PATHWORKS for DOS

digital

SETHOST Terminal Emulation Guide

PATHWORKS for DOS

SETHOST Terminal Emulation Guide

Order Number: AA–PAFCC–TK

August 1991

Revision/Update Information: This document supersedes *SETHOST Terminal Emulation Guide*, Version 4.0, order number AA–PAFCB–TK.

Software Version: PATHWORKS for DOS Version 4.1

Digital Equipment Corporation
Maynard, Massachusetts

First Published, 1989
Revised, January 1991, July 1991

The postpaid Reader's Comments form at the end of this document requests your critical evaluation to assist in preparing future documentation.

This document was prepared using VAX DOCUMENT, Version 1.2

HOW TO ORDER ADDITIONAL DOCUMENTATION
DIRECT MAIL ORDERS

USA*
Digital Equipment Corporation
P.O. Box CS2008
Nashua, New Hampshire 03061

CANADA
Digital Equipment
of Canada Ltd.
100 Herzberg Road
Kanata, Ontario K2K 2A6
Attn: Direct Order Desk

INTERNATIONAL
Digital Equipment Corporation
PSG Business Manager
c/o Digital's local subsidiary
or approved distributor

In Continental USA, Alaska, and Hawaii call 800–DIGITAL.

In Canada call 800-267-6215.

*Any order from Puerto Rico must be placed with the local Digital subsidiary (809-754-7575).

Internal orders should be placed through the Software Distribution Center (SDC), Digital Equipment Corporation, Westminster, Massachusetts 01473.

Contents

3 Customizing Your SETHOST Session

4 Performing Functions Using SETHOST Scripts

A The SETHOST Default Settings

B SETHOST Error Messages and Codes

C SETHOST Character Sets

Glossary

Index

Examples

Figures

Tables

Preface

Purpose

This manual explains how to use the SETHOST utility on your personal computer. SETHOST is a utility that enables PATHWORKS users to connect their personal computer to a host node over a network and use their personal computer as a terminal. This manual provides the following information:

- An overview of SETHOST
- Instructions on how to start up a SETHOST session
- Instructions on how to customize your SETHOST session
- Instructions on how to create your own keyboard map
- Instructions on how to use the script language facility for both SETHOST and VT320 emulation

Audience

Users who want to connect their personal computers to a host node and emulate a terminal can do so by using SETHOST. This manual assumes that users are familiar with their personal computers and the DOS operating system. Users should also be familiar with the terminal they are emulating as well as PATHWORKS software for DOS. To find out more about PATHWORKS for DOS, see the *User's Handbook*.

This manual is also for VT320 emulation users who want to write and use scripts.

Organization

The following table can help you find information in this manual.

Chapter 1	Provides an overview of the SETHOST utility and instructions on what to do before you start your SETHOST session. This chapter also describes the SETHOST function keys.
Chapter 2	Provides instructions on how to start a SETHOST session for the first time.
Chapter 3	Describes each Set-Up screen and its options as well as the default parameters for each Set-Up screen. This chapter also tells you how to create new keyboard maps and how to use environment variables.
Chapter 4	Describes the script language facility and contains descriptions of the script commands and sample script programs. It applies to both SETHOST and VT320 emulation.
Appendix A	Lists the default Set-Up Settings.
Appendix B	Provides a list of CTERM, LAT, and TELNET error messages and recovery procedures.
Appendix C	Describes how to use international character sets and code pages with SETHOST.

Related Documents

The following is a list of PATHWORKS for DOS documents that provide additional information for users:

- *User's Handbook*
- *Client Commands Reference*
- *DECnet Network Management Guide*
- *DECnet User's Guide*
- *Client Installation and Configuration Guide with Diskettes*
- *Microsoft Windows Support Guide*
- *Software Product Description*
- *TCP/IP User's Reference*

Conventions

This manual uses the following conventions:

Convention	Meaning
Ctrl/*x*	While you hold down the Ctrl key, press another key or a pointing device button.
Ctrl/Alt/Del	While you hold down the Ctrl and Alt keys, press the Del key.
Esc *x*	Press the Esc key, release it, and then press another key or a pointing device button.
Return	Press the key that executes commands or terminates a sequence. This key is labeled Return, Enter, or ↵, depending on your keyboard.
"enter"	Type all required text, spaces, and punctuation marks; then press Return, Enter, or ↵, depending on your keyboard.
UPPERCASE	In VMS, DOS, and OS/2 syntax, uppercase letters indicate commands and qualifiers. You can enter commands and qualifiers in any combination of uppercase or lowercase, unless otherwise noted. ULTRIX commands are case-sensitive. You must enter commands in the correct case, as printed in the text.
lowercase	Lowercase letters in VMS, DOS, and OS/2 syntax indicate parameters. You must substitute a word or value, unless the parameter is optional.
teal blue type	In examples of dialog between you and the system, teal blue type indicates information that you enter. In online (Bookreader) files, this information appears in boldface.
boldface	Boldface type indicates a new term that appears in the glossary. In online (Bookreader) files, boldface indicates information you enter.
kp*n*	Press the specified key on the numeric keypad of your keyboard.
two-line commands	In VMS commands, a hyphen (-) at the end of a command line indicates that the command continues to the next line. If you type the hyphen and press Return, the system displays the _$ prompt at the beginning of the next line. Continue entering the command. If you do not type the hyphen, VMS automatically wraps text to the next line. In ULTRIX commands, a backslash (\ ) performs the same function.

Convention	Meaning
	In DOS and OS/2 commands, no character is used at the end of the first line; DOS automatically wraps text. Enter the complete command, then press Return at the end of the command.
[ ]	Square brackets in command descriptions enclose the optional command qualifiers. Do not type the brackets when entering information enclosed in the brackets.
/	A forward slash in command descriptions indicates that a command qualifier follows.
|	A vertical bar in command descriptions indicates that you have a choice between two or more entries. Select one entry unless the entries are optional.
. . .	A horizontal ellipsis following an entry in a command line indicates that the entry or a similar entry can be repeated any number of times. An ellipsis following a file name indicates that additional parameters, values, or information can be entered.
. . .	A vertical ellipsis in an example indicates that not all the data is shown.
NOTE	Notes provide information of special importance.
CAUTION	Cautions provide information to prevent damage to equipment or software.
WARNING	Warnings provide information to prevent personal injury.

Terminology

The terms "personal computer" (PC) and "PC workstation" refer to standalone systems. The term "client" refers to a PC, connected to the network by PATHWORKS software, that can access resources on a server. A server is a system that offers services to clients.

The term "PATHWORKS" refers to PATHWORKS software. PATHWORKS is a trademark of Digital Equipment Corporation.

1

Before You Begin

PATHWORKS software for DOS includes the SETHOST utility, a network utility that you can use to connect your personal computer to a host node. After your personal computer is connected with SETHOST, it can emulate a terminal connected to a host node or service. This emulation is called a virtual terminal. SETHOST also provides a script language facility for automating host commands and provides the ability to custom tailor your environment.

SETHOST supports the DECnet transport, the Transmissions Control Protocol/Internet Protocol (TCP/IP) transport, Token Ring, and the Local Area Transport (LAT). Over these transports, SETHOST provides VT320, VT220, VT100, and VT52 terminal emulation.

You can make asynchronous connections through your personal computer's COMM port or multiple local or wide area connections over a network connection. Using SETHOST, you can use one of the following service types to connect your personal computer to host computer systems:

- Through LAT over Ethernet to any local service
- Through DECnet by using CTERM to any host in the DECnet network
- Through the asynchronous communications port (COM1 or COM2)
- Through TCP/IP by using TELNET to any host in the TCP/IP network
- Through Token Ring by using TELNET, CTERM, or NET

Each time you connect to a host or service, the connection you make is called a **session**. Using SETHOST, you can create and maintain multiple sessions. You can have a maximum of four sessions active at once. The following are the different types and numbers of sessions you can choose from:

- COM1—one session only.
- COM2—one session only.
- LAT and CTERM—a combination of four sessions.
- TELNET—four sessions.

Getting Started

The remainder of this chapter describes the tasks that you should complete before starting your SETHOST session:

- Select a network service
- Check that transport drivers are installed
  - If you use TCP/IP, check the order of the terminal drivers installation
  - If you use Microsoft Windows, check that the transport drivers are installed correctly
- Choose a keyboard map
- Determine a character set
- Set up your printer for printing from within SETHOST
- Set up your monitor to display 132 columns
- Get help on the tasks if necessary

Selecting a Network Service

The SETHOST utility defaults to an asynchronous connection to COM1 at 9600 baud.

The SETHOST utility offers four types of network services to connect your personal computer to a remote host. Each service type has its own characteristics. Selecting a Network Service Type explains how to select and change your network service type from the Set-Up screens.

- Local Area Transport (LAT)

  The Local Area Transport (LAT) protocol is layered on DLL and uses Ethernet to provide access to other LAT services. With LAT, you can connect your personal computer to a host computer or a terminal server that supports printers or modems.

  Connections made using the LAT protocol are usually the fastest way to communicate with a host in a local area network (LAN). The host to which you connect echoes the characters that you type. Use the LAT service type to connect to a host on a local area network.

  When you use LAT to send or receive files, data can be lost because LAT does not use a protocol to ensure that the file arrives intact.

- Command Terminal (CTERM)

  The CTERM protocol is layered on DECnet. CTERM provides wide area network (WAN) capabilities allowing direct access from your personal computer to the DECnet network, including hosts that reside outside your local area network. Because CTERM is designed for use in a WAN, it echoes the characters you type locally whenever possible.

  To create SETHOST connections by using CTERM, the host node must support terminal emulation from remote nodes and run Phase IV or Phase IV+ of the DECnet software or Token Ring on a supported operating system. The PATHWORKS for DOS, *Software Product Description* lists the supported operating systems.

  When you use a CTERM connection to send or receive files, DECnet ensures that no data is lost.

  The CTERM protocol is used over an asynchronous DECnet connection or Token Ring connections.

- Asynchronous Communications

  The asynchronous communications port (COM1 or COM2) provides an asynchronous connection. Use either port to connect your personal computer to a terminal server or a modem.

  When you use a COMM port connection to send or receive files, data can be lost because no protocol is used to ensure that the file arrives intact.

  If you are running asynchronous DECnet that uses a COMM port, you can make this port available to asynchronous communications by issuing the Network Control Program (NCP) command SET LINE STATE OFF. The command SET LINE STATE ON returns the COM port to DECnet use. (For more information about this command, refer to the PATHWORKS for DOS *DECnet Network Management Guide*.)

- TELNET

  The TELNET protocol lets users establish remote TCP/IP connections to other systems running the TCP/IP protocol. Because TELNET uses the TCP/IP stack, it is the only way to connect to other systems that are running the TCP/IP protocol. You can also use TELNET with Token Ring connections.

  To transfer files using TELNET, use the FTP utility. Refer to the PATHWORKS for DOS, *TCP/IP User's Guide* for information on using the FTP utility.

Loading the Network Terminal Drivers

When SETHOST connects your system to another node or service, it checks to see if you have the appropriate files installed. If you do not, SETHOST loads the network terminal drivers for the type of network service that you request, then unloads them when you finish.

The network terminal drivers are loaded from the server from whichever drive the system service files are on. The drivers are loaded into either conventional memory or Expanded Memory (EMS). The files are loaded automatically except when SETHOST is used with DOS 5.0, Microsoft Windows, or other task-switching environments. See Running SETHOST with TCP/IP and Running SETHOST Under Microsoft Windows for more information.

Table 1–1 lists the files necessary to run each network service.

Table 1–1 Network Terminal Driver Files

To Use This Network Service...	Install These Files...
LAT	LAT.EXE (LAT driver)
CTERM	CTERM.EXE (CTERM driver)
TELNET	TN.EXE (TELNET driver) BAPI.EXE (Bridge Architecture Protocol Interface (BAPI) driver) DNRTSR.EXE (Domain Name Resolver driver)

Running SETHOST with TCP/IP

Be aware of the following guidelines when using the SETHOST utility with TCP/IP and TELNET:

- To use SETHOST on TCP/IP with task-switching environments such as Microsoft Windows 3.0, DOS 5.0, DESQview, or Software Carousel:

  1. Set the DOS environment variable NVTWIN to 1 (SET NVTWIN=1) to prevent SETHOST from trying to load or unload the network terminal drivers. Refer to Tailoring Your SETHOST Environment in Chapter 3 for information on how to set the SET NVTWIN environment variable.
  2. Load the network terminal drivers *before* running a task switcher. You must load DNRTSR first, TN second, and BAPI last.
  3. After exiting the task-switching environment, you can unload the network terminal drivers. The drivers unload in the reverse order from which they were installed. To unload the drivers, enter:
     - TCPUNLD/TOPONLY (unloads BAPI)
     - TCPUNLD/TOPONLY (unloads TN)
     - TCPUNLD/TOPONLY (unloads DNRTSR)

- TELNET connections are case sensitive for the host names, so SETHOST passes to the BAPI interface the exact case of the host name you supply. If TELNET does not know the host name, SETHOST reverses the case of the name entered and tries again.

- The SETHOST setup screens may appear corrupted if you select Network Communications Port when using a TCP/IP configuration. The poor appearance occurs because SETHOST loads the TCP/IP drivers DNRTSR.EXE, TN.EXE, and BAPI.EXE.

  As each of these drivers are installed, text displays on the screen. The text can cause the screen to appear corrupted. If the screen appears corrupted, press the F3 key twice to exit and then reenter setup to clear the problem.

Running SETHOST Under Microsoft Windows

If you want to run SETHOST in a Microsoft Windows environment, make sure that you load your network terminal drivers *before* you invoke Microsoft Windows. For example, if you plan on using CTERM as your network terminal driver, load CTERM.EXE before running Microsoft Windows. Loading the Network Terminal Drivers lists the available network terminal drivers.

Note

Whenever you use SETHOST in task-switching environments (such as DOS 5.0, DESQview, Software Carousel, or Microsoft Windows 3.0), load the network terminal driver *before* running the task-switching software.

