect:

   ```
   on - spell memo&                    (incorrect)
   ```

2. Initiate an ON command and later interrupt it with CTRL-C or CTRL-BREAK. For example:

   ```
   on - spell memo
   <CTRL-C>
   ```

   The following prompt is displayed:

   ```
   a - abort, c - continue, d - detach
   ```

   When you enter `d`, the task is placed in the background under UNIX.

In either case, ON responds with a prompt of the form:

[jobnumber] pid

where *jobnumber* is the job number of the task in the DOS job table and *pid* is the UNIX process ID number returned by UNIX. Your DOS prompt then returns and you can issue additional commands (including ON commands) while your detached process continues executing in the background.

Any output produced by a detached ON task (for example, the spelling errors found by the **spell** command shown previously) is by default stored in a temporary file. You can see the contents of this temporary file by reattaching the task as described later under "Reattaching To Detached Tasks." You can also use pipes and redirection to send the output of detached tasks to any file or program you choose. See "Pipes and Redirection," later in this chapter.

Keep these factors in mind when initiating detached tasks:

- If the job table is full at the time you attempt to start another process, ON returns the error message:

  ***unixcommand*: `job table full`**

  where *unixcommand* is the name of the UNIX process that ON attempted to run. This error message is returned when every position in the job table is filled, whether with "Done" entries or "Running" entries.

- If the maximum number of UNIX processes has been reached when you issue an ON command, ON returns this error message:

  `Unix exec failed`

  This message may occur when there are still places to fill in the job table. The total number of UNIX processes you can start is set by UNIX independently of the maximum number of job table entries available.

Keeping Track Of Detached Tasks

You can use the JOBS command to perform any of the following tasks:

- Display job table information.
- Clear the job table of entries for completed jobs.
- Reattach to detached jobs.

Invoking JOBS with no arguments displays the current job table in the following format:

```
JOB   HOST       STATE    EXIT STATUS   COMMAND
[ 1]  d: host1   Running                 unixcommand1
[ 2]  e: host2   Done       exit(0)      unixcommand2
```

If the job table is currently clear, invoking JOBS returns you to the DOS prompt. The job table is clear when no detached jobs have yet been run or when you have cleared the job table of all entries. The columns in the job table report the following information:

1. The "JOB" column shows the job ID number that ON assigned to each process when the process was detached.
2. The "HOST" column shows the host on which the process is running or has run.
3. The "STATE" column indicates whether the process is running or done.
4. The "COMMAND" column shows the UNIX command that was requested.

5. The "EXIT STATUS" column shows one of the following values:

exit(*nn*)	The job terminated with an exit status of *nn*. An exit status of 0 usually means the process terminated normally. Any other value may indicate an abnormal termination of the process.
unknown	The job terminated, but ON was unable to determine its exit status. This happens when you log out of a file services session before the job is complete.
signal(*nn*)	A signal was received that killed the process. In this case, *nn* indicates the signal received.
coredump(*nn*)	The signal received caused a core dump to occur. This is a special case of **signal**.
err3(*nn*)	An error in the functioning of ON or PC-Interface has occurred.

Completed processes remain in the job table until you clear them by entering JOBS with a single hyphen as an argument. When you enter:

```
jobs -
```

these events occur:

- The current status of the job table is displayed, including all currently "Done" entries.
- The "Done" entries are cleared from the job table.

- Any temporary files in the **/tmp** directory that were associated with the "Done" jobs are removed.

To see the cleared job table, issue another JOBS command.

Once jobs are cleared from the job table, they can no longer be reattached. Output from any task being saved in temporary files for reattachment and review is discarded.

Reattaching To Detached Tasks

Using the ON utilities, you can reattach a task that has been detached as long as you have not cleared it from the job table. Reattaching allows you to:

- View all previous output of a currently running program and continue to view new output as the program sends it to your screen.

- View the output of a program that has finished running.

To reattach either to a currently running job or to a completed job, use the JOBS command in the form:

```
jobs %jobnumber
```

where *jobnumber* is the job number of the detached task. For example, suppose you start the UNIX **spell** program as a detached task by typing:

```
on - spell memo &
```

When you type this command, ON prints a job number and process ID similar to:

```
[1] 4376
```

To reattach to your **spell** job, type the command:

```
jobs %1
```

If you decide to reattach to the detached job and don't know the job number, type the JOBS command to find out. A display similar to this is printed:

```
JOB   HOST          STATE    EXIT STATUS     COMMAND
[ 1]  d: host1      Running                  spell memo
[ 2]  e: host2      Done       exit(0)       who
```

Entering `JOBS %` without a number reattaches you to the lowest numbered task in the job table.

When you reattach to a currently running job (one that appears as "Running" in the job table), you see all the output produced by the job up to the time you reattach to it. The job then continues to run, and you see any additional output as it is printed. If you wish, you can detach the task again at any time while it is running. Each time you reattach it, you see *all* output printed up to the time you reattach.

When you reattach to a completed job (one that appears as "Done" in the job table), you see all output printed by the job. The temporary file storing the output is then removed, and the job table entry for that job is cleared. The temporary file is also removed and the job table entry cleared if the job completes while it is attached.

Saving Output From Completed Jobs

One way to save the output of detached jobs for future reference is to use DOS redirection at the time you reattach a job. For example, assume an ON - SPELL job has completed and appears as follows in the job table:

```
JOB   HOST          STATE    EXIT STATUS     COMMAND
[ 1]  e: host2      Done       exit(0)       spell memo
```

You can save the list of spelling errors when you reattach to the job by typing the command:

```
jobs %1 > typos
```

This command redirects the list from your screen into a file called TYPOS.

You can also use pipes and redirection when you first issue the ON command. See "Pipes and Redirection," later in this chapter, as well as the ON command section of Chapter 6, Command Summary.

Stopping Detached Jobs

You can halt detached jobs and clear them from the job table by reattaching, typing CTRL-C or CTRL-BREAK, and responding `a` to the resulting prompt.

You can also stop a UNIX process by using the ON utilities KILL command, which runs under DOS, in either of these forms:

kill [*–signal*] *%jobnumber* [. . .]
kill [*–signal*] [*hostname*\|*driveletter*:] *pid*[. . .]

In the first form, *jobnumber* is the task number of the job in the job table and *signal* is the UNIX signal to be sent to the process.

In the second form, *pid* is the UNIX process ID of the process to be signaled. If *pid* is negative, then the entire UNIX process group is signaled. If no host or drive is specified, the host

associated with the current drive is used. (If it is not a virtual drive, an error occurs.) When this form is used, the job table entry is not automatically cleared.

The ON utilities KILL command is similar to the UNIX **kill** command except that it accepts *%jobnumber* in addition to the process ID number argument. See the *kill(1)* and *signal(2)* entries in your UNIX manuals for more information. The default signal sent is 15. A signal of 9 may be used for a sure kill. Note, however, that entering `kill -9` is bad practice unless other KILL commands have failed, because the UNIX program has no chance to perform cleanup operations before exiting.

Pipes And Redirection

With the ON command, you can use either DOS or UNIX pipes or redirection, or even combine the DOS and UNIX versions of these mechanisms in a single command. You can accomplish nearly all useful operations using the DOS mechanisms. The following sections therefore concentrate on DOS pipes and redirection. A few comments on UNIX pipes and redirection are included in "Using Pipes and Redirection in Detached Tasks."

DOS Pipes

You can use DOS pipes (|) with UNIX programs that you invoke using ON just like you use them with DOS programs. The following examples show two different ways to accomplish the same task:

```
on - ls | find "chap"
on - ls | on - grep chap
```

Both versions locate and display all the file names in the current directory that contain the characters **chap**. The first example returns the output of the UNIX **ls** command to the DOS environment and redirects it through the DOS FIND command.

The second example pipes the output of **ls** to another command started with ON. The output of **ls** is returned to the DOS environment by ON and is then rechanneled with the DOS pipe symbol to the command ON - GREP CHAP (which runs the UNIX **grep** command).

DOS Output Redirection

You can redirect the output of ON-initiated commands to specified files. DOS output redirection (>) allows you to direct the output to a file on any drive and directory where you have write permission. For example, you could put the results of a command onto a diskette:

```
on - ls | find "chap" > a:tempfile
```

DOS redirection of the final output of an ON command works because ON sends the output back to your DOS environment. Thus, you can redirect the output of both UNIX and DOS commands with the DOS output redirection mechanism (>).

DOS Input Redirection

You can use the DOS input redirection mechanism (<) to redirect standard input from a file on a local or virtual drive or from the DOS keyboard to an ON command.

Input redirection from a file is useful when you want to invoke a UNIX command to operate on a file contained on a DOS

device, such as drive A:. The following command, for example, does not work because A:NAMES is not a meaningful UNIX file name:

```
on - spell a:names                    (incorrect)
```

Because the UNIX **spell** command can read from the standard input as well as from a file, you can use input redirection in a command like this to accomplish the operation:

```
on - spell < a:names
```

The DOS input redirection mechanism works together with the DOS drive designation A: to transfer the contents of the file NAMES to the UNIX **spell** command. Note, however, that you cannot use this form of redirection with UNIX programs that cannot read from the standard input. For example, the UNIX **ls** (directory listing) program cannot read from standard input, so the following command does not work:

```
on - ls < a:                          (incorrect)
```

You can also redirect the input for a UNIX command from the DOS keyboard. For example, you can open a text file TEMP on drive C: and type into it as follows:

```
on - cat > temp < con:
This is line 1 of input.
This is line 2 of input.
<CTRL-Z>
```

CTRL-Z is entered to close the DOS standard input, which also effectively closes the UNIX standard input.

Using Pipes And Redirection In Detached Tasks

A detached ON command that includes DOS pipes or redirection may not produce the results you want. This can happen because the DOS command processor, COMMAND.COM, interprets the pipe and redirection symbols (<, >, and |), while ON interprets other parts of the command line, including the ampersand (&).

For example, if you want to redirect the output of a **spell** command into a file called TYPOS, you might issue the following command:

```
on - spell memo > typos &          (incorrect)
```

This command, however, does not put the list of spelling errors in the file TYPOS. Instead, it puts the job number and process ID of the **spell** command (which ON returns to the DOS environment) in TYPOS.

To save output produced by **spell** you can do either of the following:

- Issue the command in the form:

  ```
  on - spell memo &
  ```

  and then redirect the output when you reattach, as described previously under "Saving Output from Completed Jobs."

- Use UNIX output redirection rather than DOS output redirection when you issue the ON - SPELL command.

The ON utilities use the special characters { to mean UNIX input redirection, } to mean UNIX output redirection, and ! to

mean a UNIX pipe.[1] When you use these characters in an ON command, UNIX does the redirection or piping operation, and the results are often easier to understand. To use UNIX output redirection in the example illustrated above, type:

```
on - spell memo } typos &
```

This example displays a job number and process ID on your screen when you issue the command. Your DOS prompt returns, as expected, and the task runs in the background. When the **spell** program completes, the results are in the file TYPOS.

The following example shows how UNIX pipes can be useful. It shows both incorrect and correct ways of creating a file called NAMES containing a sorted list of users currently logged into the current UNIX host.

```
on - who | on - sort > names &          (incorrect)
on - who ! sort ! unix2dos } names &    (correct)
```

If you issue the first command, it is not detached until the **who** program finishes, and the file NAMES contains a job number and process ID rather than a sorted **who** list. This is probably not the result you want. The second command uses a UNIX pipe to send the list output by the **who** program directly to the UNIX **sort** program, without returning to DOS or using a second ON command. Because the output is not returned to the DOS environment, it remains in UNIX text format rather than being converted to DOS text format. The **unix2dos** program is therefore used to convert the list to DOS format,

[1] Alternately, you can put the UNIX commands in quotes to prevent the standard UNIX pipe and redirection symbols from being interpreted by DOS. For example:

```
on - "spell memo > typos" &
```

following which UNIX output redirection (}) puts the list into the file NAMES.

To be properly interpreted by ON, the special redirection and pipe characters ({, }, and !) require spaces surrounding them. The following examples, which omit the required spaces, are incorrect:

```
on - spell memos}typos &          (incorrect)
on - who!sort!unix2dos}names &    (incorrect)
```

For further information on using UNIX pipes and redirection with ON, see the ON command in Chapter 6, Command Summary.

Summary Of Restrictions And Cautions

- ON cannot run interactive UNIX programs, except those that can be run with DOS console redirection as described in "Redirecting from the DOS Console" earlier in this chapter.

- Renamed copies of ON cannot be named after UNIX commands or shell scripts that contain an uppercase letter in their names; however, they can still be run by preceding the command name with ON -.

- ON cannot, in a single command line, initiate multiple UNIX commands separated by semicolons. Use multiple ON commands.

- ON's special use of the exclamation point and braces (as aliases for the UNIX pipe and redirection symbols) works

only when these symbols are preceded with blank space. When one of these symbols is embedded in a character string or is preceded with a backslash, it is interpreted as a UNIX character.

- Although you can mix DOS and UNIX pipes and redirection in an ON command, if you place a command containing DOS redirection in the background, the results may not be what you expect. Similarly, you cannot redirect DOS output using UNIX redirection. See "Pipe and Redirection Restrictions" earlier in this chapter.

- The default 160-byte size of the DOS environment may not leave enough room to export all necessary variables to the UNIX environment. See "Expanding DOS Environment Space" earlier in this chapter for more information.

- The ON utilities are not tested with, and may not run under, alternate DOS shell processors that completely replace COMMAND.COM.

Chapter 4. Using Your Computer As A UNIX Terminal

Introduction

This chapter tells you how to use the PC-Interface terminal emulation program. The terminal emulator, EM2, lets you use your personal computer like a VT220 or VT100 terminal, enabling you to conduct a standard UNIX session on a host computer. While this chapter focuses on emulation of the VT220 terminal, most of the discussion also applies to VT100 emulation.

PC-Interface terminal emulation supports most of the attributes of a standard VT220 terminal, including multinational character sets, 20 programmable function keys, numeric keypad cursor control, flow control (CTRL-Q and CTRL-S), and control and escape sequences. The terminal emulator does not support full 132-column mode or smooth scrolling.

In addition to terminal emulation, there are two other ways you can use PC-Interface to interact with a UNIX host. Chapter 2 tells you how to use host file services with DOS applications, and Chapter 3 discusses using UNIX processing capabilities from the DOS environment. You can work entirely in one mode, or you can switch between modes.

This chapter describes:

- Beginning and ending a terminal emulation session.
- Switching between terminal emulation and host file services or other local DOS session.
- Configuring terminal emulation parameters.
- Getting help with terminal emulation.

The procedures described in this chapter and the examples shown assume that PC-Interface has been installed for your personal computer as described in Appendix A.

Beginning A Terminal Emulation Session

You can conduct a terminal emulation session independently of a host file services session or in conjunction with a file services session. When you conduct a terminal emulation session in conjunction with a file services session, you must begin your file services session before you begin your terminal emulation session.

The following procedure starts a terminal emulation session. This allows you to communicate directly with a host computer. If you already have a host file services session in progress, start with step 3.

1. If you are using a fixed disk, leave the drive A: door open. If you are using a diskette, insert your PC-Interface working diskette in drive A: and close the drive door.