To ensure that SETHOST runs correctly in a Microsoft Windows environment, you also need to set an environment variable. Setting the SET NVTWIN environment variable equal to 1 (SET NVTWIN=1) prevents SETHOST from trying to dynamically load and unload the network terminal drivers (LAT.EXE, CTERM.EXE, DNRTSR.EXE, TN.EXE, and BAPI.EXE). Microsoft Windows does not allow the dynamic loading and unloading of these drivers and can cause your system function incorrectly if you do not load the drivers first. Refer to Tailoring Your SETHOST Environment for information on how to set the SET NVTWIN environment variable.

After exiting the task-switching environment, you can unload the network terminal drivers by entering the appropriate driver command at the DOS prompt:

To unload LAT, enter:

```
LAT /U Return
```

To unload CTERM, enter:

```
CTERM /U Return
```

To unload TELNET, enter:

```
TCPUNLD/TOPONLY
TCPUNLD/TOPONLY
TCPUNLD/TOPONLY
```

The commands unload TELNET drivers in the reverse order from which they were loaded. The commands unload BAPI first, then TN, then DNRTSR.

If you are using LAT, you must define a Session Control Block (SCB) for each LAT session you plan to start *before* loading the LAT drivers and running Microsoft Windows. An SCB is a data structure used by the system that contains all the interrupt and exceptions for the session. An SCB enables the system to keep track of the LAT session. You define SCBs for LAT connections using the LATCP DEFINE SCB command. For example, to define 4 SCBs, enter the following command after you invoke the LATCP utility:

```
LATCP>DEFINE SCB 4 Return
```

For more information about the LATCP utility and the DEFINE SCB command, refer to the *Client Commands Reference*.

Choosing the Appropriate Keymap

If you have a keyboard other than an IBM XT style keyboard, select the keymap that most resembles yours. If SETHOST cannot match your keyboard to a supported one, it uses the IBM XT style keyboard as the default setting.

Help (.HLP) files in the HELP directory illustrate the keyboard mappings included with SETHOST. Each file begins with the letters KBD and has the file type .HLP.

The files beginning with KBD and having a file type of .KBD are the keyboard-mapping files that control the way the personal computer keys map to a particular terminal emulator function.

If your keyboard does not appear in the following list, compare the .HLP files to your keyboard to find the keyboard that most resembles yours. If your keyboard is not an exact match, use the information in Chapter 3, Customizing Your Keyboard Mapping Using Keyboard-Mapping Files, to customize it. You may also use the generic keymap. The generic mapping prevents SETHOST from replacing the keyboard interrupt 9, and relies only on the keys that the personal computer normally sends.

Table 1–2 lists the keyboards that SETHOST provides .HLP and .KBD files for.

Table 1–2 SETHOST Keymap Help and Keymap Files

To Use This Keyboard...	Refer to These Files...
IBM PC XT (default)	KBDXT.HLP and KBDXT.KBD
IBM PC AT	KBDAT.HLP and KBDAT.KBD
IBM Enhanced PC	KBDEPC.HLP and KBDEPC.KBD
LK250	KBDLK.HLP and KBDLK.KBD
Laptop (generic)	KBDLAP.HLP and KBDLAP.KBD
Generic	KBDGEN.HLP and KBDGEN.KBD
Olivetti M28	KBDM28.HLP and KBDM28.KBD
Olivetti M24	KBDM24.HLP and KBDM24.KBD
Other Use the Other keymap as a template for creating your own custom keyboard map.	KBDOTHER.HLP and KBDOTHER.KBD
COMPAQ SLT/286 Laptop	KBDSLT.HLP and KBDSLT.KBD
COMPAQ LTE/286 Laptop	KBDLTE.HLP and KBDLTE.KBD
COMPAQ LTE/286 Laptop with Keypad mapping	KBDLTEKP.HLP and KBDLTEKP.KBD
Zenith AT	KBDZENAT.HLP
Zenith XT	KBDZENXT.HLP

For example, to view the .HLP file for the IBM XT keyboard, use the following command at the DOS prompt:

```
C:\>TYPE KBDXT.HLP Return
```

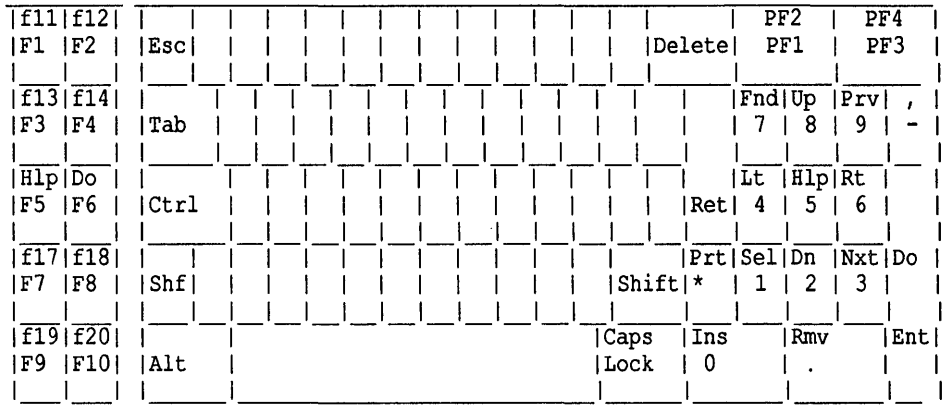
If you are using a PATHWORKS server, you can also use:

```
C:\>HELP KBDXT Return
```

Figure 1–1 displays sections of the KBDXT.HLP file that scrolls on your screen.

Figure 1–1 Sample Keyboard Help File

XT keyboard mapping for SETHOST.
Use this keyboard map if your keyboard is similar to the IBM XT keyboard.



Alt/NumLock can be used to toggle the Right Hand Side Keypad between
DEC Numeric/Application Keypad and a DEC Editing Keypad.

Upper Keypad

KEY	NORMAL	SHIFT/KEY	CTRL/KEY
F1	Hold Screen	Autotyping	Start Session Logging
F2	Print Screen	Not Mapped	Auto Print Mode SET/RESET
F3	Set-Up	Not Mapped	Stop Session Logging

.
.
.

KEY	ALT/KEY	SHIFT-ALT/KEY
F1	F11	Shifted F11 (UDK)
F2	F12	Shifted F12 (UDK)
F3	F13	Shifted F13 (UDK)

.
.
.

Right Hand Side (Numeric) Keypad

KEY	NORMAL	SHIFT/KEY	CTRL/KEY

(continued on next page)

Figure 1–1 (Cont.) Sample Keyboard Help File

```
Num Lock        PF1             PF2             Not Mapped
Scroll Lock     PF3             PF4             Not Mapped
7               7               Find            Not Mapped
.
.
.
                        Main Keyboard

KEY             NORMAL          SHIFT/KEY       CTRL/KEY        ALT/KEY
1               Not Mapped      Not Mapped      Not Mapped      Not Mapped
2               Not Mapped      Not Mapped      Null            Not Mapped
3               Not Mapped      Not Mapped      Esc             Not Mapped
.
.
.
```

Note

To conserve disk space, store only the keyboard map and help files you want to use.

Using SETHOST Function Keys

Function keys have fixed meanings for SETHOST. For example, SETHOST uses F1 to freeze the currently displayed screen. Table 1–3 lists these function keys and their descriptions.

On LK250 keyboards, the 20 function keys are numbered F1 through F20.

On keyboards that have only ten function keys, SETHOST uses these function keys for F1 through F10. You can use the second set of SETHOST function keys, by using the Alt key in combination with the ten function keys. Thus, Alt/F1 through Alt/F10 correspond to F11 through F20.

Note

The PF1 key (or the GOLD key) is usually the upper left most key on the numeric keypad. Refer to the appropriate keyboard help file to find the location of the PF keys on your keyboard.

Table 1–3 SETHOST Function Key Definitions

To Perform This Function...	Press This Function Key...
Freeze (and unfreeze) the screen.	F1 (Hold Screen)
Send the screen text to the printer.	F2 (Print Screen)
Enter (and exit) Set-Up.	F3 (Set-Up)
Toggle between multiple SETHOST sessions.	F4 (Next Session)
Transmit a break (if you enabled break in Set-Up).	F5 (Break)
Send a file.	Shift/F1
Dial a telephone number stored in the Telephone Set-Up menu.	Shift/F4
Initiate a communications line disconnect.	Shift/F5
Receive a file. (Start or stop logging the SETHOST session to a disk file.)	Ctrl/F1
Toggle Auto Print mode.	Ctrl/F2
Close session logging.	Ctrl/F3
Toggle between data and talk mode.	Ctrl/F4
Send an answerback message as defined in Set-Up.	Ctrl/F5
Initiate a COMPOSE sequence.	Ctrl/F6 or Compose on the LK250 keyboard
Toggle the screen left and right in 132-column scroll mode.	Ctrl/F8
Suspend the session and go to the DOS operating environment.	Ctrl/F9
Exit from SETHOST.	Ctrl/F10

Determining Which Character Set to Use

The SETHOST utility uses its own character sets to ensure that the characters generated by host system application programs are accurately represented on your personal computer. A **character set** is a group of graphic characters and control characters stored as a unit. The graphic characters are characters that you can display on your screen and control characters perform special functions. SETHOST uses the DEC Multinational Character Set (MCS) as its default

character set. (See Appendix C, Tables C–3 through C–5.) The MCS character set is an 8-bit character.

If you are using SETHOST to run a host system application in American English, the default MCS is all you need. However, if you are using SETHOST to run a host system application in a European language, SETHOST provides the National Replacement Character sets (NRC) (see Table C–6).

If the application you are running was written to support VT300- and VT200-type terminals, the MCS (or ISO Latin) character set should work properly. If the application you are running was written to support VT52 or VT100 type terminals, use the NRC for your particular country. SETHOST supplies NRCs for the following countries:

- Britain
- Canada (French)
- Finland
- France
- Germany
- Holland
- Italy
- Norway or Denmark
- Portugal
- Spain
- Sweden
- Switzerland

The PATHWORKS for DOS installation procedure installs the SETHOST code page files that match the code page you are running. If you change character sets, check to make sure you have the correct code page installed.

It is important that the appropriate code page is installed. Refer to the PATHWORKS for DOS, *Client Installation and Configuration Guide* for information about installing code pages.

The SETHOST utility supports the following character set code pages:

- Code page 437 - US English (same as STD ROMS)
- Code page 850 - Multilingual
- Code page 865 - Nordic (same as ST2 ROMS)
- Code page 860 - Portuguese
- Code page 863 - Canadian-French

Refer to Appendix C for more information about code pages.

Setting Up Your Printer for Printing from Within SETHOST

The PATHWORKS for DOS software allows you to set up printers accessible through your personal computer from within SETHOST.

To use the print options from within SETHOST, you must configure your printer to print from the DOS operating system before you invoke SETHOST. PATHWORKS for DOS uses the XON/XOFF protocol to support serial printers. Before invoking SETHOST, you must execute the XONXOFF and DECMODE commands. Refer to the PATHWORKS for DOS, *Client Commands Reference* (Chapter 4), for information on how to execute these commands and for a list of supported printers.

After you configure your printer, follow these steps to complete the printer setup:

1. Invoke SETHOST.
2. Tell SETHOST that the printer is connected by selecting the Printer Connected option from the Printer Set-Up screen.
3. Tell SETHOST which device name to print to by highlighting the Print File Name option from the Printer Set-Up screen. Use ⇨ to move the cursor to the SELECTIONS column and type in a device name. For example, you can type LPT1.
4. Highlight the type of printer that you have (DEC Printer or Other Printer) from the Printer Set-Up screen.
5. Save these settings in the Actions Set-Up screen.

For more information on using the Printer Set-Up screen, refer to Chapter 3, Defining Printer Operations.

Setting Up Your Monitor to Display 132 Columns

If you have a video adapter that supports 132-column mode, you can set up your monitor to display 132 columns. The SETHOST utility uses 80 columns with screen shifting as the default.

To use 132-column mode, you must set the NVT80 and NVT132 environment variables before invoking SETHOST. Refer to Tailoring Your SETHOST Environment for information on how to set environment variables.

Getting Help for SETHOST

The SETHOST utility provides help text within its own utility. To view the SETHOST help text, enter the following command at the DOS prompt:

```
C:\>SETHOST HELP Return
```

Several screens of help information are available. SETHOST prompts you for more at the end of every screen. To get the next screen of help text, type Y (or Return). To get back to the DOS prompt, type N.

This help file is called SETHOST.HLP. It is an ASCII text file that you can modify if desired.

The PATHWORKS for DOS product also provides a Help utility outside of SETHOST. If you are using a PATHWORKS server, you can get help on SETHOST from outside of the SETHOST utility. Type the following command at the DOS prompt:

```
C:\>HELP SETHOST Return
```

2

Using SETHOST for the First Time

When you first run SETHOST, it uses the default Set-up settings to define terminal characteristics, such as the type of connection you want or the keyboard map to use. For example, the default connection method for SETHOST is an asynchronous connection over the COM1 port at 9600 baud.

To make a network connection (LAT, CTERM, or TELNET) to a specific node or service, use one of the following two methods:

- The command line method

  Specify the service type and node or service name as part of the SETHOST command line. If you use the command line method, you specify that information every time you make a connection.

- The Set-Up screen method

  Specify settings for your session and save them for use with future sessions using the Set-Up screens. The SETHOST utility stores this information and any other Set-Up parameters you choose in a Set-Up file. The default Set-Up file is named SETHOST.DAT.

  The Set-Up file contains information about the system to connect to (called a preferred service), the connection method to use, the keyboard map to load, the character set to use, and other parameters. Once you save this information, you can make successful network connections using the information stored in the Set-Up file.

Check the following settings before starting your SETHOST session:

- Keyboard Map
- Character Set
- Service Type
- Service Name

If you change any of these settings, consider saving the changes. Refer to Saving Your Set-Up File for more information on saving your settings.

Limitations in Service Types

The following limitations exist when running SETHOST with a CTERM connection:

- When you use the VMS command recall feature immediately after switching from one CTERM session to another, the first line that is recalled may be from the first CTERM session.
- When you type command lines that are longer than one line, and your terminal is set to wrap, the backspace/delete key works only on the last line of the command.