2. Switch ON your personal computer if you have not already done so. If the system unit is already on, press and hold the CTRL and ALT keys; then press the DEL key.

   The following message appears:[1]

```
PC-Interface Release 3.0 (Serial #x.xxxx) Initialized
```

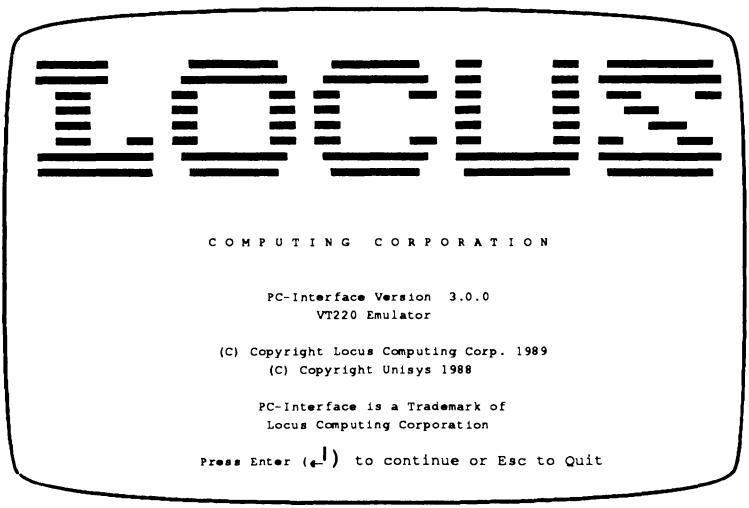
[1] Depending on other software that may be installed for your personal computer, you might see other messages as well.

3. In response to the DOS prompt, type:[1]

   em2

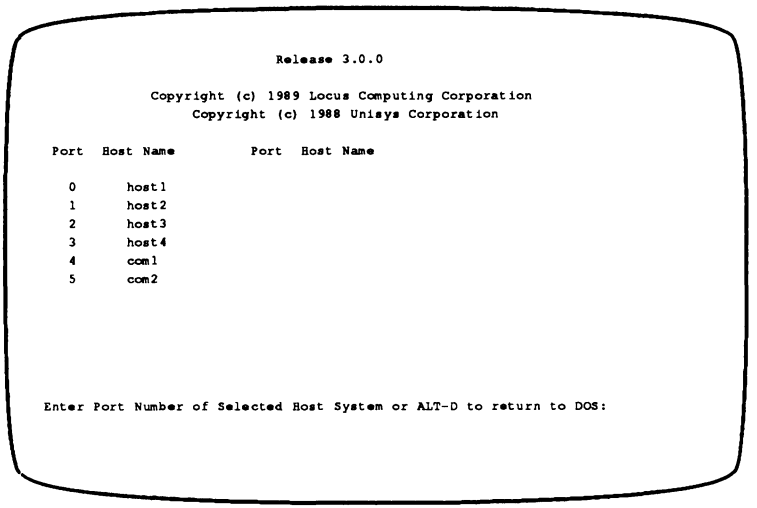
   Press Enter (↵).

   The PC-Interface logo screen appears:



   Press Enter to continue. The PC-Interface terminal emulation host-selection menu screen appears.

[1] The EM2 command can be issued from any drive, including a virtual drive if you have a host file services session in progress, as long as the program is in a directory on your path or you qualify the command with the appropriate drive specification (for example, D:EM2). When you save your terminal session, the PC-Interface terminal emulator writes the session file (EM.SES) to the current directory on the current drive (unless you set the EMSES environment variable when you tailor your PC-Interface installation as described in Chapter 5).



The PC-Interface terminal emulation host menu screen shows you the UNIX host connection paths available to you. When your personal computer is connected to one or more hosts over a LAN, the host names appear on the menu. In addition, PC-Interface supports host connections through one or two serial ports. When your personal computer is equipped with serial ports, the word **com***n* appears on the menu once for each available serial port.

You may have one or more choices on your menu screen. Depending on the size of your network, there may be more hosts than can be displayed on the screen. In that case, you can establish a session with a host that is not shown (as long as the host is part of your network) by using the command-line emulation procedure. See "Beginning A Terminal Emulation Session From The Command Line," later in this chapter.

4. In response to the host-selection prompt, type the port number that appears on the menu next to the desired host

name (or the desired serial port if you want to establish an RS-232 connection) and press Enter (↵).

A connection-attempt message replaces the host-selection prompt. When the connection is successfully established, the screen clears and the UNIX host login banner and prompt appear. (If the connection cannot be established, press the ALT and D keys in combination to return to host file services or your local DOS session and contact your system administrator.)

5. Respond as you normally do when conducting a UNIX terminal session (usually by entering your UNIX user name and password).[1]

Beginning A Terminal Emulation Session From The Command Line

You can bypass the PC-Interface logo screen and the host menu selection screen by starting your terminal emulation session from the command line. This method is also useful when your network has more hosts than can be displayed on

[1] The UNIX host must recognize your terminal as a VT220 terminal. UNIX recognizes the terminal capabilities of your personal computer by referring to the TERM environment variable. You should include the following lines, which set the TERM variable properly, in the **.profile** file in your home directory if you use the PC-Interface terminal emulator regularly:

```
TERM=VT220
export TERM
```

If you use the C shell, put this line in your **.login** file:

```
setenv TERM vt220
```

the host menu screen and you want to start an emulation session on a host whose name is not displayed because there is not room for it on the screen.

Use the following procedure to start a terminal emulation session from the command line:

1. In response to the DOS prompt, type:

   `em2` *hostname*

   where *hostname* is the name of the host on which you want to start the emulation session. (To start an RS-232 terminal emulation session from the command line, use `com1` or `com2` as the host name.)

2. Press Enter (↵).

   When the connection is established, the screen clears and the UNIX host login banner and prompt appear. (If the host you specified is not available, the message:

   `em2:` *hostname* **`not available`**

   is displayed. In that case, you may want to just enter `em2` to see what hosts are available.)

3. Respond as you normally do when conducting a UNIX terminal session.

Session Screen And Status Line

During your terminal emulation session, your terminal screen shows the results of communication between your personal

computer and the UNIX host, just as if you were using a VT220 terminal connected directly to the host.

The last line on your screen shows the status of the terminal emulator (unless the Status Line parameter has been set to No Display) as follows:

```
ALT: D=DOS L=Logout H=Help M=Menu ||hostname|| 8/VT220 CsrMd=Nml KpMd=Num
```

The left side of the status screen shows you the following information:

- Pressing the ALT and D keys in combination returns you to DOS (but does not end your terminal emulation session).

- Pressing the ALT and L keys in combination closes the connection between the host and the personal computer. In most cases, this logs you out of your UNIX session.

- Pressing the ALT and H keys in combination invokes the Help screen.

- Pressing the ALT and M keys in combination invokes the Configure menu.

The center area of the status screen (between the bars) shows the name of the host to which you are currently connected. If you are connected over RS-232, this field displays `com1` or `com2`, depending on which serial port you are connected through.

The next field of the screen shows the character size, if applicable, and the terminal type. In the example above, the terminal type is 8-bit VT220. For a VT100 terminal, the character size is always 7-bit.

The right portion of the screen shows the cursor mode (CsrMd) and the keypad mode (KpMd). The default mode for the cursor keys is Normal; that is, the keys are used to move the cursor. The default mode for the keypad keys is Numerical, which means that the keys have their face value. The alternate value for both sets of keys is Application, which means that the keys generate application-specific codes. You can change the mode of either or both sets of keys as necessary. See "Changing Configuration Parameters" later in this chapter for further information.

Ending A Terminal Emulation Session

The following procedure ends your terminal emulation session:

1. Log out of your UNIX session as usual—by typing `logout` or pressing CTRL-D.

2. Press ALT-L. The host-selection menu appears, giving you an opportunity to establish another terminal emulation session.

3. To return to host file services or local DOS, press ALT-D.

If you always log into terminal emulation over the same connection path, you can skip the host selection procedure by using the alternate logout method described below:

1. Log out of your UNIX session in the usual way.

2. Press ALT-D.

The PC-Interface terminal emulator creates an EM.SES file in the current directory on the current drive from which you issued the last EM2 command.[1] Your terminal emulation session screen is saved in this file. The size of the file is about 4K. Your screen now appears just as it did when you issued the EM2 command.

The next time you issue the EM2 command, PC-Interface establishes a connection over the same path as your previous session, and displays the session screen from the current EM.SES file. Respond in the normal way to the new login prompt from the UNIX host.

If you want to log in to terminal emulation over a different connection path or host, use the first procedure, closing your terminal emulation session with ALT-L.

Switching Between Terminal Emulation And Host File Services Or Local DOS

The following procedure allows you to switch back and forth between your terminal emulation session and host file services (or local DOS):

1. While in terminal emulation mode, press ALT-D.

[1] Alternately, you can tailor your PC-Interface installation to specify an absolute location for the EM.SES file regardless of the current directory or drive. Refer to "Setting the Terminal Emulation Environment Variable" in Chapter 5.

The PC-Interface terminal emulator creates an EM.SES file in the current directory on the current drive from which you issued the last EM2 command.[1] Your terminal emulation session screen is saved in this file. The size of the file is about 4K. Your terminal emulation session screen clears and your screen appears just as it looked when you issued the EM2 command.

2. You can now continue your host file services or local DOS session.

3. When you want to return to your terminal emulation session, change back to the same directory and drive from which you entered the previous EM2 command (the directory and drive that contains the EM.SES file) and type:

   ```
   em2
   ```

   Press Enter (↵).

   Your terminal emulation session screen appears just as it looked when you pressed ALT-D. You are still connected to the UNIX host, logged in, and in the same directory you were in when you left terminal emulation. That is, CD (change directory) commands issued while you are in host file services do not affect the current directory for your terminal emulation session.

[1] Alternately, you can tailor your PC-Interface installation to specify an absolute location for the EM.SES file regardless of the current directory and drive. Refer to "Setting the Terminal Emulation Environment Variable" in Chapter 5.

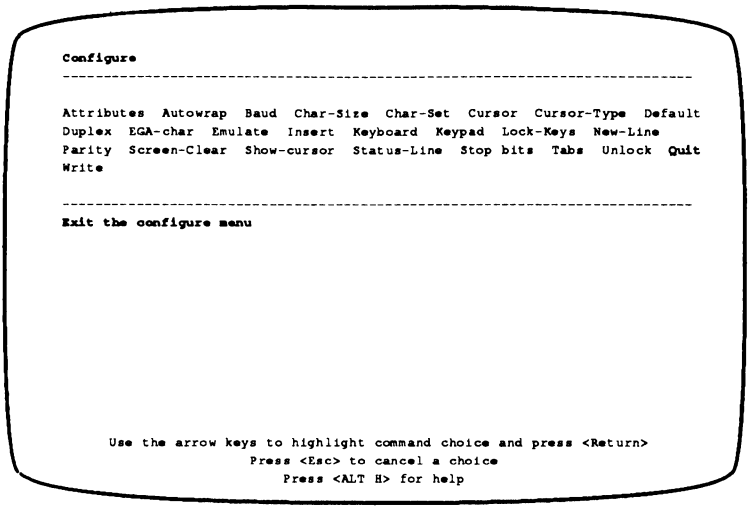
Changing Configuration Parameters

The PC-Interface terminal emulator has a number of configurable parameters whose values determine the way terminal emulation works. Configurable attributes include the type of terminal to emulate, the keyboard nationality, aspects of the way the screen looks (including cursor type and presence or absence of a status line), and the modes in which the keys operate, among others.

Each parameter has a default value. You may wish to use the EM2 program with the preset parameter values, or you may wish to change the values of some or all of the parameters. The Configure menu allows you the flexibility of setting these attributes to values that best suit you.

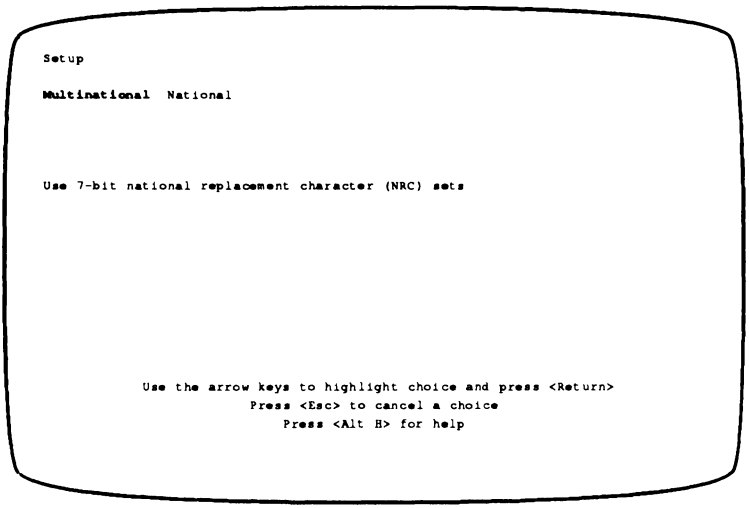
You can change configuration parameters either when you establish a terminal emulation session or at any time while you have an emulation session active. You can reset parameters each time you use the emulator, or you can save your selections in a file and reuse them each time you invoke the emulator.

Before you can invoke the Configure menu, you must have already selected a host and received a login prompt. Once the login prompt displays, press ALT-M to invoke the Configure menu. The Configure menu is shown below:



The currently selected item, **Quit**, is highlighted on your terminal screen. (In the illustration above, it is shown in boldface.) To select an item for configuring, press the right or left cursor key until the desired item is highlighted. When the item is highlighted, a phrase describing the item displays underneath the list of items, at the left of the screen. For example, in the screen shown above, the phrase **Exit the configure menu** describes the item **Quit**.

After you select an item, press Enter (↵). A submenu called the Setup menu displays. The Setup menu for the item Char-Size is shown below:



As shown above, the Setup menu for a parameter shows the possible values for that parameter. The default value is highlighted (represented above as boldface). To select a value different from the default, use the cursor keys to move to the desired value. When the desired value is highlighted, press Enter (↵).

Once you have selected a value from a Setup menu, you are returned to the Configure main menu, with the **Quit** item highlighted. To configure more items, repeat the procedure. When you finish configuration, press ALT-M again to return to (or begin) your terminal emulation session.

Saving Parameter Choices

You can save your parameter selections in a file so that you do not need to reconfigure each time you invoke the emulator. To

save your configuration choices for reuse in future terminal emulation sessions, use the following procedure:

1. Select the desired values for items on the Configure menu. If you want to use an item's default value, you do not need to select that item.

2. When all parameters are configured as desired, select the **`Write`** item (the last item on the Configure menu) and press Enter (↵).

Your configuration choices are always saved in a file called EM.DEF. This file is saved in the same directory as your EM.SES file. See "Setting the Terminal Emulation Environment Variable" in Chapter 5 for more information.

The next time you invoke the emulator, your emulation session is automatically configured according to the parameter selections saved in the EM.DEF file.

Getting Help

The terminal emulator provides context-sensitive help in different forms depending on what you are trying to do when you invoke the emulator. Help is always invoked by pressing ALT-H.

Help From Terminal Mode

If you invoke Help from a UNIX session, the help displayed consists of a set of screens showing how PC keys and key combinations map to VT220 and VT100 terminal keys. The first page of this screen is illustrated below:

```
You are now in terminal communications mode.  After this help screen goes away,
press ALT-M to enter Configure mode.  Press ALT-L to log out of the
current host and return to the host selection menu.  Press ALT-D to exit this
emulation session and return to DOS.  This emulation session is resumed the
next time you invoke EM2.  Press CTRL-BREAK to send a BREAK signal to the host
computer.  Press ALT-P to send all screen output into a capture file.

VT100/VT220 Keyboard Layout:

 VT220          PC                 VT220          PC
keypad 0      ALT_0               PF1           F1
keypad 1      ALT_1               PF2           F2
keypad 2      ALT_2               PF3           F3
keypad 3      ALT_3               PF4           F4
keypad 4      ALT_4               F6            F6*       (VT220 Only)
keypad 5      ALT_5               F7            F7        (VT220 Only)
keypad 6      ALT_6               F8            F8        (VT220 Only)
keypad 7      ALT_7               F9            F9        (VT220 Only)
keypad 8      ALT_8               F10           F10       (VT220 Only)
keypad 9      ALT_9               F11           CTRL_F1   (VT220 Only)
keypad ,      ALT_=               F12           CTRL_F2   (VT220 Only)
keypad -      ALT_-               F13           CTRL_F3   (VT220 Only)
keypad .      SHFT_F5             F14           CTRL_F4   (VT220 Only)

Press ALT-H to quit Help. Press any other key to continue Help.
```

The above screen shows you, for example, that if you want to generate the code sequence that the F13 key on a VT220 keyboard produces, you press CTRL-F3 on the personal computer keyboard.

Help From The Configure Menu

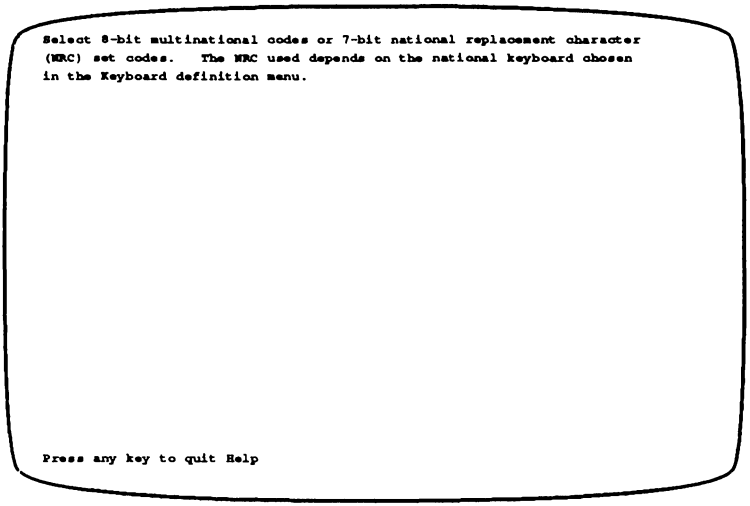
When you invoke Help from the Configure main menu, the following screen displays:

```
From this menu you may configure EM2 to match your needs and to match the
requirements of the host computer.  An asterisk (*) indicates EM2's default
settings.  You may choose from the following menu items:

Autowrap        Specify *No or Yes for automatic character wrap to next line.
Baud            Set the baud rate for RS-232 communication to one of the
                following:  45.5, 50, 75, 110, 134.5, 150, 300, 600, 1200,
                1800, 2000, 2400, 3600, 4800, 7200, *9600, 19200, 38400.
Char-Size       Set the character size for RS-232 communication to 5, 6, 7,
                or *8 bits.
Char-Set        VT220 emulator can use *8-bit multinational character set or
                7-bit national replacement character (NRC) sets.  The NRC
                matching the National Keyboard parameter chosen is used.
Cursor          Set the cursor keys to *Normal or Application Mode.
Cursor-Type     Set cursor type to either *Underline or Block.
Default         Reset all parameters to their default values.  This happens
                automatically each time you change emulators.
Duplex          *Full, Half
Emulate         DEC VT100, *DEC VT220
Insert          Set characters sent to screen to *Overwrite existing
                characters or Insert in the line.


Press ALT-H to quit Help, press any other key to continue Help
```

The screen above provides descriptions for as many of the Configure menu items as can fit on one screen. Descriptions of the rest of the items are on the next page of the screen, which you display by pressing any key.

Help From A Setup Menu

You can get more specific help by invoking Help from a Setup menu. For example, suppose you want help with the item Char-Size. After selecting that item from the Configure menu, press ALT-H. The following screen displays:



To return to your terminal emulation session after invoking one of the Help screens, simply press ALT-H again.

Chapter 5. Tailoring PC-Interface

Introduction

The installation instructions in Appendix A combine all the PC-Interface files with the DOS operating system on a working diskette or on your fixed disk. You may not find it convenient to work with diskettes or to have all the PC-Interface files or other DOS applications you use on your local drive.

This chapter tells you how to tailor your installation to put PC-Interface where it is most convenient for your use. It also tells you how to create directories on a virtual drive for your DOS applications and how to access public directories to share DOS applications with other users.

This chapter also describes how to change certain configuration options and how to set certain environment variables that let you establish how you want PC-Interface to perform certain functions. By setting these in your PC-Interface CONFIG.SYS and AUTOEXEC.BAT files, you can ensure that your PC-Interface environment is always the way you want it.

The PC-Interface Files

The PC-Interface files are installed on your working diskette or on your fixed disk when you choose a standard installation according to the installation instructions in Appendix A. Some of these files are required to initialize PC-Interface; others are needed to establish a connection to your UNIX host; the rest provide additional utilities that extend the power of PC-Interface.

The following table lists the PC-Interface files:

Files Required to Initialize PC-Interface	
AUTOEXEC.BAT	Invokes PCIINIT automatically when DOS is loaded.
BRIDGE.DRV	Provides basic PC-Interface support.
CONFIG.SYS	Identifies device drivers to be loaded by DOS.
\ETC\HOSTS	Searched by PCIINIT for your personal computer's internet address.
driver.DRV	LAN device driver (*driver* represents the name of your network interface board driver).
PCIINIT.EXE	Initializes PC-Interface.
Files Required for Host File Services	
LOGIN.EXE	Establishes a host connection for host file services.
LOGOUT.EXE	Terminates a host file services session.
Files Required for Terminal Emulation	
EM2.EXE	Establishes a terminal emulation session.
PC-Interface Utilities	
DOS2UNIX.EXE	Converts text files in DOS format to UNIX format.
DOSWHAT.EXE	Returns program version numbers. Support tool.
JOBS.EXE	Displays the status of remote UNIX jobs initiated with the ON command.
KILL.EXE	Kills remote UNIX jobs initiated with the ON command.
PCICONF.EXE	Lets you configure certain PC-Interface parameters.
ON.EXE	Executes UNIX commands on a specified remote host.
PRINTER.EXE	Directs output to local or remote printer.
UDIR.EXE	Lists virtual drive directory contents in UNIX style.
UNIX2DOS.EXE	Converts text files in UNIX format to DOS format.

Putting PC-Interface In A Subdirectory

When you install PC-Interface on your personal computer with a fixed disk according to the instructions in Appendix A, you can choose to install all the PC-Interface files in the root directory or to install some of them in a subdirectory.[1] If you install all the files in the root directory and later decide it would be more convenient to work from a subdirectory, you can relocate all the PC-Interface files except AUTOEXEC.BAT, CONFIG.SYS, and HOSTS to a subdirectory. AUTOEXEC.BAT and CONFIG.SYS must remain in your root directory; HOSTS must remain in \ETC.

Use the following procedure to relocate PC-Interface files to a subdirectory:

1. Change your working directory to the root directory by typing:

```
cd \
```

2. Create a subdirectory (in this example, PCIBIN) by executing:

```
mkdir pcibin
```

3. Copy all the PC-Interface files *except* AUTOEXEC.BAT and CONFIG.SYS to PCIBIN, one at a time. For example:

```
copy dos2unix.exe pcibin
```

[1] Note that the HOSTS file is always installed in the \ETC subdirectory, even when you choose to install all files in the root directory.

copies the file DOS2UNIX.EXE to the subdirectory PCIBIN.

4. Edit your AUTOEXEC.BAT file, adding the following line before the line containing the PCIINIT command:

   ```
   path=c:\pcibin[1]
   ```

   (If you already have a path command in AUTOEXEC.BAT, add `c:\pcibin` to it; do not add another PATH command.)

5. Modify the CONFIG.SYS file to insert the appropriate path qualifier before each DEVICE command. For example, change:

   ```
   device=driver.drv
   ```

   (where *driver*.DRV represents the driver for your network interface board) to:

   ```
   device=pcibin\driver.drv
   ```

6. Erase the copied PC-Interface files from the root directory. For example:

   ```
   erase dos2unix.exe
   ```

Putting PC-Interface Files On A Virtual Drive

You need to keep only the files required to initialize PC-Interface and the files required for host file services on your

[1] Substitute the drive specifier of your local drive if other than C:.

local drive. The remaining files can be executed just as effectively from a virtual drive once you have established a host file services session. In fact, you can even put the LOGOUT program on a virtual drive.

If you want to establish a terminal emulation session without first establishing a host file services session, you must also keep the EM2.EXE file on your local drive.

Use the following procedure to install the PC-Interface utilities and any other executable files you want on the virtual drive in your home directory:

1. Initialize PC-Interface and establish a host file services session.

2. Change your current drive to a virtual drive by typing:

   ```
   d:
   ```

   (Substitute the drive specifier of the virtual drive you want to use.)

3. Make sure your home directory is the current directory on the virtual drive by typing:

   ```
   cd
   ```

   Verify that the directory displayed is the directory in which you want to install PC-Interface files. (If not, use the CD command to change to the appropriate directory.)

4. Change back to your local drive by typing:

   ```
   c:
   ```

   (Substitute the drive specifier of your local drive.)

5. Copy the PC-Interface utility files one by one to the virtual drive. For example:

```
copy dos2unix.exe d:
```

6. Erase the files you have copied from your local drive. (You can restore them at any time by copying them back from the virtual drive or from your backup diskettes.) Do *not* erase the following files:

   BRIDGE.DRV
   driver.DRV
   LOGIN.EXE
   PCIINIT.EXE
   \ETC\HOSTS

   These files are required for initializing PC-Interface and using host file services. (Do not erase the file EM2.EXE if you plan to use terminal emulation without a concurrent host file services session.)

Often you want to consolidate frequently used commands or programs in a central, easily accessible directory. Since such directories usually contain only executable binary files, they are sometimes called *bin* directories (short for binary).

You can create a personal bin directory on your virtual drive (use the DOS MKDIR command to create a directory called PCIBIN) and copy the PC-Interface utilities to that directory instead of your home directory, using the procedure described above.

Be sure your search path includes these directories so DOS can find your PC-Interface files.

Putting DOS Applications On A Virtual Drive

You can also use personal bin directories on a virtual drive to contain your DOS applications. To install DOS applications on a virtual drive, first establish a host file services session, create a bin directory for the application, and change to that directory. Then, in most cases you can simply follow the manufacturer's instructions for installing the DOS application on a fixed disk.

Some DOS applications create or append to an existing AUTOEXEC.BAT file in the root directory to invoke the application automatically when you turn on your personal computer. When you install such an application on a virtual drive, this file is created on that virtual drive. However, since you do not have a connection to the virtual drive when you turn on your personal computer, the AUTOEXEC.BAT file cannot automatically start the application. You should rename the AUTOEXEC.BAT file to a unique name that suggests the application and has a .BAT extension (for example, STARTWS.BAT). Whenever you want to invoke the application, change your current directory to the bin directory in which you installed the application and type the file name without the extension. For example:

```
startws
```

Copy-protected DOS applications that require a "key" diskette in drive A: before they can be started from a fixed disk require the same "key" diskette when you invoke them from a virtual drive. Installing DOS applications on a virtual drive does not circumvent any copy-protection mechanism that may exist for the application.

Copy Protection/Licensing Issues

You are responsible for observing the licensing restrictions associated with DOS applications you install on a virtual drive. Unless the licensing agreement permits you to share the software, you should only install applications in your own directories, and you should remove read and execute permission for all other users from those directories.

Public-domain software and DOS applications for which you have a multiuser licensing agreement can be installed in public bin directories on a virtual drive. Generally, the system administrator installs these packages and can tell you what applications are available at your installation and where they are.

Be sure your search path includes these directories so DOS can find the applications.

Setting Search Paths

In order to access PC-Interface utilities or DOS applications stored on a virtual drive, you have to tell DOS where to find them. You can do this each time you enter a command by first changing to the drive and directory that contain that command.

An easier way to tell DOS where to find commands is to set a search path. A path is a list of directory locations that you want DOS to search, in the order you specify, whenever DOS needs to find a command or program. You can specify a search path at any time using the DOS PATH command. This search path stays in effect until you issue another PATH command.

There are two very important things to remember when you specify directories on a virtual drive in your PATH command:

- You must have a host file services session in progress for DOS to search directories on a virtual drive.
- When you log out of a host file services session but plan to continue using local DOS, you should reset your search path to exclude any references to virtual drives.

If you relocated PC-Interface files to a subdirectory on your personal computer fixed disk, be sure you include the subdirectory any time you change the search path.

Setting A Default Search Path

To avoid having to remember to set your search path to include a virtual drive, you can include the PATH command in your AUTOEXEC.BAT file, specifying the virtual drive identifier you want to use.

However, specifying a virtual drive in your PATH in AUTOEXEC.BAT has some disadvantages. Any time you want to use a virtual drive that is not set in AUTOEXEC.BAT, you have to reset the search path. You also have to reset the search path whenever you want to use DOS without establishing a host file services session. This is because when DOS uses the search path and encounters a drive specification for which it does not have access (for example, the virtual drive when no host file services session is in progress), DOS returns an "Invalid Drive" message and stops searching for the command or program you requested.

You can solve this problem by creating a default search path that you set only after you establish a host file services session, using the following procedure:

1. Establish a host file services session.

2. In your home directory on the virtual drive, create a file called SETPATH.BAT, which contains a PATH command in the following form:

   ```
   path=%1:\usr\pcibin;a:\
   ```

   Be sure to include all the directories you want DOS to search in the order in which you want them searched.

You can set the default search path as soon as you establish a host file services session by changing to the virtual drive, then typing:

```
setpath d:
```

(Substitute the drive identifier of the virtual drive you want to use.) DOS substitutes this identifier for every %1 in the PATH command included in SETPATH.BAT and executes the command.

Changing The Maximum Number Of Virtual Drives

By default, PC-Interface is configured to support a maximum of four virtual drives. If you want to be able to establish connections to more than four virtual drives, you can change this configuration option.

Using any editor, append:

```
/d:nn
```

to the DEVICE=BRIDGE.DRV line in your CONFIG.SYS file, where *nn* is the maximum number of virtual drives, up to 16. For example, if you want to be able to establish connections to as many as 10 virtual drives, you would modify the DEVICE command to read:

```
device=bridge.drv /d:10
```

The change takes effect the next time you reboot the system.