Starting SETHOST Sessions from the Command Line

If you specify SETHOST by itself, it searches SETHOST.DAT for a service type and host node or service name. If you have not specified a service type or connection method in SETHOST.DAT and you are running the DECnet transport, SETHOST attempts a LAT connection first then tries a CTERM connection if LAT fails. If you are running the TCP/IP transport, SETHOST tries only a TELNET connection.

If you specify SETHOST by itself and you do not have a SETHOST.DAT file, SETHOST defaults to COM1 at 9600 baud.

To start a SETHOST session using the default settings (or settings saved in the default Set-Up file), enter:

```
c:\> SETHOST Return
```

You can also start a SETHOST session by specifying first the connection method and any of a number of options. To start SETHOST and specify options, use the following syntax:

Format

SETHOST [*service* | LAT: *service* | CTERM: *service* | NET: *service* | TELNET: *service* | COM*n*: | MCOM*n*: | MODEM:] [/SCRIPT=*name*] [/COLOR | /MONO] [/KEYBOARD=*style*] [/R]

LAT:	Instructs SETHOST to attempt only a LAT connection over the Ethernet to any local service. You may also specify up to four service names to connect to using LAT.
CTERM:	Instructs SETHOST to attempt only a CTERM connection over the DECnet protocol to any node in the DECnet network. You may also specify up to four node names to connect to using CTERM.
NET:	Instructs SETHOST to use the method that connects to the network fastest. For DECnet connections, SETHOST tries LAT first, then CTERM if LAT fails. For TCP/IP connections, SETHOST tries only TELNET.
TELNET:	Instructs SETHOST to attempt only a TELNET connection over the TCP/IP protocol. You may also specify up to four service names to connect to using TELNET.
COM	Specifies a connection over a COM port using data leads only. If you specify a connection method, it must be the first option specified.
MCOM	Specifies a connection over a COM port using full modem control. If you specify a connection method, it must be the first option specified.
MODEM	Specifies a connection over COM2 using a Hayes™-compatible dialing string. This option automatically begins the dialing string with "ATDT." If you specify a connection method, it must be the first option specified.
/SCRIPT=	Specifies that a certain script is to be used.
/COLOR	Sets the display to color.
/MONO	Sets the display to monochrome. Use if you have a color display and cannot see the screen highlights in the set-up screens. Usually, this applies to monitors that do not support 16 colors in text mode.
/KEYBOARD=	Specifies the keyboard and uses the appropriate keymap for it.

/R — If you specified a COM port, reorders the interrupt priorities preventing data overruns and lost data. Use this switch only if you are having a problem with data overrun or lost data.

Variables:

service — Specifies the name of the host node or service to connect to. You can specify up to four node or service names.

n — Specifies the COM port to use (1 or 2).

name — Specifies the name of the script that you want to use for this session.

style — Specifies the keyboard style. Available keyboard settings are:

XT	XT style keyboard
AT	AT style keyboard
EPC	IBM XT/AT (Enhanced) PC keyboard
LK250	DEC LK250 style keyboard
LAPTOP	Laptop style keyboard
Generic	Generic keyboard
M28	Olivetti M28 86-key standard
M24	Olivetti M24 extended keyboard
Other	User-customizable entry
SLT	COMPAQ® SLT/286 laptop
LTE	COMPAQ LTE/286 laptop
LTEKP	COMPAQ LTE/286 laptop with external keypad

Examples

1. To connect to node HOST1 using a script called SCR1.SCR, enter the following command:

   ```
   C:\>SETHOST HOST1 /SCRIPT=SCR1.SCR Return
   ```

2. To connect to node HOST3 using the SETHOST.DAT settings, use the following command:

   ```
   C:\>SETHOST HOST3 Return
   ```

3. To establish an asynchronous connection through COM1 at 9600 baud, use the following command line:

   ```
   C:\>SETHOST COM1: Return
   ```

4. To establish a connection using a laptop with a monochrome screen (which is linked to the network through a communications port), use the following command line:

   ```
   C:\>SETHOST COM2: /KEYBOARD=LAPTOP /MONO[Return]
   ```

5. To establish a LAT connection to node HOST6 using a script called LOGIN.SCR, use the following command at the DOS prompt:

   ```
   C:\>SETHOST LAT: HOST6 /SCRIPT=LOGIN.SCR [Return]
   ```

6. To establish CTERM connections to nodes HOST5 and HOST7, use the following command at the DOS prompt:

   ```
   C:\>SETHOST CTERM: HOST5  HOST7 [Return]
   ```

7. To connect to node HOST8 using the fastest connection method, use the following command:

   ```
   C:\>SETHOST NET: HOST8 [Return]
   ```

8. To establish a TCP/IP connection with a host called HOST9, use the following command:

   ```
   C:\>SETHOST TELNET: HOST9 [Return]
   ```

   If the host is not on the network, TELNET fails immediately. If the host is on the network, but is not reachable, you must wait for the connection to time out before you can exit SETHOST.

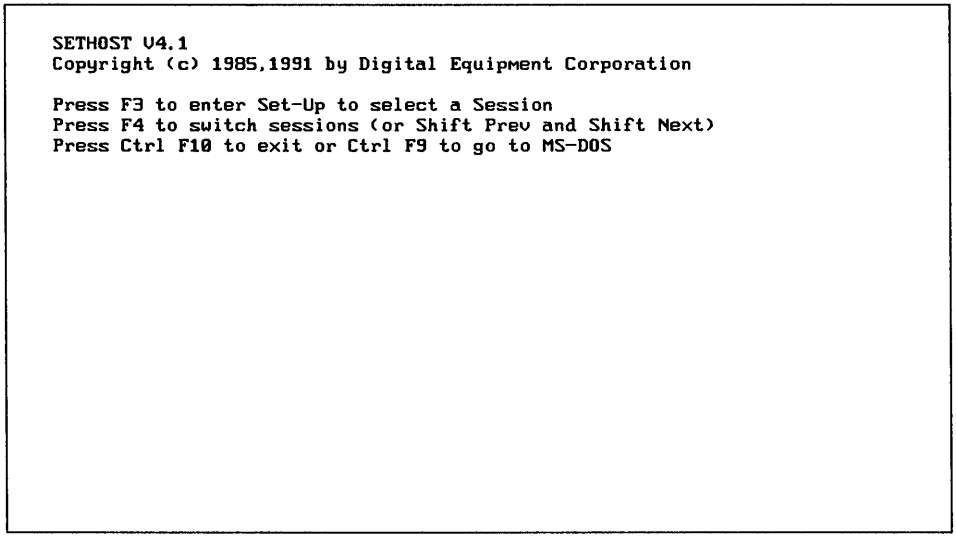
Starting SETHOST Using the Set-Up Screens

Use the Set-Up screens to supply parameters to your Set-Up file. To access the screens, type SETHOST at the system prompt. For example:

```
C:>SETHOST Return
```

Figure 2–1 shows a sample log-on screen.

Figure 2–1 Sample Log-on Screen

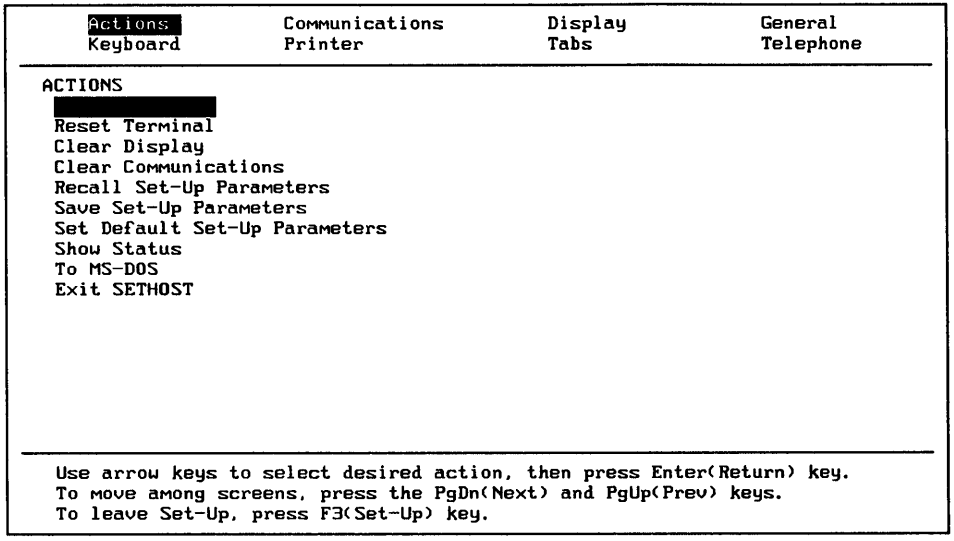


Now, you have a number of choices:

- Press F3 to enter the Set-Up screens.
- Press F4 to switch between multiple sessions.
- Press Ctrl/F9 to access the DOS environment.
- Press Ctrl/F10 to end the SETHOST session.

If you enter the Set-Up screen, a screen similar to Figure 2–2 appears:

Figure 2–2 Sample Actions Set-Up Screen



Each Set-Up screen is divided into three sections:

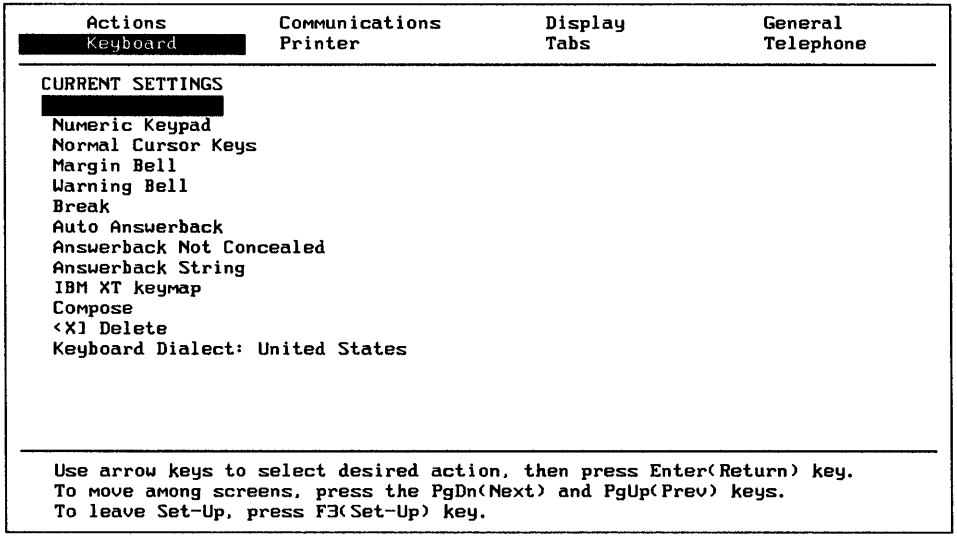
- A top section indicating the available Set-Up screens. The selected screen is highlighted by a reverse video bar.
- A middle section reflecting information about the selected Set-Up screen.
- A bottom section describing the keys you can use to change or select information.

Selecting a Keymap

If you are not using an IBM XT style keyboard, check your keyboard setting. To access the keyboard mapping files, follow these steps:

1. Use [Pg Up] and [Pg Dn] to move the highlighted video bar to the Keyboard selection on the top of the Communications Set-Up screen. The Keyboard Set-Up screen appears, as shown in Figure 2–3.

Figure 2–3 Sample Keyboard Set-Up Screen



2. Use ⇩ to move the video bar to the keymap choice in the CURRENT SETTINGS column.

   For example, use ⇩ to move the video bar to the IBM XT selection. A list of keymaps appears in the SELECTIONS column, as shown in Figure 2–4.

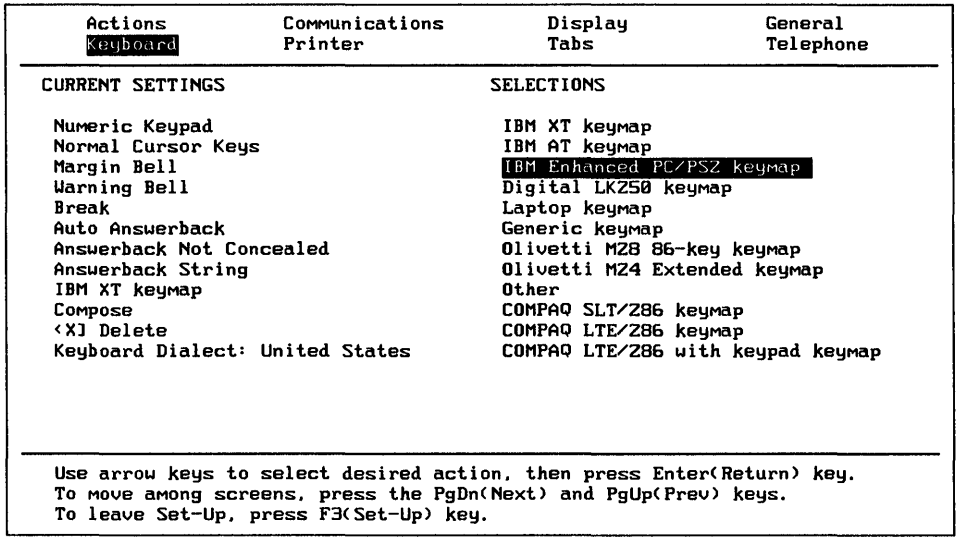
Figure 2–4 Sample Keymap Selections

```
   Actions          Communications      Display          General
   Keyboard         Printer             Tabs             Telephone

CURRENT SETTINGS                    SELECTIONS

 Numeric Keypad                      IBM XT keymap
 Normal Cursor Keys                  IBM AT keymap
 Margin Bell                         IBM Enhanced PC/PS2 keymap
 Warning Bell                        Digital LK250 keymap
 Break                               Laptop keymap
 Auto Answerback                     Generic keymap
 Answerback Not Concealed            Olivetti M28 86-key keymap
 Answerback String                   Olivetti M24 Extended keymap
 IBM XT keymap                       Other
 Compose                             COMPAQ SLT/286 keymap
 <X] Delete                          COMPAQ LTE/286 keymap
 Keyboard Dialect: United States     COMPAQ LTE/286 with keypad keymap

  Use arrow keys to select desired action, then press Enter(Return) key.
  To move among screens, press the PgDn(Next) and PgUp(Prev) keys.
  To leave Set-Up, press F3(Set-Up) key.
```

3. If the default keymap entry is the one you want, do not make any changes. To choose a different keymap, press ⇨ to move the video bar to the SELECTIONS column, then press ⇧ and ⇩ to highlight your selection.