Changing The Job Table Size

By default, PC-Interface supports a job table with 10 entries. In most cases, this job table size is sufficient.

If you wish to configure a job table with a larger or smaller number of job entries, you can change the job table size.

Using any editor, append:

```
/j:nn
```

to the DEVICE=BRIDGE.DRV line in your CONFIG.SYS file, where *nn* is an integer from 1 to 30 representing the maximum number of entries in the job table. For example, if you want the job table to be able to hold up to 20 entries, you would modify the DEVICE command to read:

```
device=bridge.drv /j:20
```

The job table is configured to the new size the next time your personal computer is booted.

The job table holds one entry per ON-initiated task. However, remember that one UNIX task may spawn several child processes. For example, a single **spell** command may show up in the DOS job table as one running process but would show up in a UNIX **ps** list as four or five processes.

All UNIX computers have a limit on the number of processes each user can have running at one time. This limit varies from system to system but normally falls in the range of 20 to 30 processes per user. If several ON-initiated UNIX tasks each spawn several child processes, the total number-of-processes-per-user limit may be reached long before a job table configured for 20 to 30 entries is filled.

Setting The Printer Environment

PC-Interface allows you to direct your remote printing requests to any one of three remote print streams—LPT1, LPT2, and LPT3—and to associate a particular UNIX command string with each, using the PRINTER command.

You can establish your own default environment for print streams by putting PRINTER commands in a batch file that you run after logging into PC-Interface. Once you set up print streams, your specifications remain in effect until you explicitly change them by entering new PRINTER commands. (See "Printing With PC-Interface" in Chapter 2.)

In addition, you can specify a default UNIX command string that PC-Interface uses whenever no other print command is in effect for a print stream by including the following line in your AUTOEXEC.BAT file:

```
set printprog=printprogram
```

where *printprogram* is the default UNIX command string for all undefined print streams. This default overrides the system default set by the system administrator.

Setting ON Variables

The ONNAME Environment Variable

ON recognizes the DOS environment variable ONNAME, which is provided to help resolve file name conflicts.

Since there can be several copies of ON, each with a different name, ON uses the contents of the ONNAME variable to recognize its default name for itself. If no ONNAME variable is specified, the value "ON" is presumed. This is how ON distinguishes between the commands **on - ls** and **ls** in the case where you have made a copy of ON.EXE named LS.EXE.

In the event that you wish to rename ON itself, you must specify the new name in the ONNAME variable. You may also rename KILL.EXE and JOBS.EXE in this manner.

The ONNAME variable recognizes three comma-separated fields representing, respectively, the internal names of ON.EXE, KILL.EXE and JOBS.EXE. By default, the three

fields have the values ON, KILL, and JOBS. Any field can be blank to retain the default name, and trailing commas may be omitted.

For example, suppose you want to enter:

```
RUN - ls -l
```

instead of:

```
ON - ls -l
```

Follow these steps:

1. Rename ON.EXE to RUN.EXE.

2. Put this line in your AUTOEXEC.BAT file:

```
set onname=run
```

If you want to rename JOBS.EXE to, for example, J.EXE at the same time, use this SET command:

```
set onname=run,,j
```

If you simply rename ON.EXE, KILL.EXE or JOBS.EXE without also changing the ONNAME variable, ON attempts to run a UNIX command with the specified name.

The arguments specified in the ONNAME variable must be in lowercase letters. That is, in a SET ONNAME command, every letter to the right of the equals sign must be lowercase.

The ONPREFIX Environment Variable

ON passes a UNIX command to the host by prefixing the specified UNIX command with **/bin/sh -c**. Thus, an ON command such as:

```
on - cal
```

is passed to the host as:

```
/bin/sh -c cal
```

The value of the ONPREFIX variable can be used to interpose a shell command before the UNIX command to be executed. If ONPREFIX is set, the command:

```
on - cal
```

is sent to the host in this form:

```
/bin/sh -c prefix cal
```

where *prefix* is the value of the ONPREFIX variable.

This feature can be useful in these cases:

- If you want *all* UNIX commands initiated by ON to execute on the host under an alternate shell such as the C shell, you can set ONPREFIX as follows:

  ```
  set onprefix=/bin/csh -c
  ```

  This use of ONPREFIX initiates a Bourne shell that calls the alternate C shell to execute the UNIX command.

  If you want only one command to run under the alternate shell, it is better accomplished by leaving ONPREFIX

unset and using ON as follows:

```
on - "/bin/csh -c unixcommand"
```

Note that if you invoke an alternate shell, the default path for that shell applies.

- You can use ONPREFIX to specify a shell script that sets up a specific UNIX environment in which to run commands. The shell script must contain the following line:

  ```
  $*
  ```

  This line causes the UNIX command to be run by the script.

  This method of setting up the UNIX environment is an alternative to setting DOS environment variables and exporting them to the UNIX environment.

  For example, you can set up a shell script that initializes any number of environment variables and then performs accounting tasks. If this script is named **setup** in your personal **bin** subdirectory, set ONPREFIX as follows:

  ```
  set onprefix=/usr/homedir/bin/setup
  ```

  Thereafter, the command:

  ```
  on - unixcommand
  ```

  is passed to the UNIX host as follows:

  ```
  /bin/sh -c /usr/homedir/bin/setup unixcommand
  ```

Setting The Terminal Emulation Environment Variable

When you leave a terminal emulation session without closing the session, PC-Interface creates a file called EM.SES that contains your current session screen. PC-Interface uses this file to recreate your session screen when you reenter terminal emulation.

Normally, PC-Interface creates EM.SES on the drive and in the directory from which you invoked the EM2 command. If you change directories between one invocation of EM2 and the next, PC-Interface either finds a different EM.SES file and displays the screen from a session other than the one you just left or finds no EM.SES file at all and displays a blank screen. Either way can be confusing.

You can tell PC-Interface that you always want the EM.SES file created and searched for in a particular location by including the following line in your AUTOEXEC.BAT file:

```
set emses=path
```

where *path* specifies the drive, directory, and filename.

Note that the EM.DEF file, which contains your terminal emulator configuration choices if you decide to save them, is saved in the same directory.

Chapter 6. Command Summary

Introduction

This chapter presents a detailed description of the PC-Interface commands. The commands are presented in alphabetical order, and each description includes the purpose, format, and remarks about the command.

Command Notation

The following notational conventions are used to describe the format of PC-Interface commands:

[ ] Items shown inside square brackets are optional. To include optional items when you enter a command, type only the information inside the brackets. Do not type the brackets.

CAPS Words shown in capital letters are called *keywords*. The PC-Interface command names are keywords. You type keywords exactly as shown, although you can use either uppercase or lowercase letters when you issue the command from DOS. When a PC-Interface command is issued from UNIX, it must be entered in lowercase letters.

italics Items shown in lowercase italic letters indicate generic items for which you are to substitute an appropriate value. For example:

filename

indicates that you should type the name of your file in place of the word *filename*.

Italic items shown inside of square brackets are optional.

| A vertical bar means either/or. Choose one of the separated items and type it as part of the command. For example:

ON|OFF

indicates that you should type either ON or OFF, but not both. Do not type the vertical bar.

. . . An ellipsis indicates that you can repeat the item.

Include all punctuation such as commas, equal signs, question marks, asterisks, colons, slashes, and backslashes. Punctuation shown inside square brackets is optional.

DOS2UNIX Command

Purpose: The DOS2UNIX command converts a text file from DOS format—lines ending in a carriage-return and line-feed sequence and end-of-file marked with a CTRL-Z (octal 018)—to UNIX format—lines ending in a new-line (ASCII line-feed). The command can be issued from either the DOS operating system or the UNIX operating system.

Format: **DOS2UNIX** [/B|/U|/L][/F] [[*path*]*filename*] [[*path*]*filename*]

Remarks: The first file specified is the source file. The second file specified is the target file. The source file and the target file must not have the same name. When neither file parameter is specified, DOS2UNIX reads from standard input and writes to standard output. When only one file parameter is specified, that file is considered the source file, and the output is written to standard output.

If the source file is already in UNIX style, DOS2UNIX does not alter the format of the file.

The /B, /U, and /L parameters are mutually exclusive. These parameters, which can be specified either before or after the file names, have the following meanings:

/B Preserves 8-bit (binary) character representations. When this parameter is not specified, DOS2UNIX truncates all characters to 7-bit ASCII representation.

/U Converts text to uppercase.

/L Converts text to lowercase.

Note: When neither /U nor /L is specified, the case of each text character is unchanged.

DOS2UNIX normally removes all extra carriage-return characters preceding the end-of-line character during the end-of-line conversion. The /F parameter prevents the removal of these extra carriage returns and forces the translation of only the final carriage-return and line-feed sequence into a new-line. Although not normally used, this parameter preserves unusual end-of-line sequences when complete file invertibility is required.

When issued from the UNIX operating system, the /B, /U, /L, and /F parameters must be specified with the UNIX hyphen (-) rather than the DOS slash (/). From the DOS operating system, use either the slash or the hyphen.

DOS2UNIX can be used in combination with DOS and UNIX pipes and redirection. For example, the command:

```
sort text | dos2unix > newtext
```

sorts the DOS file TEXT, converts the sorted text, and writes it to the UNIX file **newtext**. Note that you cannot redirect a converted file to itself.

EM2 Command

Purpose: The EM2 command invokes the PC-Interface terminal emulator. In terminal emulation mode, PC-Interface makes your personal computer behave like either a VT100 or a VT220 terminal, allowing you to conduct a standard UNIX session.

Format: **EM2** [*hostname*]

Remarks: When you issue the EM2 command with no argument, you are presented with a choice of UNIX host connection paths over which to establish a terminal emulation session. You have the option to specify a host on the EM2 command line, bypassing the host-selection menu.

Once your terminal emulation connection has been established, you log in to your UNIX host and conduct your UNIX session just as you would over a standard VT220 terminal.

When you establish a terminal emulation session (or at any time while such a session is active) you can modify a number of parameters that affect how your personal computer communicates with the host.

JOBS Command

Purpose: The JOBS command displays the job table of ON-initiated UNIX tasks, clears it of completed tasks, and reattaches the DOS console to detached jobs.

Format: **JOBS**
JOBS -
JOBS *%[jobid]*

Remarks: The first form of the command displays the current job table. The job table is displayed in this form:

```
JOB   HOST       STATE    EXIT STATUS  COMMAND
[ 1]  d: host1   Running                unixcommand1
[ 2]  e: host2             exit (0)     unixcommand1
```

The second form of the command clears completed entries from the job table and removes any temporary files on the host associated with each task. When a job is cleared, JOBS discards any standard output being saved for viewing at the time of reattachment.

The third form of the command reattaches the DOS console to the specified detached job. If you enter:

```
JOBS %
```

without an ID number, you are reattached to the lowest numbered task in the job table.

When you reattach to a running task, the non-redirected standard output of the UNIX process is displayed on your screen. When you reattach to a completed task, the entire non-redirected standard output from the beginning

of the process is displayed. You can deattach from and reattach to a task any number of times while it is running. When you reattach after the task has completed, you see the entire output from the beginning of the job.

Detached jobs can be halted and cleared from the job table by reattaching, pressing CTRL-C or CTRL-BREAK, then responding `a` to the resulting prompt. To halt the UNIX process, use KILL.

The JOBS command is implemented as a .EXE file. This means that to execute a UNIX command called **jobs**, you must enter:

```
on - jobs
```

rather than linking JOBS.EXE to ON.EXE. If you want to change the name of the JOBS command, you can use the ONNAME environment variable.

See the entry for ON in this chapter for more information.

KILL Command

Purpose: The KILL command is used to send signals to UNIX processes, including those initiated using ON.

Format: **KILL** [*-signal*] *%jobid ...*
KILL [*-signal*] [*hostname|driveletter:*] *pid ...*

Remarks: The PC-Interface KILL command is similar to the UNIX **kill** command. *signal* is the UNIX signal to be sent to the UNIX process. *jobid* is the job control ID number of an ON-initiated detached job as reported in the job table. *pid* is a UNIX task's process ID number as determined with the UNIX **ps** command. If *pid* is negative, or *%jobid* is specified, the entire process group is killed. If no host name or drive letter is specified, the host associated with the current drive is used.

When the first form of the command is used, the job table entry for the specified job is cleared. When the second form is used, the signal is sent to the process, but the job table entry is not cleared.

If the ON-initiated detached job spawned child processes, and you used a form of KILL that does not kill the entire process group, then you must kill the child processes one at at time by ID number. To determine the process ID numbers, execute the command:

```
ON - ps -ef
```

Kill each process as follows:

```
kill pid
```

The default signal sent by KILL is 15. If this fails to kill the process, use 9 as the signal, as follows:

```
kill -9 pid
```

See **kill** and **signal** in your UNIX reference manual for more information.

The KILL command is implemented as a .EXE file. This means that to execute a UNIX command called **kill**, you must enter:

```
on - kill
```

rather than linking KILL.EXE to ON.EXE. If you want to change the name of the KILL command, you can use the ONNAME environment variable.

See the entry for ON in this chapter for more information.

LOGIN Command

Purpose: The LOGIN command initiates a PC-Interface host file services session by making a host available as a virtual drive.

Format: **LOGIN** [/D:*driveletter*] [*hostname* [*username* [*password*]]]

Remarks: When you issue the LOGIN command, you are presented with a choice of UNIX hosts and connection paths over which to establish a host file services session. After you choose a host connection path, you are prompted for your UNIX user name and password. If you select an RS-232 connection path, you are prompted to select the correct baud rate for your machine before entering your user name and password.

Each time you establish a host file services session, PC-Interface displays the drive letter associated with the UNIX file system on the selected host. For example:

hostname **File System Available as Drive D:**

By default, PC-Interface makes the selected host available to you as the next available drive. This is usually drive C: if your PC has only two diskette drives; drive D: if your PC has a fixed drive; or some other drive letter if you have already established host file sessions on other hosts.

You have the option of specifying the drive you want associated with a particular host by specifying the drive as part of the LOGIN command, as follows:

```
login /d:driveletter
```

If you try to log into a given host through more than one drive, PC-Interface displays an error message.

When you log in over a LAN connection path, you can specify the host you want as part of the LOGIN command, bypassing the host-selection menu. You can include your user name or your user name and password as part of the command line. PC-Interface prompts you for user name and password if you omit them.

When you log in over an RS-232 connection path, you do not specify a host or user name from the command line. Instead, you type:

```
login comn
```

where *n* represents the number (1 or 2) of the desired serial port. PC-Interface prompts you for the baud rate and then displays the terminal emulation window, where you enter your name and password.

Once PC-Interface has established the host connection, LOGIN indicates the virtual drive on which the UNIX file system is available during the host file services session.

LOGOUT Command

Purpose: The LOGOUT command terminates PC-Interface host file services sessions.

Format: **LOGOUT** [*hostname*|*driveletter*:]

Remarks: Entering LOGOUT with no parameters logs you out of all current host file services sessions.

When you specify the *hostname* parameter, PC-Interface logs out of the file services session on the specified host. Using the *driveletter* parameter logs you out of the file services session on the host associated with that drive letter.

ON Command

Purpose: The ON command executes UNIX commands on a specified host computer.

Format: **ON** *driveletter*: | *hostname* | - *unixcommand* [**&**]

Remarks: The ON command allows you to execute noninteractive UNIX commands from the DOS environment. The UNIX commands are executed under the Bourne shell. The current directory of the UNIX process is the same as the current directory of the selected virtual drive. Any screen output (standard output) produced by a command is displayed on the DOS screen unless it is explicitly piped or redirected elsewhere. Screen output returned to the DOS environment by ON is automatically converted to DOS format. That is, UNIX-style text with lines separated only by line-feed characters is converted to DOS-style text by adding carriage returns between lines.

There are two forms of the ON command. In the first form, you type ON followed by a drive letter, host name, or placeholder (-). The drive letter or host name lets you select a specific host on which the UNIX command is executed; the placeholder signifies that the UNIX command is executed on the current virtual drive. For example:

```
on host1 cal
```

runs the UNIX **cal** command on the host called **host1** and displays the screen output produced by **cal** on your DOS screen.[1] To run the same command on virtual drive D:, type:

```
on d: cal
```

To run the same command on the current host, type:

```
on - cal
```

With this form of the ON command, you can issue any valid UNIX command, including any valid options. To prevent ambiguity in interpreting the command, the UNIX command may be enclosed in double quotes (" "). For example, the command:

```
on host1 "date > temp"
```

executes the UNIX **date** command and places the output in the file **temp** in the current working directory on the host. However, the command:

```
on host1 date > temp
```

places the output of the **date** command in the file TEMP in the current working directory of the current drive.

The UNIX command name can be a name that is not a valid DOS name (for example, **DOIT.temp**). Do not use DOS mapped names with the ON command.

[1]The host name must be typed exactly as shown on the login map. Note that if you are connected to a host file services session over an RS-232 connection path, you use the name of the serial port (**com1** or **com2**), not the host name.

The second form of the ON command lets you omit the word "ON" from the command line and type only the UNIX command name. This form of the ON command only runs on the current virtual drive. To use this form of ON, you must first copy or link the ON file (ON.EXE) to the name of the UNIX command you wish to execute from the DOS environment. Include the DOS extension .EXE in the name of the UNIX command.

Once ON.EXE is copied or linked and renamed to resemble the UNIX command, you can run the UNIX command directly from the DOS prompt. For example:

```
copy on.exe cal.exe
cal
```

When you use this form of the ON command, the UNIX command name must be in lowercase (which is conventional for commands in UNIX) and must conform in other respects to the DOS rules for file names.

You cannot initiate multiple UNIX commands with a single ON command containing semicolons. Use separate ON commands instead.

Search Path And Environment Considerations

The UNIX environment must be able to find the UNIX command named on the ON command line.

The default UNIX search path used by ON includes, in order, the current working directory of the virtual drive, **/bin**, and **/usr/bin**. To change this path, set the DOS environment variable UPATH using the same syntax you use with the UNIX Bourne shell PATH

environment variable, including colons and forward slashes. For example:

```
set upath=:/bin:/usr/bin:/u/joe/bin
```

The value of the DOS UPATH variable is assigned to the UNIX PATH environment variable when ON is run.

To set any other UNIX environment variables needed by UNIX commands invoked via ON, first set their values in the DOS environment (using the DOS SET command). Then set the DOS EXPORT environment variable equal to the list of environment variables to be communicated to UNIX. For example:

```
set home=/usr/paula
set tz=PST8PDT
set lpdest=laser2
set export=home tz lpdest
```

Since DOS environment variable names are always stored as uppercase, ON converts all environment variable names to uppercase when they are placed in the UNIX environment. Thus, only all uppercase variable names can be exported to the UNIX environment. However, the *values* of the DOS variables are exported literally.

Pipes And Redirection

You can use pipes and redirection on UNIX, DOS, or a combination of the two. The standard characters "<" and ">" redirect DOS input and output. The character "|" pipes DOS output. The characters "{" and "}" redirect UNIX input or output. Either "!" or "^" pipes UNIX output.

Standard Input, Standard Output, and Standard Error

Input to and output from the UNIX process is transferred between UNIX and DOS via temporary files created in the directory **/tmp** on the host.

If standard input on DOS is redirected (using "<"), the standard input is copied to a temporary file on the host. Standard input to the UNIX process is then redirected from this temporary file. You can use the DOS <CON redirection argument to redirect input from the keyboard. Since the standard input file is created and closed before the UNIX process is run, you cannot enter input to the UNIX process interactively.

In the absence of any output redirection on UNIX, both standard output and standard error from the UNIX process are redirected to a file named *pid*.**chm**, where *pid* is the process ID number of the UNIX process. The contents of this file are copied to standard output on DOS as it is created on UNIX.

Upon completion of tasks running in the DOS foreground, ON deletes all temporary files created on the host.

If you have detached an ON-initiated UNIX task, the temporary standard output file is saved until you reattach to the task. At that time, the output is displayed on your screen.

Breaking Out of ON

Pressing CTRL-BREAK or CTRL-C while ON is running causes the following prompt to be displayed:

```
a - abort, c - continue, d - detach:
```

Typing `a` stops ON in the DOS environment and removes the task from the DOS job table. Typing `c` causes ON to continue from the point of interruption. Typing `d` detaches the task as described below.

Detached Jobs

The UNIX program may be run as a detached job by terminating the command line with an ampersand (&), or by responding `d` to the **`abort, continue, detach`** prompt described above.

When a task is detached, ON reports both an ON job ID number and the process ID number of the UNIX process. ON then exits immediately to DOS.

When ON is used in this way, temporary files are not deleted automatically. If you have not redirected the standard output of the UNIX command, it is saved for review at the time of reattachment.

If you detach ON commands containing DOS pipes or redirection, the results may not be what you expect.

PCICONF Command

Purpose: The PCICONF command changes the way the bridge responds to certain system calls, so that DOS programs that otherwise could not operate on a virtual drive are able to do so.

Format: **PCICONF SET NBS ON|OFF**

PCICONF SET ND REMOTE|LOCAL

PCICONF DISPLAY|RETURN ND|NBS

Remarks: NBS ON, the default, causes the PC-Interface bridge to answer yes when a DOS program asks whether NetBIOS support is available. Note that the bridge does not actually support NetBIOS; it only responds this way to the initial query from the application.

There are no programs known to be adversely affected by this configuration. However, if you find that a DOS program behaves in an unexpected way that you think may be attributable to this bridge configuration, you may want to turn off NBS, as follows:

```
pciconf set nbs off
```

ND stands for "network drive." ND REMOTE, the default, means that virtual drives look (to DOS programs) like remote drives. This is generally the correct behavior when using network versions of DOS applications. However, certain DOS applications (usually, non-network versions) can only recognize local drives. When using such an application, if ND is set to

REMOTE, the program may not be able to see a virtual drive. In this case, you should execute the following command so that the virtual drive is treated as a local drive:

```
pciconf set nd local
```

PCICONF DISPLAY shows the status of the indicated parameter. For example, if you have not changed the default settings:

```
pciconf display nbs
```

returns:

```
nbs: on
```

PCICONF RETURN (NBS or ND) returns an error code. This form of the PCICONF command is used in batch scripts. The command returns as follows:

```
pciconf return nbs
```

returns 0 if NBS is OFF and 1 if NBS is ON.

```
pciconf return nd
```

returns 0 if ND is LOCAL and 1 if ND is REMOTE.

PCIINIT Command

Purpose: The PCIINIT command initializes PC-Interface software.

Format: **PCIINIT** [/B*hostname*\|/B*address*] [/E]

Remarks: When you install PC-Interface using the INSTALL program, PCIINIT is included in the AUTOEXEC.BAT file on your PC-Interface working diskette so that initialization occurs automatically when you start your personal computer.

PCIINIT searches the file \ETC\HOSTS for the alias "localhost" and uses the network address associated with that name.

The /B option is used to specify the host from which PC-Interface requests a map. Do not use this parameter unless you are instructed to do so by your system administrator.

The /E parameter echoes the current network address setting.

PRINTER Command

Purpose: The PRINTER command controls whether PC-Interface printing is directed to a local printer attached to your personal computer or to a remote UNIX printer.

Format: **PRINTER** [**LPT***n*] **LOCAL**

PRINTER [**LPT***n*] *driveletter*: | *hostname* | - [*printprogram*] [**/X0** | **/X1**] [**/T**[*timeout*]]

PRINTER [**LPT***n*] [**/P**|**/D**|**/R**|**/X0**|**/X1**] [**/T**[*timeout*]]

Remarks: **Local printing.** The LOCAL parameter specifies that print requests are sent to a local printer attached to your personal computer. This is the normal DOS case. Local print requests can be issued in any of the following ways:

- Using the DOS PRINT command.
- Using the Shift and print-screen (PRT SC)[1] keys.
- Using the CTRL and PRT SC (or CTRL and P) keys together to print continuous screen contents.
- Using the DOS COPY command and specifying the local printer device (PRN) as the target.
- Invoking the printer from an application program.

[1] If you have an enhanced keyboard, use the print-screen key without the Shift key.

Remote printing. To invoke a print job on a remote UNIX host, you use the same methods as for local printing, except that you cannot use the DOS PRINT command. However, before starting a print job, you need to set up the print streams to tell the system where, how, and when you want the job to print. The following paragraphs describe how to use the second form of the PRINTER command to set up the print streams.

Setting up the print stream involves specifying a print stream, a host, and a printing program, and indicating when you want the job to be processed. Defaults are available for most of these parameters.

LPT*n* is a print stream (either LPT1, LPT2, or LPT3). If you don't specify a print stream, the default is LPT1.

Either *driveletter*, *hostname*, or - (hyphen) is used to specify the host. For example:

```
printer lpt2 d:
```

causes all subsequent print requests directed to LPT2 to be sent to the printer associated with the host attached to drive D:, while:

```
printer frog
```

sends all print requests sent to LPT1 (since no print stream was specified) to the printer associated with host **frog**.[1]

[1]Note that you cannot use a host named **local**.

If your current drive is a virtual drive, you can use a hyphen (-) to represent the host. In this case, your printing is directed to whichever host is attached to the current drive.[1] For example:

```
printer lpt2 -
```

sends all LPT2 print requests to the printer on the host associated with the current drive.

The *printprogram* parameter represents a UNIX command string to be associated with the specified print stream.[2] For example:

```
printer lpt3 - troff
```

causes all subsequent print requests directed to LPT3 to be sent to the UNIX **troff** program on the current host and printed on the printer associated with the **troff** command.

The *printprogram* parameter can include command options and pipes and redirection. When the command string contains more than one word, it must be enclosed in quotes. For example:

```
printer lpt2 -  "tbl | troff"
```

[1] Note that even if you change the current host by changing your working drive, printing remains directed to the original host until you enter a new PRINTER command to change it.

[2] Note that when you use the *printprogram* parameter, the actual printer used may be determined by the print command invoked, rather than by the specified host.

causes output directed to LPT2 to be sent to the UNIX preprocessing program **tbl** and then piped to the UNIX print command **troff**, before being printed on the printer associated with **troff**.

If you omit the *printprogram* parameter for a particular print stream, print requests for that print stream are sent to the system default print command and are output on the default printer.

Controlling When Your Jobs Print

The /T option sets a timeout period. When you do not set this option, or when you set /T0, PC-Interface does not process your print requests until you close the print stream by exiting the DOS application. To force jobs to print before you exit the application, use the /T option to set a timeout period. When this period has elapsed with no characters having been sent to the printer, the print stream closes and your job prints. For example:

```
printer lpt2 frog /t15
```

means that each time 15 seconds go by with no more characters being sent to the printer, whatever characters have already been sent are printed.

You can specify *timeout* values from 1 to 3600.[1] /T with no *timeout* value defaults to 45 seconds.

The /X option controls whether or not buffered print requests are processed when the print stream is closed. The default is

[1] When using PC-Interface over an RS-232 line, you may need to specify a longer timeout period than you would when using a LAN. In general, if your listing is broken up into multiple parts rather than being printed as one job, you may need to increase the timeout period.

/X1, which means jobs are processed when the print stream is closed (subject to timeout restrictions). To stop jobs from being processed until explicitly requested, set /X0 *and* /T0. For example:

```
printer lpt3 frog /x0 /t0
```

means that subsequent jobs sent to LPT3 are held until you explicitly request that they be printed.

The /P option causes any buffered printer output for the specified print stream to be printed. The /D option causes any buffered printer output for the specified print stream to be deleted. When the print stream specifier is omitted on the PRINTER command with the /P or /D option, the print or delete action applies to all print streams.

Modifying Print Stream Settings

The third form of the PRINTER command is used to modify the existing settings of the print streams. For example, if you have issued the command:

```
printer lpt3 - troff
```

and later want to add a timeout period of five seconds, just type:

```
printer lpt3 /t5
```

This command only adds the timer interrupt, leaving the rest of the print stream settings as they were.

The /R option resets the *printprogram* for the specified print stream to the system default. When the print stream specifier is omitted on the PRINTER command with the /R option, only LPT1 is reset.

The PRINTER command issued with no options displays the current PRINTER setting and options.

UDIR Command

Purpose: The UDIR command lists the contents of directories on a virtual drive in a combination of UNIX and DOS styles, displaying both UNIX and mapped names, owner, file permissions, size, and modification date and time for each file or directory.

Format: **UDIR** [-h] [*driveletter*:][*path*][*filename*]

Remarks: The *driveletter* parameter specifies a virtual drive.

The *filename* parameter can specify either a file or directory. Directories are indicated in the UDIR display with both an initial "d" in the permission field and the <DIR> notation.

File and directory names are entered in the UNIX form. The UDIR display shows all the file and directory names in both their UNIX and mapped-name form. You can also use the global characters * and ? in the parameters; however, UDIR interprets these as DOS does, not as UNIX does.

The -h option must be specified exactly as shown (this option is case and switch sensitive). Specifying this option displays "hidden" files (files that start with a .).

UNIX2DOS Command

Purpose: The UNIX2DOS command converts a text file from UNIX format—lines ending in a new-line (ASCII line-feed)—to DOS format—lines ending in a carriage-return and line-feed sequence and end-of-file marked with a CTRL-Z (octal 018). The command can be issued from either the DOS operating system or the UNIX operating system.

Format: **UNIX2DOS** [/B|/U|/L][/F] [[*path*]*filename*] [[*path*]*filename*]

Remarks: The first file specified is the source file. The second file specified is the target file. The source file and the target file must not have the same name. When neither file parameter is specified, UNIX2DOS reads from standard input and writes to standard output. When only one file parameter is specified, that file is considered the source file, and the output is written to standard output.

If the source file is already in DOS style, UNIX2DOS does not alter the format of the file.

The /B, /U, and /L parameters are mutually exclusive. These parameters, which can be specified either before or after the file names, have the following meanings:

/B Preserves 8-bit (binary) character representations. When this parameter is not specified, UNIX2DOS truncates all characters to 7-bit ASCII representation.

/U Converts text to uppercase.

/L Converts text to lowercase.

Note: When neither /U nor /L is specified, the case of each text character is unchanged.