   For example, to select the IBM Enhanced PC/PS2 keymap, move the video bar to that choice, as shown in Figure 2–5.

Figure 2–5 Selecting a Keymap



4. To finalize your selection, press Return. The name of the selected keymap appears in the CURRENT SETTINGS column.

   To move back to the CURRENT SETTINGS column without making a selection, press ←.

Selecting a Keyboard Dialect

Select the keyboard dialect that most resembles the country keyboard you are using. SETHOST uses the United States keyboard dialect as the default. The following keyboard dialects are available:

- United States
- Britain
- France
- Germany
- Italy
- Spain
- Sweden
- Finland

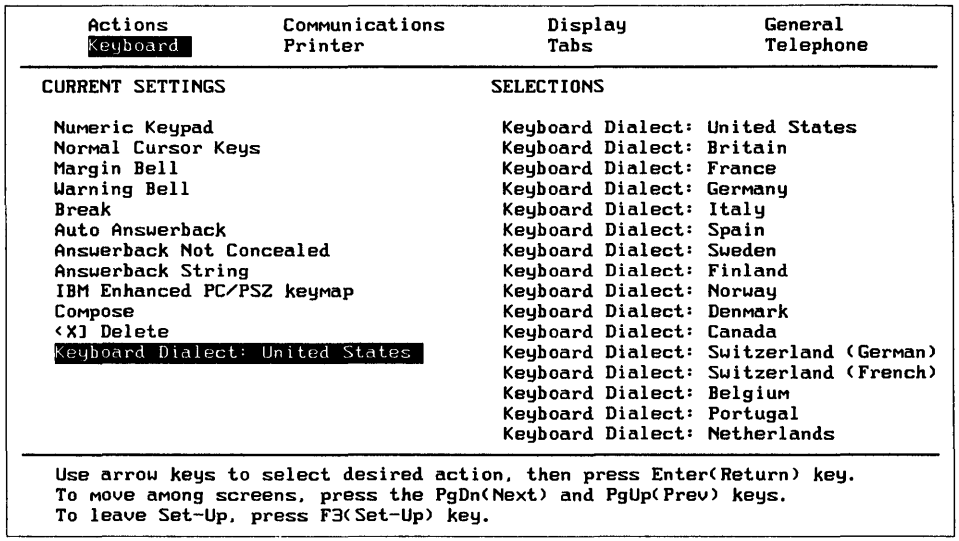
- Norway
- Denmark
- Canada
- Switzerland (German)
- Switzerland (French)
- Belgium
- Portugal
- Netherlands

To select a keyboard dialect, follow these steps:

1. Use [Pg Up] and [Pg Dn] to move the highlighted video bar to the Keyboard selection on the top of the Communications Set-Up screen. The Keyboard Set-Up screen appears
2. Use ⇩ to move the video bar to the keymap choice in the CURRENT SETTINGS column.

   For example, use ⇩ to move the video bar to the Keyboard Dialect: United States selection. A list of keymaps appears in the SELECTIONS column, as shown in Figure 2–6.

Figure 2–6 Sample Keyboard Dialect Selections



3. If the default Keyboard Dialect is the one you want, do not make any changes. To choose a different Keyboard Dialect, press ⇨ to move the video bar to the SELECTIONS column, then press ⇧ and ⇩ to highlight your selection.

   For example, to select the Keyboard Dialect: Canada selection, move the video bar to that choice in the CURRENT SELECTIONS column.

4. To finalize your selection, press Return. The name of the selected keymap appears in the CURRENT SETTINGS column.

   To move back to the CURRENT SETTINGS column without making a selection, press ⇦.

Selecting a Character Set

To replace the MCS with the character set for your country, use these instructions:

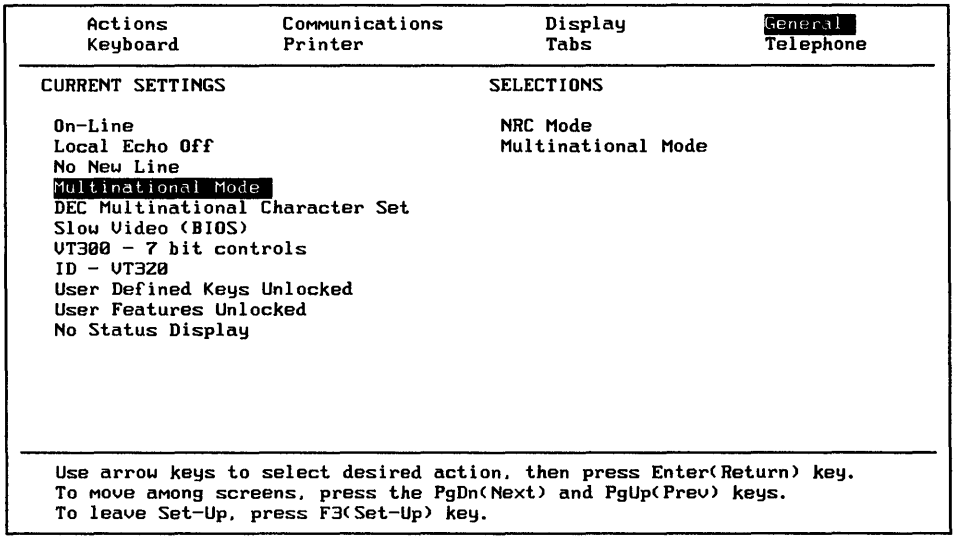
1. Use [Pg Up] and [Pg Dn] to move the highlighted video bar at the top of the screen to the General choice. The General Set-Up screen appears, as shown in Figure 2–7.

Figure 2–7 Sample General Set-Up Screen

```
   Actions          Communications      Display          General
   Keyboard         Printer             Tabs             Telephone

CURRENT SETTINGS
 ▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮
 On-Line
 Local Echo Off
 No New Line
 Multinational Mode
 DEC Multinational Character Set
 Slow Video (BIOS)
 VT300 - 7 bit controls
 ID - VT320
 User Defined Keys Unlocked
 User Features Unlocked
 No Status Display

 Use arrow keys to select desired action, then press Enter(Return) key.
 To move among screens, press the PgDn(Next) and PgUp(Prev) keys.
 To leave Set-Up, press F3(Set-Up) key.
```

2. Use ⇩ to move the video bar to the Multinational Mode choice in the CURRENT SETTINGS column. Two choices (NRC and Multinational Mode) appear in the SELECTIONS column, as shown in Figure 2–8.

Figure 2–8 Sample Character Mode Selection



3. Use ⇨ to move to the SELECTIONS column. Then, use ⇧ and ⇩ to move the video bar to the NRC Mode option. After you highlight NRC Mode, press Return. To move back to the CURRENT SETTINGS column without making a selection, press ⇦.

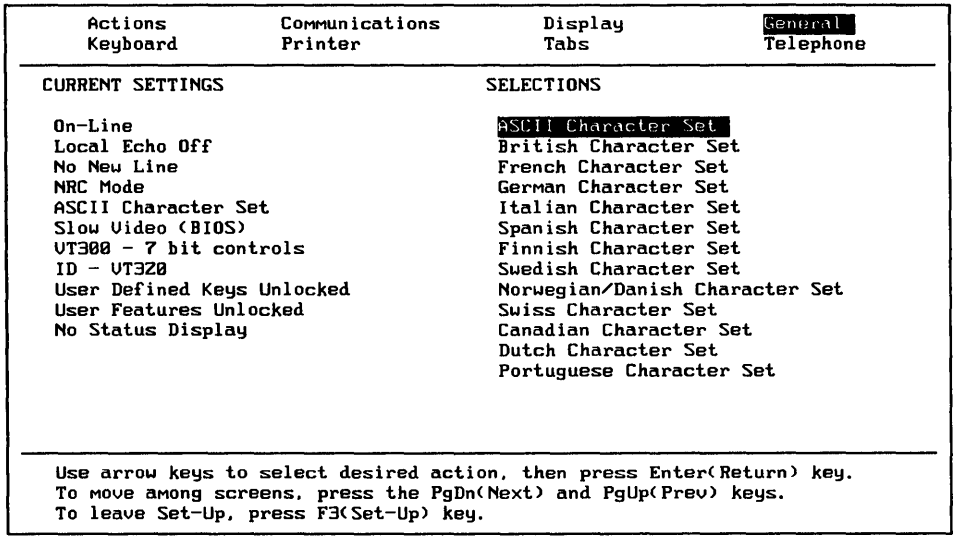
   The setting below NRC Mode in the CURRENT SETTINGS column changes from DEC Multinational Character Set to ASCII Character Set.

4. Use ⇩ to move to the ASCII Character Set option in this column.

   A list of the character sets appears in the SELECTIONS column.

5. Use [⇒] to move the video bar to the SELECTIONS column. The video bar highlights the ASCII Character Set selection, as shown in Figure 2–9.

Figure 2–9 Sample Character Set Selection Screen



6. Use [⇑] and [⇓] to highlight the character set that you want to choose and press [Return]. Your selection appears in the CURRENT SETTINGS column.

Note

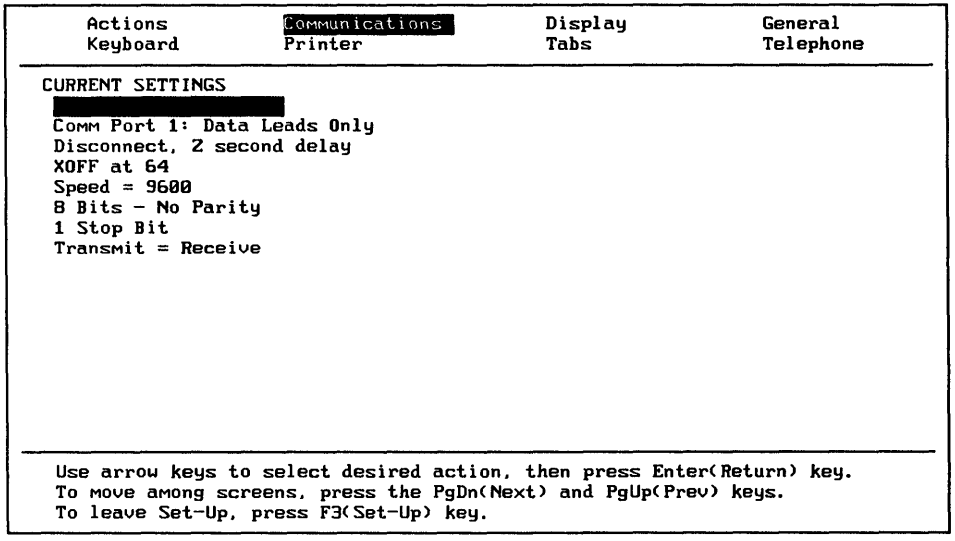
If you select an NRC and your host application does not display the proper characters, try using the DEC Multinational or ISO Latin-1 character set.

Selecting a Preferred Service

One of the parameters that SETHOST requires is the name of the remote system to connect to. This system is your preferred service and it can be a terminal server or a host node. To select this service, first access the Communications Set-Up screen.

To access this screen, use Pg Up or Pg Dn to move the highlighted video bar to the Communications option at the top of the screen. The Communications Set-Up screen appears, as shown in Figure 2–10.

Figure 2–10 Sample Communications Set-Up Screen



Selecting the Network Communications Port

Before selecting a service type and a service, follow these steps to access the network communications port:

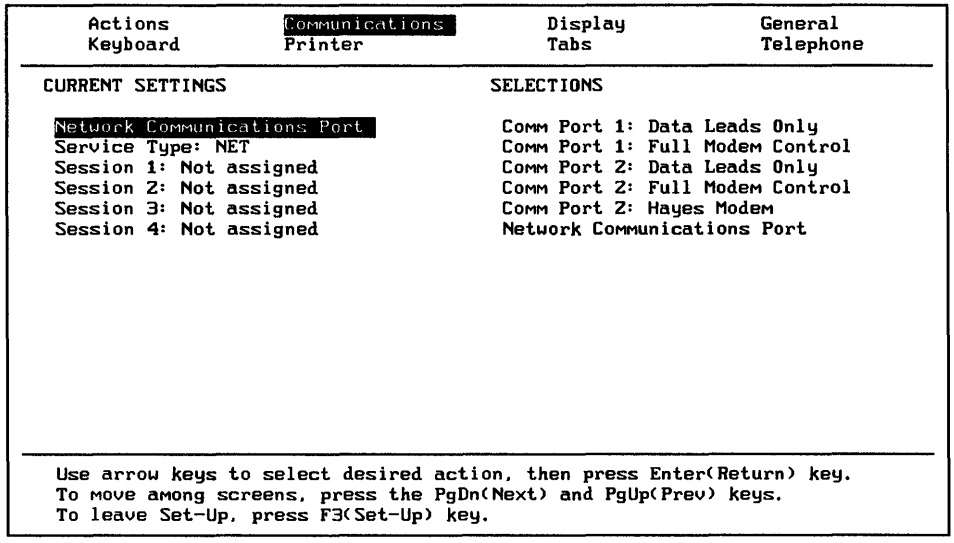
1. Use ⇩ to move the highlighted video bar to the Comm Port 1 option in the CURRENT SETTINGS column. A series of choices appears in the SELECTIONS column, as shown in Figure 2–11.

Figure 2–11 Sample Network Communications Port Screen 1

```
Actions          Communications    Display          General
Keyboard         Printer           Tabs             Telephone

CURRENT SETTINGS                   SELECTIONS

 Comm Port 1: Data Leads Only       Comm Port 1: Data Leads Only
 Disconnect, 2 second delay         Comm Port 1: Full Modem Control
 XOFF at 64                         Comm Port 2: Data Leads Only
 Speed = 9600                       Comm Port 2: Full Modem Control
 8 Bits - No Parity                 Comm Port 2: Hayes Modem
 1 Stop Bit                         Network Communications Port
 Transmit = Receive

 Use arrow keys to select desired action, then press Enter(Return) key.
 To move among screens, press the PgDn(Next) and PgUp(Prev) keys.
 To leave Set-Up, press F3(Set-Up) key.
```

2. Use ⇨ to move the video bar to the SELECTIONS column.
3. Use ⇩ to move to the Network Communications Port option; then press Return. The first choice in the CURRENT SETTINGS column changes, as shown in Figure 2–12.

Figure 2–12 Sample Network Communications Port Screen 2



Selecting a Network Service Type

Before specifying a preferred node name, specify the type of network connection that you want to make (unless you want to use the defaults). You have four choices:

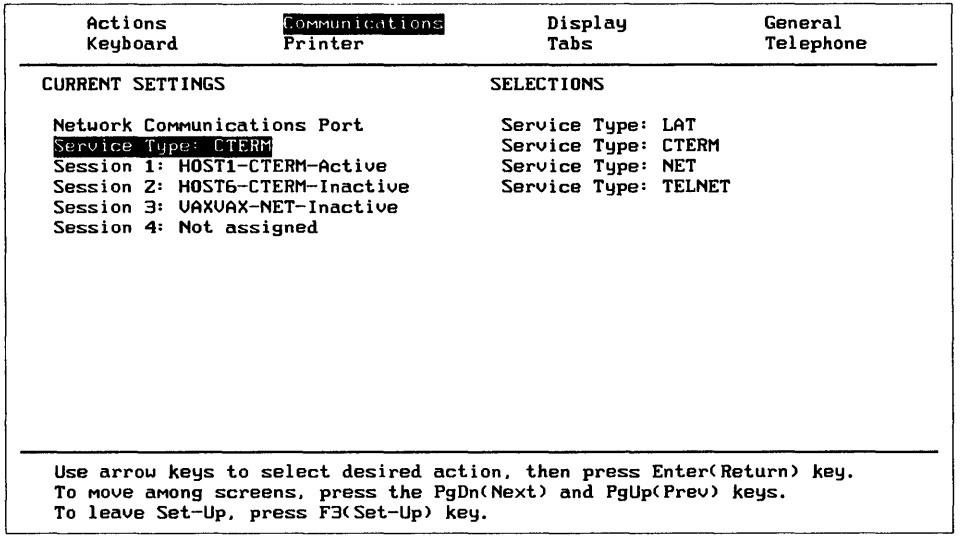
- LAT—Use the LAT service to connect to a system on your local area network. Use LAT only for Ethernet connections.
- CTERM—Use the CTERM service to connect to a DECnet system or a Token Ring network.
- NET—Use the NET service to connect to the network in the fastest way. For DECnet connections, SETHOST tries the LAT service first, then the CTERM service if LAT fails. For TCP/IP connections, SETHOST tries the TELNET service. For Token Ring connections, NET tries CTERM or TELNET.

  You can also select NET if you are unsure which method to use.
- TELNET—Use the TELNET service to connect to other systems running the TCP/IP protocol or a Token Ring network.

To select a service type, follow these steps:

1. Use ⇩ to move the video bar to the Service Type choice in the CURRENT SETTINGS column.
   The Communications Set-Up screen appears, as shown in Figure 2–13.

Figure 2–13 Sample Service Type Selection Screen



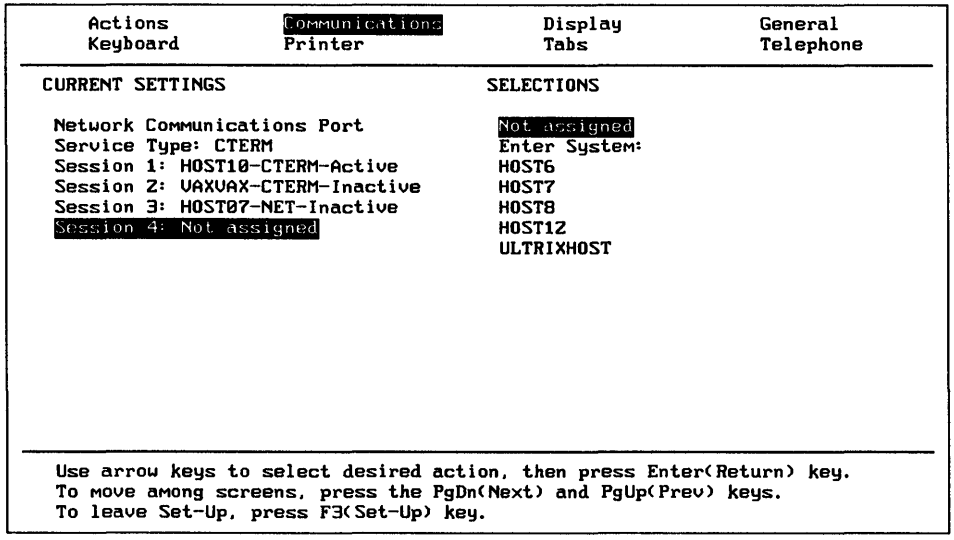
2. Use ⇨ to move the video bar to the SELECTIONS column; then use ⇧ and ⇩ to highlight one of the options. After you highlight your option, press Return to select it. The service type that you select appears in the Service Type field. You can also use ⇦ to move back to the CURRENT SETTINGS column without making a selection.

Selecting a Service Name

After selecting a service type, select the name of the service to connect to. To select a service to use for a network connection, follow these steps:

1. Use ⇩ to move the video bar to the Service Name: Not Assigned choice in the CURRENT SETTINGS column.
2. Press ⇨ to display a list of available services.
   The services appear in the SELECTIONS column, as in Figure 2–14.

Figure 2–14 Sample Network Service Selection Screen



3. The list of services depends on which service type you chose. Highlight the service you want to access by scrolling through the list with ⇧ and ⇩. Additionally, you can highlight the desired service by typing the first letter of its node name. This action highlights the first entry in the list that starts with the same letter.

4. After you highlight your option, press Return to enter the option as an assigned service.

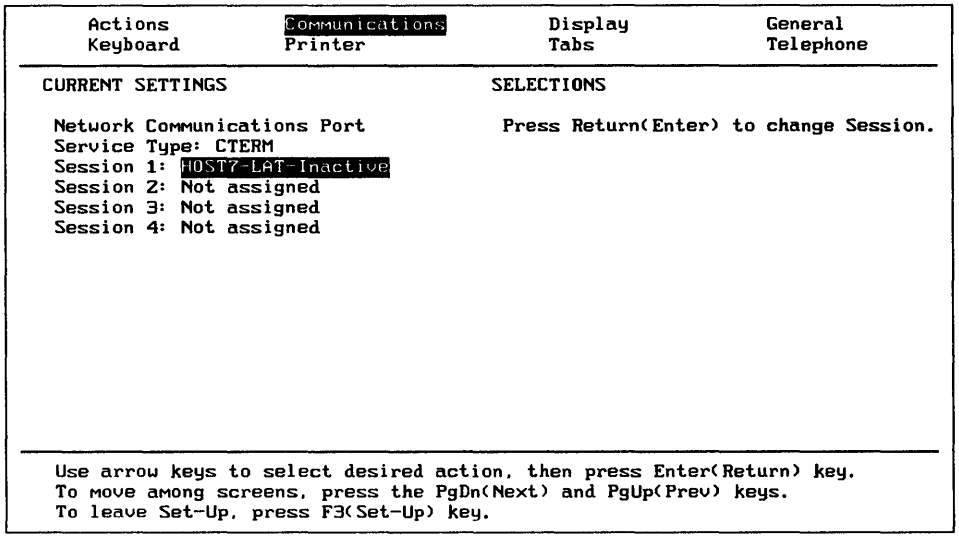
   If you highlighted "Enter System," a cursor blinks to prompt you to enter a system name. Type in the name of a system and press Return.

   If you are using LAT, type in a service name. For example, HOST3.
   If you are using DECnet, type in a node name or node number. For example, HOST7 or 9.175.
   If you are using TELNET, type in a host name or Internet Protocol (IP) address. For example, einstein.princeton.edu.

   That system is entered as the assigned service and the name is displayed in the CURRENT SETTINGS column.

Figure 2–15 displays a sample Communications screen that specifies a LAT connection to the node named HOST7.

Figure 2–15 Sample Completed Network Service Selection Screen



The SETHOST utility marks the service inactive until you leave the Set-Up mode. After you leave the Set-Up mode, SETHOST marks the service active and you can use it.

If you do not want to select a service, use ⇐ to move back into the CURRENT SETTINGS column.

Note

The following message appears when the LAT service table is too small to hold all available LAT services:

```
Warning: LAT service table full.
Some services may be missing
```

If this message appears, increase the size of the LAT service table to view all available services. (Refer to Adding a Local Area Transport Service to the Service Table.)

What to Do If a Service Does Not Appear

The list of services that appear on the Communications screen varies depending on the service type that you selected. If you chose the LAT service, the list of entries in the LAT service table appears. If you chose the CTERM service, the list contains the node names defined in the DECnet database. If you chose the TELNET service, the list contains the host names in the HOSTS database. If you chose NET and you have a DECnet connection, the list is a combination of the entries in the LAT service table and the DECnet database. If you chose NET and you have a TCP/IP connection, the list contains only the host names in the HOSTS database. If a node or service does not appear in one of these lists, the probable reasons are:

- For LAT—The LAT service table is too small or you did not define the service in the LATCP preferred services database.
- For CTERM—You did not define the node name in the DECnet database.
- For TELNET—You did not define the host name in the HOSTS file database or the host name is unknown to the domain name resolver.

Adding a Local Area Transport Service to the Service Table

The LAT service table contains LAT services on your local area network from which your node receives a broadcast message. A LAT service can be any device using LAT, including remote host systems or terminal servers (such as a server connected to a modem pool). Because LAT uses a default service table size, the service you specify may not be in the table.

To add services to the LAT service table, use the LATCP utility. You can also use LATCP to define a cluster alias or a terminal server. The simplest way to add a service is to issue the LATCP ADD command.

Follow these steps to add your service to the LAT service table:

1. Invoke the LATCP utility by entering the following command at the DOS prompt:

   ```
   C:\>LATCP Return
   ```

2. To list the available services, enter this command at the LATCP> prompt:

   ```
   LATCP>ADD Return
   ```

   When LATCP finishes listening for the service announcements, a screen appears listing the available LAT services. You can stop LATCP at any time by pressing the space bar.

3. To select services from this listing, move the highlighted video bar to the service using ⇧ and ⇩ and then press Return. The selected service appears on the left side of the screen.

4. After selecting one or more services, press ESC to leave the selection screen and return to the LATCP prompt.

   A message appears at the bottom of the screen reminding you to restart LAT when you exit. To save your changes, press Return.

5. To exit LATCP, type the following command at the LATCP> prompt:

   ```
   LATCP>EXIT Return
   ```

6. After exiting the LATCP program, reload LAT so that the new entries appear in the LAT service table. To reload LAT, type the following command at the DOS prompt:

   ```
   C:\>LAT Return
   ```

For a complete description of LATCP, see the PATHWORKS for DOS, *Client Commands Reference* or type LATCP HELP at the DOS prompt.

Defining a DECnet Node for CTERM

To add a node to the DECnet database, use the NCP DEFINE NODE command. This command has the following syntax:

NCP DEFINE NODE *node-address* NAME *node-name*

- The NODE *node-address* parameter is the unique numeric identifier for the node that includes an area number and a node number.
- The NAME *node-name* parameter is an identification string containing from one to six alphanumeric characters.

The following command enters a host system named HOST7 with the address 2.65 in the DECnet database:

```
C:\>NCP DEFINE NODE 2.65 NAME HOST7 Return
```

For more information, type NCP HELP at the DOS prompt. For a complete description of the DEFINE NODE command, see the PATHWORKS for DOS, *DECnet Network Management Guide*.

Defining a Host Name for TELNET

When there is a reference to a host name in SETHOST, TELNET tries to look up the host name in the local HOSTS file. If that attempt fails, then TELNET tries to find the host name over the network by sending a query to the domain name resolver. If TELNET does not find the host name, it cannot connect to that host.

Refer to the PATHWORKS for DOS, *TCP/IP User's Reference* for information on creating a HOSTS file database.

Saving Your Set-Up File

You specify Set-Up parameters (such as keyboard mappings, node names, and service types) to customize your session. You can create these parameters without saving them, or you can save them in a Set-Up file to make the parameters permanent. The SETHOST utility reads this Set-Up file and sets the parameters accordingly when you start the program. By default, SETHOST names the file SETHOST.DAT and stores in the DECnet directory.

You can also have multiple Set-Up files, each specifying different settings. This feature is useful if you want to connect to a number of different hosts in which each session has its own unique terminal-emulation characteristics. If SETHOST does not find the SETHOST.DAT file or any other Set-Up files, it uses the default settings to configure SETHOST.

To save your Set-Up file, follow these directions:

1. Press Pg Dn or Pg Up to move the video bar to the Actions screen.

2. Use ⇩ to move the video bar to the Save Set-Up Parameters option in the ACTIONS column. The default file name SETHOST.DAT appears in the SELECTIONS column, as shown in Figure 2–16.

Figure 2–16 Sample Actions Set-Up Screen

```
Actions          Communications      Display          General
Keyboard         Printer             Tabs             Telephone

ACTIONS

 Reset Terminal                      C:\DECNET\SETHOST.DAT
 Clear Display
 Clear Communications
 Recall Set-Up Parameters
 Save Set Up Parameters
 Set Default Set-Up Parameters
 Show Status
 To MS-DOS
 Exit SETHOST

 Use arrow keys to select desired action, then press Enter(Return) key.
 To move among screens, press the PgDn(Next) and PgUp(Prev) keys.
 To leave Set-Up, press F3(Set-Up) key.
```

To save Set-Up parameters in a file other than the SETHOST.DAT file, move the cursor to the SELECTIONS column using ⇨ and enter a new file name.