UNIX2DOS normally removes all extra carriage-return characters preceding the end-of-line character during the end-of-line conversion. The /F parameter prevents the removal of these extra carriage returns and forces the translation of each line-feed into a carriage-return and line-feed pair, even when a carriage return is already present. Although not normally used, this parameter preserves unusual end-of-line sequences when complete file invertibility is required.

When issued from the UNIX operating system, the /B, /U, /L, and /F parameters must be specified with the UNIX hyphen (-) rather than the DOS slash (/). From the DOS operating system, use either the slash or the hyphen.

UNIX2DOS can be used in combination with DOS and UNIX pipes and redirection. For example, the command:

```
unix2dos phonenos | sort > phone
```

converts the UNIX file **phonenos**, sorts the converted text, and writes it to the DOS file PHONE. Note that you cannot redirect a converted file to itself.

Installing PC-Interface

The first part of this appendix discusses the hardware you must have installed in your personal computer before you can install PC-Interface. The second part of this appendix describes how to combine the PC-Interface files from the distribution diskette (or diskettes) with your DOS files to create a working PC-Interface diskette or to install PC-Interface on your personal computer with a fixed disk.

Before You Install PC-Interface

Hardware Requirements

You can install PC-Interface to run over a local area network (LAN), over an RS-232 line, or over both.

LAN Installation–To install PC-Interface to run over a LAN, you must first install a network interface board in your personal computer. PC-Interface supports a wide range of popular network interface boards. See the *Release Notes* accompanying this manual for a list of network interface boards supported by your release of PC-Interface. You can only use one network interface board at a time, although you *can* use an asynchronous communications adapter together with a network interface board. Use the instructions provided in the board manufacturer's guide to install the network interface board. If test programs are supplied, you may use them to verify the operation of the board.

Many network interface boards can be used with either built-in transceivers or external transceivers. Either configuration works with PC-Interface. Just make sure that the board is jumpered correctly for the configuration you are using and that cables and connectors are correctly attached. Refer to the installation instructions supplied with your network board.

The factory default jumper configurations are normally the correct ones to use. However, if you have other hardware that conflicts with the default settings of your network boards, you may need to reconfigure the boards and the drivers. Configurable items may include interrupt level, I/O base address, and memory address. To reconfigure a board, refer to the manufacturer's installation instructions. To reconfigure one of the drivers supplied with PC-Interface, see "Configuring Your Driver" at the end of this chapter.

RS-232 Installation–To install PC-Interface to run over an RS-232 line, you must have an asynchronous communications adapter installed in your personal computer. Follow the instructions in the technical reference manual for your personal computer or in the product manual to install the adapter.

Installation

Your PC-Interface distribution diskette or diskettes are in a pocket at the back of this manual. The diskettes contain the programs necessary to establish a host file services session, the program you need for terminal emulation, and the PC-Interface utilities. The diskettes do not contain the DOS programs needed to start the system and load the PC-Interface programs. You must have those programs.

You should back up your PC-Interface distribution diskette or diskettes before using them for installation. However, please note that PC-Interface is copy-protected. The distribution diskettes accompanying this manual are for your use only.

This appendix describes, step by step, how to install PC-Interface on your fixed disk or create a working PC-Interface

diskette. These steps are described separately for three different personal computer configurations:

- A personal computer with a fixed disk drive.
- A personal computer with two double-sided diskette drives.
- A personal computer with one double-sided diskette drive.

When you want to install PC-Interface on a diskette, you must select a basic installation rather than a complete installation. The basic installation provides all the functionality necessary to initialize PC-Interface and start a host file services session and a terminal emulation session.

After making the working diskette, you can either copy the remaining PC-Interface files onto a second diskette, or you can copy these files onto the virtual drive.

Refer to the section of the appendix that corresponds to your personal computer configuration. We suggest that you read through the appropriate section before you begin the installation procedure.

Preparing to Install PC-Interface

To install PC-Interface, you need the following information.

- The type of network interface board installed in your personal computer.
- If installing for a LAN, your personal computer's internet address and name.

If you don't have this information, ask your system administrator for help.

Fixed-Disk Drive

Follow these instructions to install PC-Interface on your fixed disk. If you prefer to use PC-Interface on diskette, refer to the section with the diskette configuration that matches your system.

We assume that you have set up a DOS partition on your fixed disk and have copied DOS into that partition.

For this procedure, you need the following:

- A personal computer with at least one double-sided diskette drive and a fixed disk.
- Your PC-Interface distribution diskette or diskettes.

You may need to change diskettes during the INSTALL procedure. A message appears on your screen when a change is needed.

Installing the PC-Interface Files

1. Insert the PC-Interface distribution diskette into the disk drive.
2. Type:

   ```
   a:
   ```

   Press Enter (↵).
3. Type:

   ```
   install
   ```

Press Enter (↵).

A copyright message appears, followed by the prompt:

```
On which drive do you want to install
   PC-Interface ? [C:]
```

Press Enter (↵). (If you received the message "Bad command or filename," you probably missed step 2. Change the current drive to A: and repeat step 3.)

4. A list of network types is displayed, followed by the prompt:

```
Enter the number corresponding
   to your network type:
```

Enter the number corresponding to your networking environment and network interface board. (Select the number corresponding to RS-232 Interface Only if your personal computer does not have a LAN connection.)

5. When prompted, insert the PC-Interface DRIVERS diskette and press Enter.

6. The following message displays:

```
Do you want to install the complete
   PC-Interface 3.0 system? [y]
```

Press Enter (↵).

7. If you are installing for RS-232 only, skip this step. If you are installing for a LAN, the following message appears:[1]

```
Enter the internet address of your PC
   (in the form A.B.C.D):
```

   Type your internet address and press Enter (↵).

   The following message appears:[2]

```
Enter the host name of your PC:
```

   Type your personal computer's host name and press Enter (↵).

   The system now creates an \ETC\HOSTS file.

8. INSTALL displays the following message:

```
Do you want to install PC-Interface
   utilities in a subdirectory?[y]
```

   Press Enter (↵).

9. INSTALL prompts for a path name for the subdirectory, as follows:

```
Enter path: C:\
```

10. Type the full path name of the subdirectory in which you want PC-Interface utilities installed and press Enter (↵).

---

[1] If you are installing for a LAN and this prompt is not displayed, it means that you already have a HOSTS file on your system. For example, if you have Excelan TCP/IP installed on your system, you already have a HOSTS file, and you will not see this prompt.

[2] If this prompt is not displayed, it means that you already have a HOSTS file on your system.

11. When prompted, insert the PC-Interface INSTALL diskette and press Enter.

    The INSTALL program prints out the following messages:

    ```
    Installing complete PC-Interface 3.0 system.

    Copying files. Please wait.
    ```

    If you have more than one PC-Interface distribution diskette, you may need to change diskettes during the INSTALL procedure. A message appears on your screen when a change is needed. For example:

    ```
    Please insert PC-Interface 3.0 diskette "Drivers"
    and press RETURN.
    ```

12. When the INSTALL program finishes, the following message displays:

    ```
    PC-Interface 3.0 installation complete.
    ```

    Remove the PC-Interface distribution diskette from drive A: and store it in a safe place.

13. During installation, INSTALL copied the contents of any existing AUTOEXEC.BAT file to AE.BAT and appended the PCIINIT command to the end of AUTOEXEC.BAT. If AUTOEXEC.BAT did not exist, it was created. In addition, INSTALL copied the contents of any existing CONFIG.SYS file to CF.SYS and appended appropriate DEVICE= commands at the end of CONFIG.SYS. If CONFIG.SYS did not exist, it was created. You can delete AE.BAT and CF.SYS if you want.

PC-Interface is ready to use. To proceed with a PC-Interface session, hold down the CTRL and ALT keys and press the

DEL key to reset your personal computer and initialize PC-Interface.

Two-Diskette Drive System

Follow these instructions to format a diskette with the DOS operating system files and then install the PC-Interface files on it in order to create a working diskette. (For further information on the commands used in this procedure, refer to your DOS manual).

For this procedure, you need the following:

- A personal computer with two double-sided diskette drives.
- Your PC-Interface distribution diskette or diskettes.
- One blank double-sided, double-density diskette to be used as a working diskette. If you plan to use the PC-Interface utilities from a diskette, you also need a second blank double-sided, double-density diskette.
- Your DOS system diskette.

Preparing A Working Diskette

1. Label one blank diskette PC-INTERFACE WORKING.
2. Insert your DOS system diskette into diskette drive A:.
3. Switch on your system unit if you have not already done so. If the system unit is already on, press and hold the

CTRL and ALT keys; then press the DEL key (system reset). Release all three keys.

4. When DOS prompts you, enter today's date and the time.

   A message similar to the following appears:

```
Microsoft MS-DOS version 3.10
Copyright 1981,82,83 Microsoft Corp.
A>
```

5. Type:

```
format b: /s
```

   Press Enter (↵).

   The following message appears:

```
Insert new diskette for drive B:
and strike any key when ready
```

6. Insert the blank WORKING diskette into drive B:.

   Press any key. The system now formats the WORKING diskette. When the FORMAT procedure is completed, the following message appears:

```
Formatting. . .Format complete

  xxxxxx bytes total disk space
   xxxxx bytes used by system
  xxxxxx bytes available on disk

Format another (Y/N)?
```

7. Type n and press Enter (↵).

8. If you want to use PC-Interface utilities from a diskette, you need a second formatted diskette; however, it does not need to be a bootable diskette. To format the second diskette, insert it into drive B: (after removing the diskette already in there), close the latch, and type:

   ```
   format b:
   ```

   Press Enter (↵).

9. When DOS informs you that formatting is completed, remove the diskette from the B: drive. Store the DOS system diskette in a safe place. Label the second (non-bootable) formatted diskette PC-INTERFACE UTILITIES and then set it aside. You will copy the utility programs onto it later.

10. Remove the DOS system diskette from drive A: and store it in a safe place.

Installing the PC-Interface Files

You may need to change diskettes during the INSTALL procedure described below. A message appears on your screen when a change is needed.

1. Insert the PC-Interface distribution diskette into drive A:. Insert the WORKING diskette into drive B:.

2. Type:

   ```
   a:
   ```

   Press Enter (↵).

3. Type:

```
install
```

   Press Enter (↵).

   A copyright message appears, followed by the prompt:

```
On which drive do you want to install
   PC-Interface 3.0 ? [C:]
```

   (If you received the message "Bad command or filename," you probably missed step 2. Change the current drive to B: and repeat step 3.)

4. Type:

```
b:
```

   Press Enter (↵).

5. A list of network types is displayed, followed by the prompt:

```
Enter the number corresponding
   to your network type:
```

   Enter the appropriate number. (Select the number corresponding to RS-232 Interface Only if your personal computer does not have a LAN connection.)

6. When prompted, insert the PC-Interface DRIVERS diskette and press Enter.

7. The INSTALL program checks the amount of space available on the WORKING diskette. When there is not enough room to install the complete PC-Interface system,

the INSTALL program informs you of this and displays the following message:

```
Do you want to install a minimal
   PC-Interface 3.0 system? [y]
```

Press Enter.

8. If you are installing for RS-232 only, skip this step. If you are installing for a LAN, the following message appears:[1]

```
Enter the internet address of your PC
   (in the form A.B.C.D):
```

Type your internet address and press Enter (↵).

9. The following message appears:[2]

```
Enter the host name of your PC:
```

Type your personal computer's host name and press Enter (↵).

The INSTALL program now creates an \ETC\HOSTS file, an AUTOEXEC.BAT file, and a CONFIG.SYS file on the WORKING diskette.

10. When the INSTALL program prompts you, remove the diskette in drive A:, replace it with the INSTALL diskette, and press Enter.

---

[1] If you are installing for a LAN and this prompt is not displayed, it means that you already have a HOSTS file on your system. For example, if you have Excelan TCP/IP installed on your system, you already have a HOSTS file, and you will not see this (or the next) prompt.

[2] If this prompt is not displayed, it means that you already have a HOSTS file on your system.

11. Next the system prompts you to insert the DRIVERS diskette. Press Enter.

12. When the INSTALL program finishes, the following message displays:

```
PC-Interface 3.0 installation complete.
```

    Store the PC-Interface distribution diskettes in a safe place.

To proceed with a PC-Interface session, insert your WORKING diskette in drive A:, hold down the CTRL and ALT keys and presss the DEL key to reset your personal computer and initialize PC-Interface.

The PC-Interface Utility Programs

The use of the PC-Interface utility programs is described in Chapter 6, Command Summary. Following is a list of these programs:

DOS2UNIX.EXE
UNIX2DOS.EXE
ON.EXE
JOBS.EXE
KILL.EXE
PRINTER.EXE
UDIR.EXE
PCICONF.EXE
DOSWHAT.EXE

These programs are all delivered on the PC-Interface distribution diskette that you used to create your WORKING diskette. Because of space limitations, you cannot put all of

these programs on the same diskette that you use to start PC-Interface.[1]

There are two places you can put the PC-Interface utilities so that they are readily available. The best place to put them is on the virtual drive. When you do this, the utilities are available when you start a host file service session. (Be sure to include the virtual drive in your PATH.) Instructions for putting the PC-Interface utilities on a virtual drive are provided in Chapter 5, in the section entitled "Putting PC-Interface Files On A Virtual Drive."

The other place you can put the utilities is on a second diskette. To put the PC-Interface utility programs on a diskette, insert the PC-Interface distribution diskette into drive A:, insert the blank formatted diskette labeled PC-INTERFACE UTILITIES into drive B:, and copy the files listed above one at a time by typing:

```
copy a:filename b:
```

When you finish this procedure, remove the PC-Interface distribution diskette from drive A: and store it in a safe place.

If you want to proceed with a PC-Interface session, insert the PC-Interface WORKING diskette into drive A: and hold down the CTRL and ALT keys and press the DEL key to reset your personal computer and initialize PC-Interface.

[1] This is true if you are using a double-density (360K-byte) diskette. However, if you are using a high-density (720K-byte) diskette, you *can* fit all the PC-Interface program files on a single diskette.

One-Diskette Drive System

Follow these instructions to format a diskette with the DOS operating system files and then add the PC-Interface files in order to create a working diskette. (For further information on the commands used in this procedure, refer to your DOS manual).

For this procedure, you need the following:

- A personal computer with one double-sided diskette drive.
- Your PC-Interface distribution diskette or diskettes.
- One blank double-sided, double-density diskette to be used as a working diskette. If you plan to use the PC-Interface utilities from a diskette, you also need a second blank double-sided, double-density diskette.
- Your DOS system diskette.

Preparing A Working Diskette

1. Label one blank diskette PC-INTERFACE WORKING.
2. Insert your DOS system diskette into the diskette drive.
3. Switch on your system unit if you have not already done so. If the system unit is already on, press and hold the CTRL and ALT keys; then press the DEL key (system reset). Release all three keys.
4. When DOS prompts you, enter today's date and the time.

A message similar to the following appears:

```
Microsoft MS-DOS version 3.10
Copyright 1981,82,83 Microsoft Corp.
A>
```

5. Type:

   ```
   format  a: /s
   ```

   Press Enter (↵).