3. Press Return to save the settings in the specified Set-Up file.

If you use the command line method and your own Set-Up file, the following syntax forces SETHOST to use your file:

SETHOST /S=*startup.dat*

The *startup* variable represents the name of the file you created, and *dat* represents the file type. The file type can be any 3-character string.

For example, if you have a Set-Up file named MYSETUP.DAT, you can enter the file name with the SETHOST command and the /S switch as follows:

```
C:\>SETHOST /S=MYSETUP.DAT Return
```

This command forces SETHOST to use the settings in the MYSETUP.DAT file.

Changing Selections of Any Set-Up Screen

To change a Set-Up screen selection, follow these steps:

1. Start a SETHOST session, then press F3 to access the Set-Up screens.
2. Press Pg Dn or Pg Up to select the Set-Up screen.
3. Press ⇧ or ⇩ to view the current settings.

   The SETHOST utility displays the current settings in the left column and possible setting selections in the right column.

   Note

   The Actions Set-Up screen displays a set of prompts or directions instead of selections in the right column. Follow those directions.

4. Press ⇨ to move the video bar to the SELECTIONS column to view the possible selections.

   The SETHOST utility now highlights the setting in the CURRENT SETTINGS column, indicating the setting that you are changing.
5. If there are multiple choices, use ⇧ or ⇩ to move the video bar to your selection.
6. Press Return to finalize the selection. The new selection appears in the CURRENT SETTINGS column. If you do not want to choose from the SELECTIONS column, use ⇦ to move the video bar back to the CURRENT SETTINGS column.

Leaving Set-Up Mode

If you leave Set-Up without saving your changes, those changes are effective only during your current session. To permanently save your changes, select the Save Set-Up Parameters option from the Actions Set-Up screen.

To leave Set-Up, press F3 again.

3

Customizing Your SETHOST Session

To customize your SETHOST session, use the Set-Up screens. There are eight Set-Up screens, each of which offers numerous settings as well as a set of default parameters:

- Actions
- Communications
- Display
- General
- Keyboard
- Printer
- Tabs
- Telephone

This chapter contains the following information:

- A brief description of each Set-Up screen's capabilities (see Table 3–1)
- A detailed description of how to use each Set-Up screen to customize your session (see the sections Defining the Communications Environment through Determining Telephone Dialing)
- A description of environment variables and how to use them (see Tailoring Your SETHOST Environment)

Using the SETHOST Set-Up Screens

From each Set-Up screen, you can:

- View the current selections and settings.
- Change the current settings.
- Use [Pg Dn] or [Pg Up] to access any of the other Set-Up screens.

You can also save or recall settings from the Actions Set-Up screen.

Table 3–1 briefly describes what you can do with each of the Set-Up screens. The remainder of this chapter describes each Set-Up screen in detail.

Table 3–1 SETHOST Set-Up Screen Capabilities

To Do the Following...	Use This Set-Up Screen...
Reset the terminal.	Actions
Clear the screen display.	Actions
Clear communications.	Actions
Save and recall Set-Up settings.	Actions
Set default parameters.	Actions
Show status.	Actions
Spawn to DOS.	Actions
Exit SETHOST.	Actions
Define the communications environment.	Communications
Select LAT, CTERM, and TELNET services.	Communications
Define the screen display.	Display
Select local echo setting.	General
Select character sets.	General
Select terminal type.	General
Define keyboard operating characteristics.	Keyboard
Define printer operating characteristics.	Printer
Define tab settings.	Tabs
Enter the telephone numbers to use with a modem.	Telephone

Using Keyboard Keys in Set-Up Screens

A description of the keys you can use with each Set-Up screen appears at the bottom of each screen.

Table 3–2 lists the Set-Up screen functions keys and their functions.

Table 3–2 Set-Up Screen Function Keys

To Perform This Function...	Press This Key...
Display the next Set-Up screen.	Pg Dn On the LK250 keyboard, press Next
Display the previous Set-Up screen.	Pg Up On the LK250 keyboard, press Prev
Make a selection from either side of the Set-Up screen (depending on which column you are using and which item is highlighted).	⇨ ⇦
Move up or down the list of selections in either column.	⇧ or ⇩
Finalize your choice.	Return
Exit the Set-Up screens.	F3

Using the Actions Set-Up Screen

When you enter Set-Up, SETHOST displays the Actions Set-Up screen (see Figure 3–1).

Figure 3–1 Sample Actions Set-Up Screen

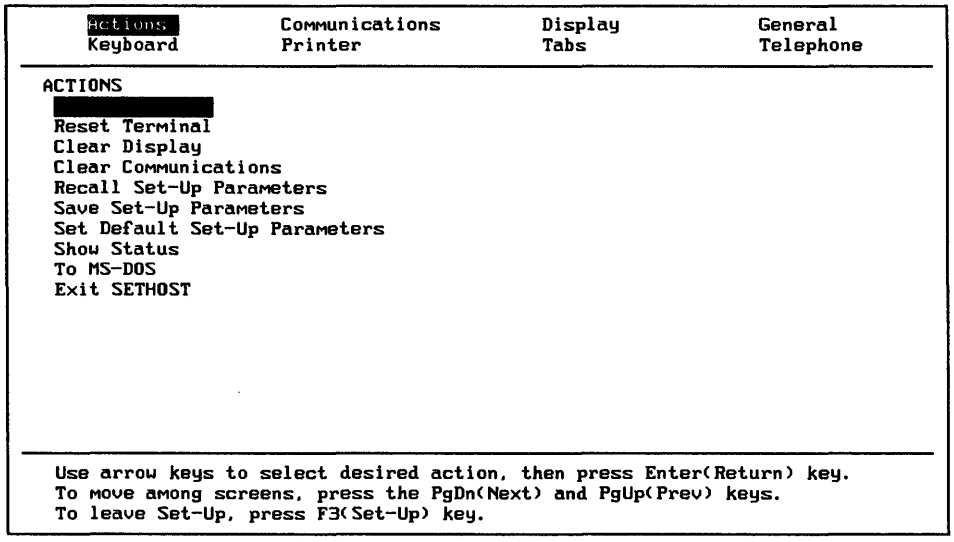


Table 3–3 identifies the Actions Set-Up screen settings and when to use them:

Table 3–3 Actions Set-Up Settings

To Do the Following...	Choose This Setting...
Reset operating settings to the power-up default state. (This setting does not affect communications or user-defined keys.)	Reset Terminal
Clear the screen and send the cursor to home position when you leave Set-Up.	Clear Display
Clear the communication port and stop any print operation in progress.	Clear Communications
End printer controller mode.	Clear Communications

(continued on next page)

Table 3–3 (Cont.) Actions Set-Up Settings

To Do the Following...	Choose This Setting...
End all network sessions.	Clear Communications
Replace existing Set-Up settings with the values saved in the specified Set-Up file. If the Set-Up file you want does not appear in the right column, change the file name by pressing ⇨, entering a new file name, and pressing Return.	Recall Set-Up Parameters
Save the Set-Up settings in the file you specify. To change the file name, press ⇨, enter a new file name, and press Return.	Save Set-Up Parameters
Replace current Set-Up settings with default settings.	Set Default Set-Up Parameters
Display information on the state of such items as modems, file sending or receiving, and communications ports.	Show Status
Leave SETHOST temporarily to use the DOS operating system. (This choice has the same effect as pressing Ctrl/F9. When you finish using DOS commands, return to your session by typing EXIT at the DOS prompt.)	To DOS
Leave SETHOST and return to the prompt you were using previously. For example, if you were using MS-Windows, you return to the window you were using. Selecting this has the same effect as pressing Ctrl/F10.	Exit SETHOST

Limitations on DOS Processes

When you create a DOS process from within a SETHOST session, be aware of the following limitations:

- Do not run terminate-and-stay-resident programs (TSRs), such as DOS PRINT.

  If you run a TSR while using the **To DOS** feature, it can cause memory fragmentation. This fragmentation can cause programs to fail if they are run after SETHOST exits.

  Note that the MODE and KEYB commands are TSRs, under certain conditions.

- Environment variables that are set in a DOS process are lost when you exit the process to return to SETHOST.

- SETHOST does not work when run with a keyboard buffer that has been extended using the DECMODE utility from the PATHWORKS for DOS Client kit. Reset the keyboard buffer to the default by using DECMODE m, 16, or use public domain utilities, such as BIGBUFF or KBFIX2, to extend the keyboard buffer.
- If there is not enough memory to run the COMMAND.COM file, the operating system displays a message.
- If you use a communications program, it may interfere with the SETHOST utility's control of the communications port. This condition occurs if both programs use the same communications port (either COM1 or COM2).

  To restore control to SETHOST, enter the Set-Up mode and select the Clear Communications setting from the Actions Set-Up screen. Additionally, you can change a communications setting (such as baud rate) and exit; then enter and change the setting back to its correct value.

Displaying Status Information

To display status information, use the Actions Set-Up screen. Table 3–4 lists the information that SETHOST displays when you select the Show Status setting.

Table 3–4 Show Status Settings

If You Select the Show Status Setting and You Are...	SETHOST Displays This Information...
Receiving a file	Idle or In Progress
Sending a file	Idle or In Progress
Printing	Normal Print Mode, Auto Print Mode, Printer Controller Mode
Using a modem	Data Set Ready or No Data Set Ready
Using a telephone mode	Manual or Automatic Answer
Using Insert/Replace mode	Insert or Replace
Using a keyboard map	The name of the current keyboard-mapping file. If SETHOST cannot find the keymap that you selected on the path, it defaults to a keyboard mapping for the IBM Personal Computer XT.
Using a character set	Character sets currently in use
Using communications	COM1, COM2, or Network Terminal Service

When you select the Show Status setting, SETHOST displays current settings. For example, SETHOST displays the settings in the following format (the settings can vary depending on your configuration):

```
SETHOST V4.0
Receive File:       Idle
Send File:          Idle
Printer:            Normal Print Mode
Telephone:          Manual Answer
Insert/Replace:     Replace
Keyboard Map:       KBDXT.KBD
Character Set:      ASCII
                    DEC Multinational
Communications:     Network:
                    MYNODE
```

Receiving a File (Session Logging)

Session logging allows you to capture in a file all the characters that a host system sends to your terminal. The host sends characters to the file one at a time with no error checking. To indicate whether session logging is active or idle, use the Show Status setting in the Actions Set-Up screen.

Follow these steps to use session logging to receive or log characters from the host into a file:

1. Enter Ctrl/F1. SETHOST displays this prompt at the bottom line of the screen:

   ```
   Receive file-name:
   ```

2. Type in a file name for storing characters to be logged in from a host.

   If the file name already exists, SETHOST displays another prompt asking you to do one of the following:

   - Replace an existing file
   - Append to an existing file
   - Cancel the file logging

   After you enter an acceptable file name, SETHOST displays the "Receive: On" message at the bottom of your screen.

Note

If using VT320 mode, the status display may overwrite this status line.

3. Enter Ctrl/F1 to stop the host from logging characters into the file. The SETHOST utility displays the "Receive: Off" message at the bottom of your screen.
4. Enter Ctrl/F3 to close the receive file.

Note

You can transfer only ASCII text files using session logging. To transfer other types of files, use the Network File Transfer (NFT) utility or the File Transfer Protocol (FTP) utility. The NFT utility works only with the DECnet transport and the FTP utility works only with the TCP/IP transport.

Refer to the PATHWORKS for DOS *DECnet User's Guide* for information on using the NFT utility and the PATHWORKS for DOS *TCP/IP User's Reference* for information on using the FTP utility.

Sending a File (Autotyping)

The Autotyping setting allows you to send characters from a file on your system to a host system. Your system sends characters to the file one at a time with no error checking. To indicate whether the autotyping setting is active or idle, use the Show Status setting in the Actions Set-Up screen.

When a file is sent to a host running VMS, first enable HOSTSYNC. For example:

```
$ SET TERMINAL/HOSTSYNC Return
```

This action helps prevent data overruns on a VMS host.

To send characters from a file to the host, as if you entered them from the keyboard, follow these steps:

1. Enter Shift/F1. SETHOST displays the following prompt on the bottom line of your screen:

   ```
   Send file-name:
   ```

2. Type in the file name you want to send to the host.

   File sending stops when SETHOST reaches the end of the file or when you enter Shift/F1 again.

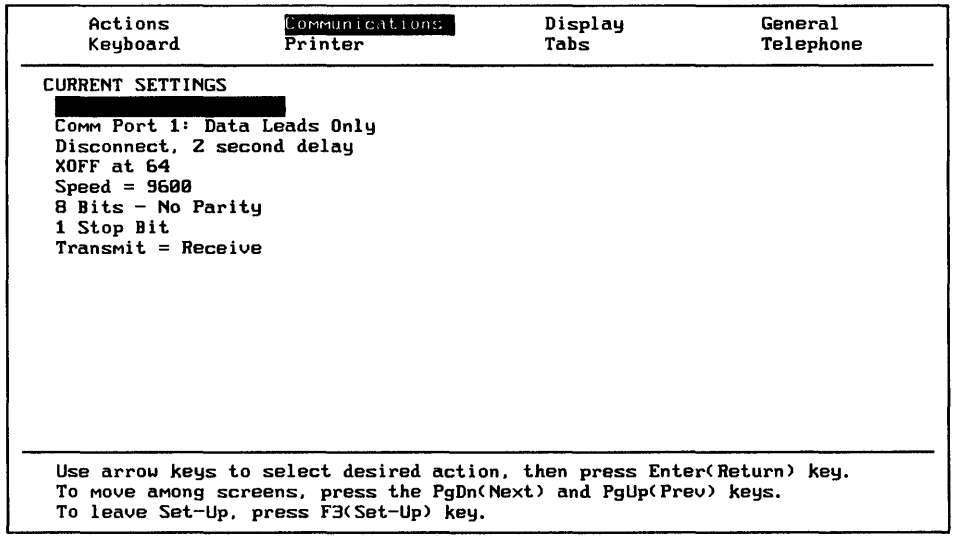
Defining the Communications Environment

To define the communications environment, you can set the following settings:

communications port
speed
parity
stop bits
XOFF information
transmit/receive information

The Communications Set-Up screen also allows you to select your network service (see Figure 3–2).