   The following message appears:

   ```
   Insert new diskette for drive A:
   and strike ENTER when ready
   ```

6. When the drive A: "in use" light is off, remove the DOS diskette from drive A: and insert the WORKING diskette into the drive.

   Press Enter (↵).

   When the FORMAT procedure is completed, the following message appears:

   ```
   Formatting. . .Format complete

     xxxxxx bytes total disk space
      xxxxx bytes used by system
     xxxxxx bytes available on disk

   Format another (Y/N)?
   ```

7. Type n and press Enter (↵).

8. Remove the WORKING diskette from the drive.

9. If you want to use PC-Interface utilities from a diskette, you need a second formatted diskette; however, it does not need to be a bootable diskette. Insert your DOS diskette into drive A: and type:

   ```
   format a:
   ```

   Press Enter (↵).

10. In response to the prompt, remove the DOS diskette from drive A: and insert the second blank diskette.

    Press Enter (↵).

11. When DOS informs you that formatting is completed, remove the diskette from the A: drive. Store the DOS system diskette in a safe place. Label the second (non-bootable) formatted diskette PC-INTERFACE UTILITIES and then set it aside. You will copy the utility programs onto it later.

Installing the PC-Interface Files

During the following procedure, files are read from the PC-Interface distribution diskette or diskettes to the WORKING diskette. With only one diskette drive, you must exchange diskettes during the procedure. A message appears on your screen each time an exchange is needed.

Messages read as if you have two drives instead of one. Think of the messages as referring to diskettes instead of drives. The PC-Interface distribution diskettes are the drive A: diskettes and the WORKING diskette is the "drive B:" diskette.

1. Insert the PC-Interface distribution diskette into the diskette drive.

2. Type:

```
install
```

Press Enter (↵).

A copyright message appears, followed by the prompt:

```
On which drive do you want to install
   PC-Interface 3.0? [C:]
```

Type:

```
b:
```

Press Enter (↵).

3. A list of network types is displayed, followed by the prompt:

```
Enter the number corresponding
   to your network type:
```

Enter the appropriate number. (Select the number corresponding to RS-232 Interface Only if your personal computer does not have a LAN connection.)

4. The following message is displayed:

```
Insert diskette for drive B:
Strike any key when ready
```

Remove the distribution diskette and insert the WORKING diskette. Press any key.

Continue switching diskettes as prompted. Remember that the diskettes for drive A: are always the PC-Interface distribution diskettes; the diskette for "drive B:" is the

WORKING diskette (the diskette onto which you are installing PC-Interface.

5. When INSTALL prompts you, insert the DRIVERS diskette in drive A: and press Enter.

6. The INSTALL program checks the amount of space available on the WORKING diskette. When there is not enough room to install the complete PC-Interface system, the INSTALL program informs you of this and displays the following message:

```
Do you want to install a minimal
   PC-Interface 3.0 system? [y]
```

   Press Enter.

7. If you are installing for RS-232 only, skip this step. If you are installing for a LAN, the following message appears.[1]

```
Enter the internet address of your PC
   (in the form A.B.C.D):
```

   Type your internet address and press Enter (↵).

8. The following message appears:[2]

```
Enter the host name of your PC:
```

   Type your personal computer's host name and press Enter (↵).

---

[1] If you are installing for a LAN but this prompt is not displayed, it means that you already have a HOSTS file on your system. For example, if you have Excelan TCP/IP installed on your system, you already have a HOSTS file, and you will not see this prompt.

[2] If this prompt is not displayed, it means that you already have a HOSTS file on your system.

The INSTALL program now creates an \ETC\HOSTS file, an AUTOEXEC.BAT file, and a CONFIG.SYS file on the WORKING diskette.

Then, the following message displays:

```
Installing required PC-Interface 3.0 system.
```

9. Insert the INSTALL diskette when prompted and presss Enter.

10. Next you are prompted to insert the PC-Interface DRIVERS diskette. Press Enter.

11. When the INSTALL program finishes, the following message displays:

```
PC-Interface 3.0 installation complete.
```

Store the PC-Interface distribution diskettes in a safe place.

At this point, the diskette in the drive is the WORKING diskette. When DOS prompts you for the drive A: diskette, just press Enter.

To proceed with a PC-Interface session, hold down the CTRL and ALT keys and press the DEL key to reset your personal computer and initialize PC-Interface.

The PC-Interface Utility Programs

The use of the PC-Interface utility programs is described in Chapter 6, Command Summary. Following is a list of these programs:

DOS2UNIX.EXE
UNIX2DOS.EXE
ON.EXE
JOBS.EXE
KILL.EXE
PRINTER.EXE
UDIR.EXE
PCICONF.EXE
DOSWHAT.EXE

These programs are all delivered on the PC-Interface distribution diskette that you used to create your WORKING diskette. Because of space limitations, you cannot put all of these programs on the same diskette that you use to start PC-Interface.[1]

There are two places you can put the PC-Interface utilities so that they are readily available. The best place to put them is on the virtual drive. When you do this, the utilities are available when you start a host file service session. (Be sure to include the virtual drive in your PATH.) Instructions for putting the PC-Interface utilities on a virtual drive are provided in Chapter 5, in the section entitled "Putting PC-Interface Files On A Virtual Drive."

The other place you can put the utilities is on a second diskette. With a single diskette drive personal computer, you must copy the files one at a time. Remember that the DOS prompts read as though you had two drives instead of one.

[1] This is true if you are using a double-density (360K-byte) diskette. However, if you are using a high-density (720K-byte) diskette, you *can* fit all the PC-Interface program files on a single diskette.

Use the following procedure to copy the PC-Interface utilities onto a diskette:

1. Insert the PC-Interface distribution diskette into drive A: and type:

   ```
   copy a:dos2unix b:
   ```

   When prompted, remove the distribution diskette and insert the the blank formatted diskette labeled PC-INTERFACE UTILITIES into the drive. Press any key.

2. Continue copying files, one at a time, until you have copied each of the files shown in the list above.

   When you finish this procedure, store the PC-Interface distribution diskette in a safe place.

   If you want to proceed with a PC-Interface session, insert the PC-Interface WORKING diskette into drive A: and hold down the CTRL and ALT keys and press the DEL key to reset your personal computer and initialize PC-Interface.

Configuring Your Driver

The INSTALL program adds a line similar to this to your CONFIG.SYS file:

```
device=\driver.drv
```

(where *driver* identifies the specific driver). The network software assumes that your network interface board is set to the factory set values. The *Release Notes* accompanying this manual contain a table showing these values.

If any of these values conflict with other hardware in your personal computer, you may have to reconfigure your network board and driver. To reconfigure the hardware, refer to the installation guide for your board. To configure the driver, add the appropriate command-line options from the following list to the DEVICE=*driver*.DRV line in your CONFIG.SYS file:

/I:*nn* Sets the I/O base address (in hexadecimal).

/X:*nn* Specifies the interrupt request level (IRQ) (in decimal).

/M:*nn* Specifies the memory base address in the personal computer where the network board memory resides (in hexadecimal; sometimes preceded by 0x).

You can use these options with either a slash (/) or a dash (-), and the options can be uppercase or lowercase. There must be a colon between the option and the number.

As an example, to configure the driver for the INTERLAN NI5210 network board so that the interrupt request level is 3 instead of 2, you would need the following CONFIG.SYS entry:

```
device=\ni5210.drv /x:3
```

As another example, to configure the driver for the Western Digital 8003E board to an I/O base address of hexadecimal 240, a shared memory base address of hexadecimal C8000, and an IRQ level of 3, you would need the following CONFIG.SYS entry:

```
device=\wd8003.drv /i:240 /m:C8000 /x:3
```

Messages

This appendix lists messages that PC-Interface programs can generate. These messages appear in alphabetical order to facilitate your locating the message you want.[1] Each message entry includes the exact wording of the message and a complete description including the cause and recommended user response.

This appendix does not list DOS messages. If your message does not appear in this appendix, check your DOS manual or contact your system administrator.

- ***hostname* is not a connected host**

Explanation: You specified a host to which you are not currently connected through host file services.

User Response: Either specify a logged-in host, or log in to the desired host and retry the operation.

- ***hostname* Not Available -- Try Another? (y or n):**

Explanation: The selected host cannot accommodate any additional users, or there was a communication failure between the personal computer and the host.

User Response: Type `y` or `n`. If you type `y` to retry the operation, you can select a different host to log in to. If you prefer, you can attempt to make the selection that caused the error message. If the problem persists, consult your system administrator. Typing `n` returns you to the DOS prompt.

[1] If the first word of the message is context-dependent (for example, a host name, a drive ID letter, or a UNIX command), the message is listed at the beginning of this appendix.

- ***driveletter*: is not a virtual drive**

Explanation: You specified a drive letter that is not a currently connected virtual drive, or you did not specify a virtual drive and the current drive is not a virtual drive.

User Response: Retry the operation, specifying a logged-in virtual drive.

- ***unixcommand*: access denied or file not found**

Explanation: A program file in the pipeline of the requested UNIX command was not located, or you do not have execute permission to use it. The field *unixcommand* is replaced with the name of the first requested UNIX command in the ON command line.

User Response: Check to make sure that you typed the name of each command correctly, that each command is located in a directory named in your UPATH setting, and that you have execute permission for each named command.

- ***unixcommand*: job table full**

Explanation: When you tried to run the *unixcommand* with ON, the job table was filled to capacity with the record of ON-initiated tasks.

User Response: Clear completed jobs from the job table with the JOBS command.

- **a - abort, c - continue, d - detach:**

Explanation: You pressed CTRL-C or CTRL-BREAK while ON was running in the DOS foreground.

User Response: Typing `a` stops the ON command. Typing `c` continues running it as before. Typing `d` places the requested UNIX command in the background.

- **Access Denied**

Explanation: The specified file is inaccessible for one of the following reasons:

- The user invoking it does not own the file.
- A directory on the path to the file is not executable.
- The file is locked (that is, it is already opened by an application that supports file sharing).

User Response: Check that you own the file. Check that the directory containing the file and all directories on the path to the file have execute permission. If the file is locked, wait until it is closed and the lock is removed.

- **An error has occurred while reading input file**

Explanation: When using DOS2UNIX or UNIX2DOS, you supplied a source file that is corrupted *or* there was an operating system error.

User Response: Check the integrity of the source file. If it does not appear to be corrupted, retry the operation. If the error occurs again, consult your system administrator.

- **An error has occurred while writing output file**

Explanation: This message can be caused by any of the following:

- Insufficient disk space to accommodate the target file.
- Incorrect UNIX permission mode for the target directory.
- An operating system error.

User Response: Verify whether there is room on your disk or diskette for the target file. If necessary, delete unneeded files. If there is adequate space for the target file, check the UNIX permission mode of the target directory and any directories on the path to the target directory. The target directory must be writable and executable, and directories on the path to the target directory must be executable. If these measures do not solve the problem, consult your system administrator.

- **Bad address class**

Explanation: The internet address specified by the /B option to PCIINIT is not valid.

User Response: Ask your system administrator for the correct address. Correct the address given in the /B option to PCIINIT and retry.

- **Bad argument**

Explanation: PCIINIT options are specified incorrectly.

User Response: Refer to the description of the PCIINIT command in Chapter 6 of this manual. Retry the command.

- **Bad print stream name**

Explanation: You gave an LPT name other than LPT1, LPT2, or LPT3 to the PRINTER command.

User Response: Retry the command with a valid LPT name.

- **BRIDGE.DRV not installed or incompatible version**

Explanation: BRIDGE.DRV was not loaded or was not initialized when you ran the ON program. (This is the same error as "Cannot find correct BRIDGE device.")

User Response: Check that your PC-Interface working disk contains the BRIDGE.DRV and CONFIG.SYS files. If either file is missing, reinstall PC-Interface. Check that the CONFIG.SYS file properly specifies the location of BRIDGE.DRV. Retry the operation.

- **Can't open input file**

Explanation: When using DOS2UNIX or UNIX2DOS, you specified a source file that does not exist or is not accessible.

User Response: Check that the source file exists and is named correctly. Check that the file is readable. Check that the directory containing the source file and any other directories on the path to the source file are executable.

- **Can't open output file**

Explanation: When using DOS2UNIX or UNIX2DOS, you specified a target file that cannot be opened for one of the following reasons:

- The directory containing it is not executable and writable.
- A directory on the path to the target directory is not executable.
- The target file already exists and is write-protected.

User Response: Check that the target directory is writable and executable. Check that all directories on the path to the target directory are executable. Check whether a file already exists with the name of the target file. If you intend to replace an existing file with a new file created by DOS2UNIX or UNIX2DOS, the existing file must have write permission. Instead of replacing an existing file, you can choose a different name for the DOS2UNIX or UNIX2DOS target file.

- **Cannot find correct BRIDGE device**

Explanation: The PC-Interface device driver was not successfully loaded.

User Response: Check that your PC-Interface working disk contains the BRIDGE.DRV and CONFIG.SYS files. If either file is missing, reinstall PC-Interface. Check that the CONFIG.SYS file properly specifies the location of BRIDGE.DRV. Retry the operation.

- **cannot open** *path*\HOSTS

Explanation: PCIINIT cannot find the HOSTS file. Either the file is not there, or there is no pointer to it in your AUTOEXEC.BAT file.

User Response: 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. The line should look like this:

```
set hosts=c:\etc
```

- **DOS 3.10 or later is required.**

Explanation: PC-Interface require DOS 3.1 or higher on your personal computer.

User Response: Upgrade your personal computer operating system to DOS 3.1 or higher.

- **DOS memory allocation error**

Explanation: An error occurred when ON attempted to reserve a portion of DOS memory to hold the request.

User Response: Reboot your personal computer and try again. If the problem persists, contact your PC-Interface supplier.

- **Error-Host did not respond**
  Retry? (Y or N)

Explanation: There are three reasons why this message may be displayed:

1. The host may have responded too slowly to a file services or terminal emulation request.

2. The host may go down between the time when you select that host from the host file services host-selection menu and the time your connection request is transmitted.

3. The host may have gone down and come back up while you were idle or working in emulation, thereby breaking your underlying host file services connection.

User Response: If you are sure the host has gone down, enter n; you are returned to DOS. If you think the host may be up, enter y and another connection request will be transmitted.

- **Error in initialization**

Explanation: The PC-Interface device driver was not successfully loaded.

User Response: Check that your PC-Interface working disk contains the BRIDGE.DRV and CONFIG.SYS files. If either file is missing, reinstall PC-Interface. Check that the CONFIG.SYS file properly specifies the location of BRIDGE.DRV. Retry the operation.

- **Error in network service.**

Explanation: A hardware or software error occurred in the LAN or RS-232 connection to the host.

User Response: Confirm with your network administrator that network services are available at this time. If so, check all hardware and wiring connections.

- **File Not Found**

Explanation: The specified file is inaccessible for one of the following reasons:

- The user invoking it does not own the file.
- A directory on the path to the file is not executable.
- The file is locked (that is, it is already opened by an application that supports file sharing).

User Response: Check that you own the file. Check that the directory containing the file and all directories on the path to the file have execute permission. If the file is locked, you must wait until it is closed and the lock is removed.