Figure 3–2 Sample Communications Set-Up Screen



The SETHOST utility allows you to connect to a host by using either a serial port or an Ethernet adapter. These devices connect your personal computer to different network services, depending on which services you define in the service table. (A service table is a list of the available nodes in the network.) The services you can connect to include terminal services on a host system or printer and modem services on a terminal server.

Table 3–5 lists what you can do with the Communications Set-Up screen when you establish asynchronous or network connections.

Table 3–5 Establishing Asynchronous or Network Connections

To Do the Following...	Choose This Setting...
Asynchronous Connections	
Make an asynchronous connection over COM1—data leads only.	Comm Port 1: Data Leads Only (default)
Make an asynchronous connection over COM1—full modem control.	Comm Port 1: Full Modem Control
Make an asynchronous connection over COM2—data leads only.	Comm Port 2: Data Leads Only
Make an asynchronous connection over COM2—full modem control.	Comm Port 2: Full Modem Control
Make an asynchronous connection over COM2—using a Hayes modem dialing string.	Comm Port 2: Hayes Modem
After you define the type of asynchronous connection, check the following settings:	
When using modem control, select the amount of time to wait before disconnecting the workstation from the communications line when the carrier detect is lost. Most countries use the 2-second delay; the United Kingdom uses the 60-millisecond (ms) delay.	Disconnect, 2-second delay (default) or Disconnect, 60-ms delay
Enable the automatic XON/XOFF flow control. For most applications, set XOFF at 64 bytes or 256 bytes. If you set XOFF greater than 256 bytes and have a buffer overflow problem, set XOFF to the next lower value.	Choose one of the following settings: XOFF at 64 (default) XOFF at 256 XOFF at 512 XOFF at 1024
Disable the automatic XON/XOFF flow control. Using No XOFF, SETHOST receives characters continuously from the host and some of those characters can be lost.	No XOFF

(continued on next page)

Table 3–5 (Cont.) Establishing Asynchronous or Network Connections

To Do the Following...	Choose This Setting...
Select the rate at which SETHOST receives or transmits characters.	Choose one of the following values: 50 300 2400 75 600 3600 110 1200 4800 134.5 1800 9600 150 2000 19200 The default is 9600 baud.
Select the character format used for communication with a host computer.	Choose one of the following settings: 8 bits - No Parity (default) 8 bits - Even Parity 8 bits - Even Parity - No Check 8 bits - Odd Parity 8 bits - Odd Parity - No Check 7 bits - No Parity 7 bits - Even Parity 7 bits - Even Parity - No Check 7 bits - Odd Parity 7 bits - Odd Parity - No Check 7 bits - Mark Parity 7 bits - Space Parity
Select the number of stop bits used for communicating with the host.	1 Stop Bit (default) or 2 Stop Bits
Set the transmit rate equal to the receive rate.	Transmit = Receive (default)
Set the transmit rate to 1200.	Transmit = 1200
Network Connections	
Make a connection over the network communications port.	Network Communications Port
After you select a network port connection, select one of the following service types:	
Choose LAT as the service type. The LAT service type tries only LAT connections.	LAT

(continued on next page)

Table 3–5 (Cont.) Establishing Asynchronous or Network Connections

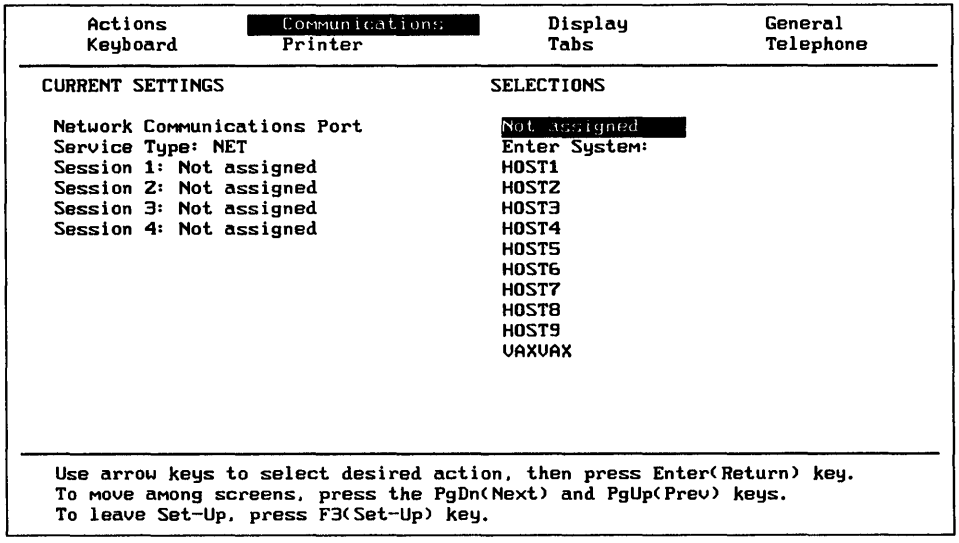
To Do the Following...	Choose This Setting...
Choose CTERM as the service type. The CTERM service type tries only CTERM connections.	CTERM
Choose NET as the service type. The NET service type connects to the network in the fastest way. For DECnet connections, the NET service type tries the LAT service first, then the CTERM service if the LAT connection fails. For TCP/IP connections, the NET service type tries only a TELNET connection. For Token Ring connections, NET uses CTERM or TELNET.	NET
Choose TELNET as the service type. The TELNET service type tries only TELNET connections over the Token Ring or TCP/IP transport.	TELNET

Changing Your Network Terminal Service

To select a service to use for a network connection, follow these steps:

1. Move the video bar to Session 1 in the CURRENT SETTINGS column.
2. Press ⇨ to display a list of available services. The services appear in the SELECTIONS column (see Figure 3–3).

Figure 3–3 Sample Network Service Selection Screen



3. Initially, this service table contains ten service names from your local area network. Highlight the service you want to access by scrolling through the list using ⇧, ⇩, or type the first letter of its name and use ⇩ to highlight the name you want.

4. To enter the highlighted service as an assigned service, press Return. The service you select appears in the CURRENT SETTINGS column.

   The SETHOST utility marks the service inactive until you leave the Set-Up mode. After you leave the Set-Up mode, the service becomes active and is available for use. If you do not wish to select a service, use ⇦ to move back into the CURRENT SETTINGS column.

You can have up to four network sessions. Use ⇑ and ⇓ to select the session for which you want to assign a service, then press ⇨. Available services (based on the service type you chose) appear in the SELECTIONS column. To move between sessions, press F4.

Customizing Your Display Characteristics

To define screen display characteristics, such as the number of columns, auto wrap, background, and cursor style, use the Display Set-Up screen (see Figure 3–4).

Figure 3–4 Sample Display Set-Up Screen

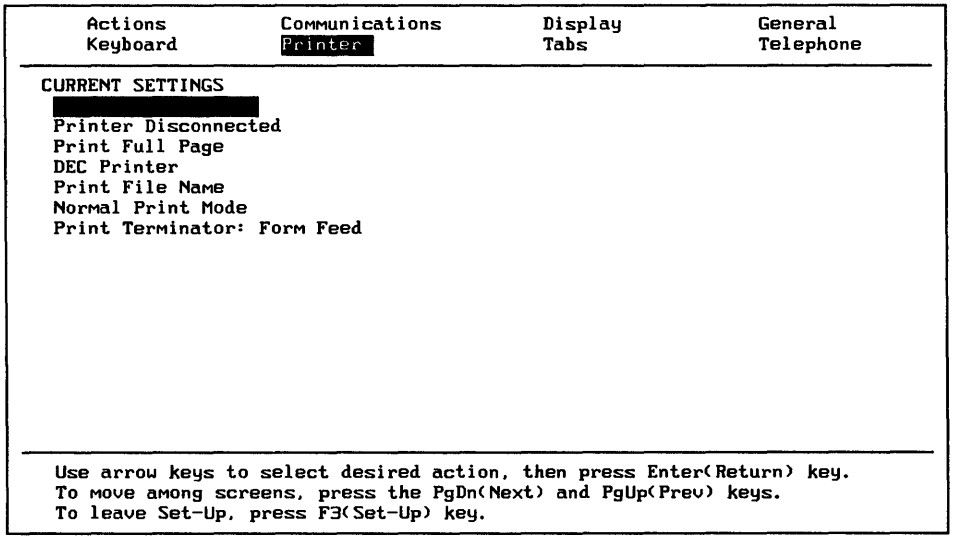


Table 3–6 lists the characteristics you can customize using the Display Set-Up screen.

Table 3–6 Setting Display Characteristics

To Do the Following...	Choose This Setting...
Select an 80-column screen for text. (Changes take effect when you exit Set-Up.)	80 Columns (default)
Select 132-column screen for text. (Changes take effect when you exit Set-Up.)	132 Columns
If your video adapter supports 132-column mode, see Section 3.10 for information about defining SETHOST environment variables.	
If you select 132 columns, be aware that some personal computer video adapters display only 80 columns at a time. This means that the screen must be shifted to display the rest of the data. To move between the 80 column displays, press Ctrl/F8.	
Note: Screen shifting does not work on IBM monochrome display adapters or Hercules graphics cards.	
Turn automatic wrap on. The Auto Wrap On setting causes characters that reach the right margin to automatically display in the first character position of the next line.	Auto Wrap On
Turn automatic wrap off. The Auto Wrap Off setting causes characters that go beyond the right margin to overwrite the last character position of the current line.	Auto Wrap Off (default)
Select a dark background as the screen display type.	Dark Background (default)
Select a light background as the screen display type.	Light Background
Select a block cursor as the cursor style.	Block Cursor (default)
Select an underline cursor as the cursor style.	Underline Cursor
Select visible cursor as the cursor style.	Visible Cursor (default)
Select invisible cursor as the cursor style.	Invisible Cursor

(continued on next page)

Table 3–6 (Cont.) Setting Display Characteristics

To Do the Following...	Choose This Setting...
Select monochrome as the type of video adapter. The Monochrome setting produces more legible text on some portable displays.	Monochrome
Select color as the type of video adapter. If you select the Color setting, you can determine the degree of intensity for the character display. Black is the darkest and intense white is the lightest.	Color (default)
Select the background color.	Choose one of these settings: Black Blue Green Cyan Red Magenta Brown White
Select the reverse video color.	Choose one of these settings: Black Blue Green Cyan Red Magenta Brown White

(continued on next page)

Table 3–6 (Cont.) Setting Display Characteristics

To Do the Following...	Choose This Setting...
Select the color for bolding.	Choose one of these settings: Black Blue Green Cyan Red Magenta Brown White Dark Gray Light Blue Light Green Light Cyan Light Red Light Magenta Yellow Intense White
Select the foreground color.	Choose one of these settings: Black Blue Green Cyan Red Magenta Brown White Dark Gray Light Blue Light Green Light Cyan Light Red Light Magenta Yellow Intense White

(continued on next page)

Table 3–6 (Cont.) Setting Display Characteristics

To Do the Following...	Choose This Setting...
Select the underline color.	Choose one of these settings: Black Blue Green Cyan Red Magenta Brown White Dark Gray Light Blue Light Green Light Cyan Light Red Light Magenta Yellow Intense White

Displaying Line Attributes

The SETHOST utility displays certain line attributes differently than dedicated terminals display them.

Double-width characters appear as the character followed by a space. For example:

```
t e s t
```

Double height/double-width characters appear as the character followed by a space, with a blank line inserted before the next line of characters. For example, the word "test" entered on two separate lines would appear like this:

```
t e s t

t e s t
```

Displaying Character Attributes

Underlined characters appear without underlining on a color adapter but are a different color than background text. To change the underline color, use the Display Set-Up screen.

On a monochrome adapter, underlined characters appear as underlined. However, if the characters are also in reverse video, they appear in reverse video with no underlining.

Note

Because of differences in how different video cards handle video attributes, you might have to adjust the values SETHOST uses for COLOR or MONO attributes. Refer to Tailoring Your SETHOST Environment for information on the NVTCOLOR and NVTMONO environment variables that control these attributes.

Setting General Operating Characteristics

To set general operating characteristics, such as local echo, character sets, and terminal type, use the General Set-Up screen (see Figure 3–5).

Figure 3–5 Sample General Set-Up Screen

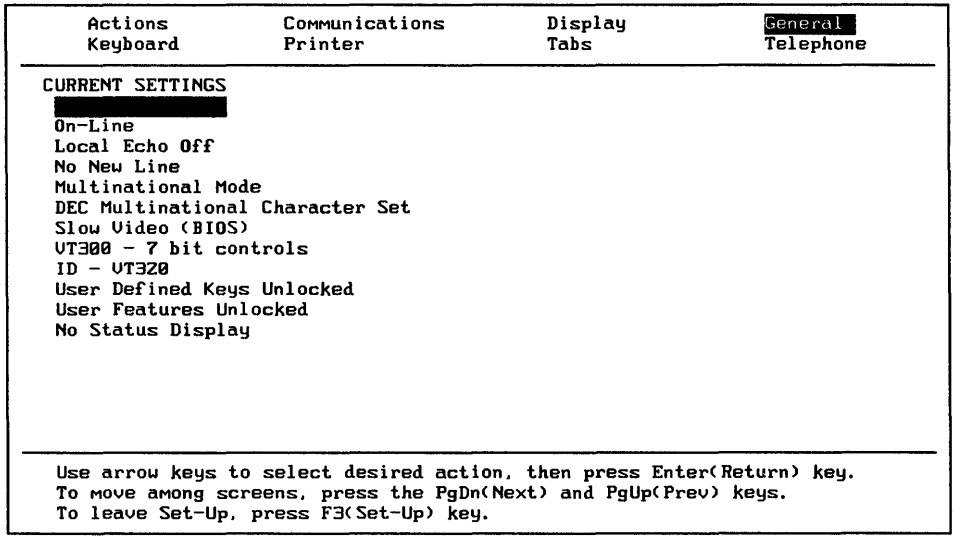


Table 3–7 lists the characteristics that you can set by using the General Set-Up screen.