- **"HOSTS" and "EXCELAN\tcp" must agree**

Explanation: You only get this message if you have an Excelan network interface board and the Excelan software installed in your personal computer. This message means that the environment variables HOSTS (set by PC-Interface) and EXCELAN (set by the Excelan software) in your AUTOEXEC.BAT file are pointing to different HOSTS files. (The Excelan software assumes that the HOSTS file is in a subdirectory of EXCELAN called TCP.)

User Response: Change the value of the HOSTS variable so that it matches the value of the EXCELAN variable.

- **Input and output files cannot be the same**

Explanation: DOS2UNIX or UNIX2DOS was invoked using the same name for the source file and the target file.

User Response: Issue the DOS2UNIX or UNIX2DOS command again using different names for the source file and target file.

- **Input file is zero length**

Explanation: You specified a zero-length file as input to UNIX2DOS or DOS2UNIX.

User Response: Make sure the specified input is correct.

- **invalid broadcast address** *address*

Explanation: You specified an invalid argument to the /B option for PCIINIT.

User Response: You must specify either a valid host name or a valid IP address to the /B option. If you specified a host name, be sure it is in your HOSTS file. Ask your system administrator if you need assistance with a host name or an IP address.

- **Invalid combination of options**

Explanation: The DOS2UNIX or UNIX2DOS options are combined in an illegal way. The /B, /U, and /L options are mutually exclusive.

User Response: Retry the operation with a legal combination of options.

- **Invalid Drive**

Explanation: The specified drive does not exist or is inaccessible, or a drive other than a UNIX drive was specified.

User Response: Check that you are specifying a UNIX drive when a virtual drive specification is required. Make sure that the drive is accessible.

- **Invalid format.**

Explanation: ON could not execute the command as typed.

User Response: Check your ON command for logical consistency, especially as regards DOS and UNIX pipes and redirection, and type a corrected command.

- **invalid localhost address** *address*

Explanation: In the HOSTS file, the internet address specified for the "localhost" host is invalid.

User Response: Ask your system administrator for the correct address for your personal computer.

- **Invalid option–***option.*

Explanation: You specified an invalid option to DOS2UNIX or UNIX2DOS.

User Response: Refer to Chapter 6, Command Summary, for a description of the valid options to DOS2UNIX and UNIX2DOS. Reenter the command with a valid option.

- **jobs: bad numeric argument (*nn*)**

Explanation: The job number *nn* you typed was not found in the job table.

User Response: Retype a corrected command.

- **jobs: job not found**

Explanation: You attempted to reattach to a detached task with an invalid job ID number.

User Response: Display the current job table with the JOBS command. Retype your reattachment request with the correct job ID number.

- **kill: Must specify job or process ID.**

Explanation: KILL was entered without designating the job you want to terminate.

User Response: Retype a corrected command.

- **'localhost' entry not found**

Explanation: There is no entry with the alias "localhost" in the HOSTS file.

User Response: You should have a line like the following in the file \ETC\HOSTS:

internet address *host name* `localhost`

If you don't have this line, or if the word "localhost" is not next to the entry for your personal computer, add it.

- **Login Aborted**

Explanation: This message appears when you interrupt the login process over an RS-232 connection path by pressing F10 or when you answer "no" to the "Login Incorrect. Try Again?" prompt. The message acknowledges that the process has been aborted, as requested by the user.

User Response: None.

- **Login Incorrect. Try Again? (y or n)**

Explanation: Your user name or password was entered incorrectly.

User Response: Type `y` or `n`. If you choose `y` to retry the log in operation, type your user name and password carefully. If your login is still unsuccessful, consult the system administrator.

If you type `n`, the message "Login Aborted" is displayed and you are returned to the DOS prompt.

- **no free virtual drives**

Explanation: You tried to log in to a host while all of your PC-Interface virtual drives were in use.

User Response: Log out of one of your sessions to free one of the virtual drives, and retry. You may wish to increase the maximum number of virtual drives, as described in Chapter 5.

- **No Hosts Available For Service**

Explanation: The personal computer's request for a host map received no response, and there are no serial ports on this personal computer.

User Response: This is a host system problem. Consult your system administrator.

- **Not connected to a host.**

Explanation: You did not initiate host file services before attempting to run ON.

User Response: Use the PC-Interface LOGIN command to initiate host file services.

- **PC-Interface Copy Protection Violation - SYSTEM DISABLED**

Explanation: Two personal computer users attempted to use copies of the same PC-Interface distribution disk to log into PC-Interface simultaneously. The first user to log in is unaffected. The second user attempting to log in receives this message.

User Response: The disabled personal computer must be powered off and powered on again to be usable. Do not attempt to log in to PC-Interface using a duplicate disk. Simultaneous PC-Interface sessions can be established only by using a unique working disk for each session. Contact your system administrator.

- **PC-Interface Logged Out**

Explanation: This message appears whenever the LOGOUT command is issued.

User Response: None.

- **PC-Interface not initialized**

Explanation: The LOGIN command was invoked before PC-Interface was initialized.

User Response: Initialize PC-Interface using the PCIINIT command. Then retry the LOGIN command.

- **PC-Interface session terminated. Login to retry.**

Explanation: You answered "no" to the "Error -- Host did not respond, retry?" message. When you choose not to retry, your session is terminated.

User Response: To establish a host file services session to that host, type `login` and select the desired host.

- **Parameter Error**

Explanation: An incorrect entry exists on the BRIDGE.DRV line in CONFIG.SYS.

User Response: Examine the BRIDGE.DRV line in your CONFIG.SYS file. Refer to "Changing the Maximum Number of Virtual Drives" and "Changing the Job Table Size" in Chapter 5 for the correct syntax.

- **Path Not Found**

Explanation: The path specifying the location of the file does not exist.

User Response: Verify the location of the file and enter the path in the correct form.

- **printer: could not set state**

Explanation: You tried to set a printer path without having logged in to PC-Interface.

User Response: Log in to PC-Interface and retry.

- **Selected Baud Rate Invalid -- Try Again? (y or n):**

Explanation: An invalid baud rate was specified for an RS-232 path connection.

User Response: Type `y` or `n`. If you type `y` to retry the operation, enter a valid baud rate chosen from the baud rates shown.

- **Selected Host Not Available. Try Another? (y or n):**

Explanation: The selected host cannot accommodate any additional users, or there was a communication failure between the personal computer and the host.

User Response: Type `y` or `n`. If you type `y` to retry the operation, you can select a different host to log in to. If you prefer, you can attempt to make the selection that caused the error message. If the problem persists, consult your system administrator.

- **Selected Host Not in Table -- Try again?**

Explanation: An invalid response, including a carriage return, was given to the host-selection menu for host file services.

User Response: Type `y` or `n`. If you type `y` to retry the operation, give a valid host identification chosen from the host menu.

- **The PC-Interface software is not initialized**

Explanation: An attempt was made to invoke the emulator before PC-Interface was initialized. PC-Interface must be initialized before invoking the emulator.

User Response: Initialize PC-Interface using the PCIINIT command.

- **There is not enough memory for a new session**

Explanation: The personal computer memory is inadequate.

User Response: Retry the operation. If the message persists, install additional memory in your personal computer.

- **There was a syntax error in the arguments to the driver**

Explanation: In your CONFIG.SYS file, you used incorrect syntax in specifying jumper settings for your network interface board.

User Response: Refer to "Configuring Your Driver" in Appendix A of this manual for information on jumper settings. If you need more information, also see the documentation for

your personal computer and your network interface board.

- **Unknown switch**

Explanation: An invalid PCIINIT option is specified.

User Response: Read the description of PCIINIT in this manual and retry the operation with valid options.

- **Unix exec failed.**

Explanation: ON was unable to initiate the requested UNIX command on the host for a reason other than inability to access the requested program files. This message is generated when the maximum-processes-per-user-limit is reached on the host, among other reasons.

User Response: Type `on - ps -ef` to determine if the maximum processes limit has been reached. If so, wait until some processes finish before initiating others. If not, consult your network administrator for other possible failures on the host.

- **Unknown uexec error** (*nn*)

Explanation: An error occurred when ON tried to initiate the requested UNIX command on the host, but ON was unable to determine the cause of the error. The internal error code is shown in *nn*.

User Response: Confirm with your network administrator that the host computer and network services are available at this time.

- **Warning: 8 bit character detected without -b option**

Explanation: You may have given the wrong file name as input to UNIX2DOS or DOS2UNIX.

User Response: If the file name you gave is correct, you failed to specify the /B option to the command. Retry the command, specifying the /B option.

- **Warning: cannot set machine name**

Explanation: The attempt to set the local machine name failed. You are probably using the wrong version of the PC-Interface bridge.

User Response: Install Release 3.0 of the PC-Interface bridge.

Using A Dial-Up Modem

This appendix provides instructions for establishing a PC-Interface connection to a UNIX host through a dial-up modem. You can use PC-Interface through a modem in the same way that you use a direct RS-232 connection.

The exact steps required for establishing communications through a modem depend on which modem you are using. However, in general, the steps are similar. Once PC-Interface initializes the communications port, you make a telephone connection to the UNIX host. If you are using an automatic-dialing modem, you instruct the modem to dial the number from the keyboard. If you are using a manual modem, you dial the number and transfer control of the line to the modem after receiving a data tone.

This appendix describes the procedure using the internal Hayes Smartmodem for illustration. You should refer to the product manual for your dial-up modem for more detailed information on its operation.

When you plan to use PC-Interface only through a dial-up modem, you should install for RS-232 only. Follow the instructions in Appendix A for RS-232-only installation.

Before beginning your PC-Interface session, be sure your modem is attached to, or installed correctly in, your personal computer and that it is connected to a standard telephone jack.

Establishing A Host File Services Session

The following procedure describes logging in to a host file services session:

1. Boot your personal computer to initialize PC-Interface.

2. Type:

   ```
   login
   ```

   Press Enter (↵). The PC-Interface logo screen appears.

3. Press Enter again. The PC-Interface host menu screen appears.

   Enter the port number that appears next to the serial port to which the modem is connected.

4. The following prompt replaces the host-selection prompt:

   ```
   Enter Baud Rate (300,1200,2400,4800,9600):
   ```

5. Choose the baud rate appropriate for your modem. The internal Hayes Smartmodem uses 1200 baud.

6. When the five-line login window appears, it remains blank. Establish the telephone connection to your UNIX host, using the appropriate method for your modem.

   For example, if you have the Hayes Smartmodem, type:

   AT DT *number*

   where *number* is the telephone number of your UNIX host's dial-in modem (without hyphens or other punctuation).

   You hear the number being dialed, the phone ringing, and the host answer through the speaker in your personal computer. When the modem detects the host data tone, the words CONNECTED are displayed on your screen.

7. Press Enter. The UNIX host identification and login banner should appear.

8. Continue with the RS-232 login sequence described in "Logging In Over An RS-2323 Connection Path" in Chapter 2.

Establishing A Terminal Emulation Session

When you log in to a terminal emulation session from a host file services session, you do not need to re-establish the modem connection. Simply choose the same serial connection path, and log in as described in Chapter 4.

The following procedure describes logging in to a terminal emulation session over a modem when you do not have an existing host file services session:

1. Boot your personal computer to initialize PC-Interface.

2. Type `em2`.

3. Choose a serial connection path.

4. On the Configure menu, choose the communication parameters and baud rate appropriate for your modem. The internal Hayes Smartmodem uses 1200 baud and the default values for the remaining parameters.

5. When the terminal emulation screen appears, it remains blank. Establish the telephone connection to your UNIX host.

   With the Hayes Smartmodem, type:

   `AT DT` *number*

where *number* is the telephone number of your UNIX host's dial-in modem (without hyphens or other punctuation).

You hear the number being dialed, the phone ringing, and the host answer through the speaker in your personal computer. When the modem detects the host data tone, the words CONNECTED are displayed on your screen.

6. Press Enter. The UNIX host identification and login banner should appear.

7. Log in to the UNIX host and continue with your terminal emulation session as described in Chapter 4.

Tips On Using DOS Applications

To most DOS applications, the UNIX system is transparent; that is, once you have logged in to PC-Interface host file services, you can use UNIX file and print services exactly the same way you use your personal computer's local fixed disk drive and printer. However, there are a few DOS applications and commands that operate differently in the PC-Interface environment then they do in a conventional DOS environment. This appendix may help you resolve some of the differences you encounter when using these DOS programs with PC-Interface.

In the discussion below, we assume that the situations occur when you are logged into PC-Interface host file services and attempting to run DOS applications on a virtual drive.

DOS Program Cannot Create Files

Issue: A DOS application or command fails because it cannot create files. The error message displayed depends on the application or command being run.

Resolution: Some DOS applications and commands need to create files in the root directory. When you are working on a local drive, the files are written in the DOS root, for example, C:\. However, when you work on a virtual drive, the root directory is the UNIX root, which is normally not writable. This causes some applications to fail. There are several easy ways to circumvent this.

1. Check whether the application has an option to specify where temporary files are written. If so, specify a writable directory.

2. Use the DOS SUBST command[1] to create a drive specifier that references the virtual drive but whose root is a writable directory. Then run the application on the SUBSTed drive.

   For example, suppose you are logged into PC-Interface on drive D: and your home directory is \USR\SAM. The root directory on drive D: is the UNIX root (/), and is not writable. You can SUBST another drive letter (using any drive letter within the range of the DOS LASTDRIVE parameter) for the path D:\USR\SAM. Let's say you enter:

   ```
   subst e: d:\usr\sam
   ```

   Now the root directory of E: is \USR\SAM, which is writable, so you can run the DOS application from the E: drive.

   PC-Interface treats the SUBSTed drive like any other logged-in drive. When you initiate another login process without specifying a drive, PC-Interface chooses the next available drive, leaving the SUBSTed drive as it was.

DOS Program Does Not Work With Virtual Drive

Issue: Some DOS programs expect to run only with local drives. When such programs are used with PC-Interface, they

---

[1] If you are using DOS 3.1 or 3.2, you must execute the following command before using SUBST with PC-Interface:

```
pciconf set nd local
```

This command makes a remote drive look to DOS programs as if it were a local drive. See the description of PCICONF in Chapter 6 for more information.

fail to recognize virtual drives. One such application is Ashton Tate's Framework III.

Resolution: You need to tell the PC-Interface bridge to treat network drives as if they were local drives. To do this, execute the PCICONF program as follows:

```
pciconf set nd local
```

Note that for applications that expect to be run on a remote drive, this setting causes problems. When using applications that expect to work on a remote drive, you need to make sure ND is set to REMOTE. You can make sure of this by executing:

```
pciconf set nd remote
```

Using DOS Applications With A Serial Connection

Issue: You are using Microsoft Windows or Framework over an RS-232 connection, and you are having trouble—either the PC hangs, or you experience other unexpected behavior.

Resolution: When you log into host file services over a serial port, after you get the UNIX prompt but before you press F9, execute the following command:

```
stty clocal
```

If you always use Windows and Framework over RS-232, you can put the **stty clocal** command in your UNIX **.login** or **.profile** file, so that it is run automatically when you log in.

However, note that when you want to run other applications, you first need to execute the command:

```
stty -clocal
```

to reverse the effect of having run **stty clocal**.

Index

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

R

S

T

U

V