Table 3–7 Setting General Operating Characteristics

To Do the Following...	Choose This Setting...
Select local as the mode of operation. The Local setting stops communication between your workstation and the host. Data entered at the keyboard goes directly to the workstation screen.	Local

(continued on next page)

Table 3–7 (Cont.) Setting General Operating Characteristics

To Do the Following...	Choose This Setting...
Select on-line as the mode of operation. The On-Line setting allows your workstation to communicate with the host.	On-Line (default)
Enable the local echo. The Local Echo On setting directs characters from the keyboard to the screen as well as to the host. Use this setting if the host does not send characters back to the screen.	Local Echo On
Disable the local echo. The Local Echo Off setting directs characters from the keyboard only to a host. The host in turn may or may not send the characters back to the screen.	Local Echo Off (default)
Select carriage return as the result of pressing the Return key. Received carriage returns do not cause a new line operation.	No New Line (default)
Select combination carriage return and line feed as the result of pressing Return. Received carriage returns cause a new line operation.	New Line
When SETHOST is in numeric keypad mode, this setting affects the Enter and Return keys in the same way.	
Select multinational as the character set mappings category.	Multinational Mode (default) If you choose the Multinational Mode setting, choose one of the following 8-bit character sets (both include the 7-bit ASCII character set): ISO Latin-1 Character Set DEC Multinational Character Set (default)

(continued on next page)

Table 3–7 (Cont.) Setting General Operating Characteristics

To Do the Following...	Choose This Setting...
Select NRC as the character set-mapping category.	NRC Mode The NRC Mode setting allows you to choose one of the following 7-bit National Replacement Character Sets: ASCII Character Set (default) British Character Set French Character Set German Character Set Italian Character Set Spanish Character Set Finnish Character Set Swedish Character Set Norwegian/Danish Character Set Swiss Character Set French Canadian Character Set Dutch Character Set Portuguese Character Set
Select slow video as the method of displaying characters. The Slow Video (BIOS) setting uses BIOS INT 10 routines to display characters. Slow video mode is slower than Fast video mode, but it is compatible with all supported personal computers.	Slow Video (BIOS) (default)
Select fast video as the method of displaying characters. The Fast Video Mode setting writes characters directly to video memory, but it is not compatible with some personal computers.	Fast Video
Identify the type of emulation you want.	Choose one of the following settings: VT300 - 8-bit controls VT300 - 7-bit controls (default) VT100 VT52

(continued on next page)

Table 3–7 (Cont.) Setting General Operating Characteristics

To Do the Following...	Choose This Setting...
Identify the type of terminal to emulate.	Choose one of the following settings: ID-VT320 (default) ID-VT220 ID-VT102 ID-VT101 ID-VT100
Change user-defined key (UDK) definitions. Use a 7-bit ST (ESC\) to terminate UDK definitions.	User-Defined Keys Unlocked (default)
Lock user-defined key (UDK) definitions so that users cannot change them.	User-Defined Keys Locked
Change any of the following user features: Auto Repeat Light/Dark Screen Tab Stops If you select the User Features Locked setting, you cannot change certain features. If you select the User Features Unlocked setting, you can change certain user features. Some software applications expect to control these user features. If this applies to your software application, select the User Features Unlocked setting.	User Features Unlocked (default)
Lock the following user features: Auto Repeat Light/Dark Screen Tab Stops	User Features Locked
Select the type of status line display you want.	Choose one of the following settings:

(continued on next page)

Table 3–7 (Cont.) Setting General Operating Characteristics

To Do the Following...	Choose This Setting...
To display no status line.	No Status Display
To display the status line at all times.	Indicator Status Display
To allow host applications to write messages in place of the status line.	Host Writable Status Display

Customizing Keyboard Operations

To define keyboard operating characteristics, such as keypad keys, cursor keys, margin bell, answerback, keyclick, and keyboard map, use the Keyboard Set-Up screen (see Figure 3–6).

Figure 3–6 Sample Keyboard Set-Up Screen

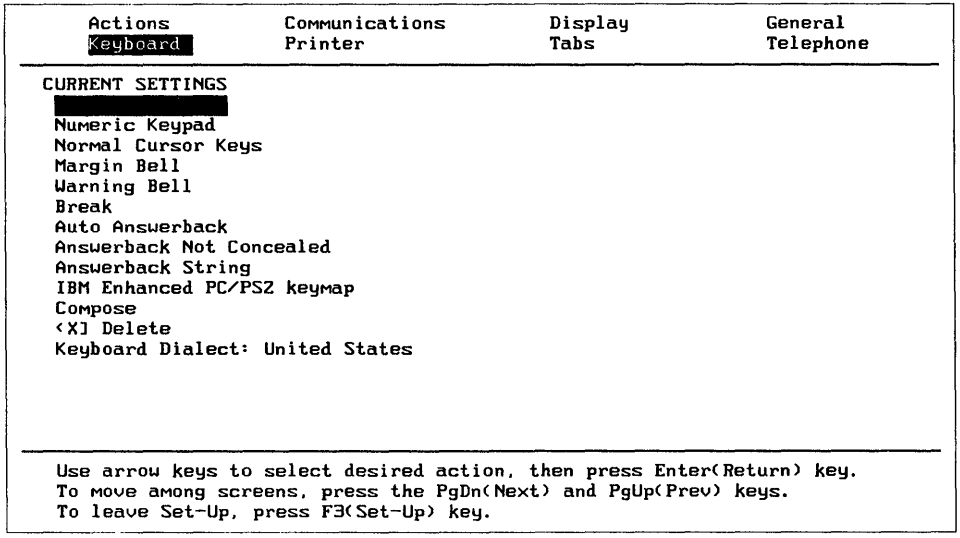


Table 3–8 lists the keyboard characteristics that you can change.

Table 3–8 Setting Keyboard Characteristics

To Do the Following...	Choose This Setting...
Send numbers from the numeric keypad.	Numeric Keypad (default)
Send escape sequences from the numeric keypad.	Application Keypad
Send ANSI cursor control sequences for the cursor keys (such as up, down, left, and right).	Normal Cursor Keys (default)
Send application program control functions for the cursor keys. This is disabled if the terminal is in VT52 mode.	Application Cursor Keys
Set the terminal to sound a bell tone when the text cursor approaches the right margin.	Margin Bell (default)
Set the terminal to turn off the bell tone when the cursor approaches the right margin.	No Margin Bell
Set the terminal to sound a bell tone for operator errors, mail messages, or system messages.	Warning Bell (default)
Set the terminal to turn off the bell tone for operator errors, mail messages, or system messages.	No Warning Bell
Enable the Break key function.	Break (default)
Disable the Break key function.	No Break
Set the terminal to send an answerback message to a host computer after a communication line connection is established.	Auto Answerback
Instruct the terminal not to send an answerback message to a host computer after a communications line connection is established.	No Auto Answerback (default)
Display your answerback message entry on the screen.	Answerback Not Concealed (default)
As you enter you message, it appears on your screen.	

(continued on next page)

Table 3–8 (Cont.) Setting Keyboard Characteristics

To Do the Following...	Choose This Setting...
Conceal your answerback message. As you enter your message, it echoes on the screen. To reset this setting to not concealed, enter a new answerback message.	Answerback Concealed
Allow an answerback message entry. The emulator sends an answerback message when it receives an inquiry control character (ENQ) or if you press Ctrl/Break. In the case of ENQ, the message you enter is sent to a host without requiring further operator action. When you see the cursor in the right column, enter any keyboard character to represent your answerback string. The limit is 30 characters.	Answerback String

Determine whether keystrokes automatically repeat when you press and hold down a key. The No Auto Repeat setting sends only one character when you press and hold down a key. The other selections send the character repeatedly until you release the key.
 	
If you are in VT100 mode and you are using EDT under VMS, the Auto Repeat setting does not work properly. To make it work properly, enter these DCL commands:
 	
`$SET TERMINAL/DEVICE=VT200`
`$SET TERMINAL/INQUIRE`
	Choose one of the following settings:
	
	No Auto Repeat (default)
	Slow Auto Repeat
	Normal Auto Repeat
	Fast Auto Repeat

Determine the keyclick volume for LK250 keyboards.	Choose one of the following settings: No Keyclick Soft Keyclick Normal Keyclick Loud Keyclick

(continued on next page)

Table 3–8 (Cont.) Setting Keyboard Characteristics

To Do the Following...	Choose This Setting...
Select a keyboard map. Use the laptop keymap if your keyboard does not have a separate numeric keypad. This layout is designed for the COMPAQ SLT/286 laptop and can require customizing for other laptop computers. Selecting the Digital LK250 keymap when you do not have that keyboard attached, can cause your personal computer to stop working when it sends commands to the keyboard.	Select one of the following keymaps: IBM XT keymap IBM AT keymap IBM Enhanced PC/PS2 keymap Digital LK250 keymap Laptop keymap Generic keymap Olivetti M28 86-key keymap Olivetti M24 Extended keymap Other keymap COMPAQ SLT/286 keymap COMPAQ LTE/286 keymap COMPAQ LTE/286 with keypad keymap
Enable the Compose key.	Compose (default)
Disable the Compose key.	No Compose
Send a Delete character when you press <x].	<x] Delete (default)
Send a Backspace character when you press <x].	<x] Backspace
Choose a keyboard dialect.	Select one of the following: United States Britain France Germany Italy Spain Sweden Finland Norway Denmark Canada Switzerland (German) Switzerland (French) Belgium Portugal Netherlands

Customizing Your Keyboard Mapping Using Keyboard-Mapping Files

If your computer does not have a keyboard that SETHOST supports, you can customize your own keyboard definitions. That is, you can map the keys on your keyboard to the functions of a standard Digital keyboard.

Files in the DECnet directory, which start with the letters KBD and have a file type of .KBD, are keyboard-mapping files that control the way the personal computer keys map to particular terminal emulator functions. There is a corresponding .HLP file for each .KBD file that describes the mapping for each type of keyboard. For example, KBDAT.KBD and KBDAT.HLP are the keyboard mapping and help files for the IBM Personal Computer AT keyboard. Both types of files are ASCII text files that you can print and review.

When you select a keyboard, SETHOST uses the contents of the corresponding mapping file to map keys to functions. To define your own keyboard mapping, replace one of the .KBD files with your own. Select the existing .KBD file that most resembles your keyboard and modify it with any ASCII text editor.

Keyboard files contain many records. Each record contains the mapping information for one key (or it can be a comment). This record consists of these elements:

- The key code is four hexadecimal digits that uniquely identifies the key on your keyboard.
- An equal sign (=) follows the key code.
- An emulator key code follows the equal sign.
- A comment to help you understand the code. The comments have no effect on the mapping.
- The first line in the file is a flag that has the following meanings:

  0 - Do not replace INT 9H (low level keyboard routine). The generic keyboard map (KBDGEN.KBD) uses this flag.

  1 - Replace INT 9H. The laptop keyboard map (KBDLAP.KBD) uses this flag.

  2 - Replace INT 9H and trap the numlock key. All other keyboard maps use this flag.

Mapping Your Keyboard

The following steps tell how to map the keys of your keyboard:

1. Choose a .KBD file to use as the basis for your own mapping file by determining which of the standard keyboards most closely matches yours.
2. Find the key code for each key on your keyboard that you want to map. To do this, run the KBDCODE.EXE program; then press the keys whose key codes you want to see on your screen.
3. Find the code for the function you want the key to perform, either in the .KBD file or in Table 3–9.
4. Create or change the mapping file record for the key.
5. Save the mapping file with the name that corresponds to the keyboard you select in SETHOST.
6. Run SETHOST. It now uses your new keyboard mapping.

Figure 3–7 shows the first few records of a sample keyboard-mapping file.

Figure 3–7 Sample Keyboard-Mapping File

```
2  ; Flag to replace int9 and trap numlock key
3B00=0100 ; PC KF1 = DEC KHOLD
3C00=0101 ; PC KF2 = DEC KPTEXT
3D00=0104 ; PC KF3 = DEC KSETUP
3E00=0157 ; PC KF4 = DEC KSESNEXT
3F00=0108 ; PC KF5 = DEC KBREAK
.
.
.
```

Table 3–9 shows the standard keyboard functions.

Table 3–9 Standard Keyboard Functions

Function Name	Value	Explanation
Local Function Keys		
KHOLD	0x0100	Hold screen
KPTEXT	0x0101	Print screen (text mode)
KPSIXEL	0x0102	Print screen (graphics mode - not available)
KPAUTO	0x0103	Auto printing On or Off
KSETUP	0x0104	Enter/Exit Set-Up screen
KDATA	0x0105	Data mode for modem
KDIAL	0x0106	Dial modem from Telephone Set-Up screen
KTALK	0x0107	Talk mode for modem
KBREAK	0x0108	Break
KDISCON	0x0109	Disconnect communications line
KANSBK	0x010A	Send answerback string
PF1 to PF4 Function Keys		
KPF1	0x010B	PF1
KPF2	0x010C	PF2
KPF3	0x010D	PF3
KPF4	0x010E	PF4
Miscellaneous Function Keys		
KCOMPOSE	0x0111	Compose character
KSCROLL	0x0112	Not available
KCOL132R	0x0113	Rightmost 80 of 132 columns
KCOL132L	0x0114	Leftmost 80 of 132 columns
KRON	0x0115	Receive file on
KROFF	0x0116	Receive file off
KSEND	0x0117	Send file on\off
KCOL132T	0x0118	Move between screens in 132 column mode
KDATATALK	0x0119	Move between modem data and talk modes

(continued on next page)

Table 3–9 (Cont.) Standard Keyboard Functions

Function Name	Value	Explanation	
"Function" Function Keys			
KF1	0x0120	F1	
KF2	0x0121	F2	
KF3	0x0122	F3	
KF4	0x0123	F4	
KF5	0x0124	F5	
KF6	0x0125	F6	
KF7	0x0126	F7	
KF8	0x0127	F8	
KF9	0x0128	F9	
KF10	0x0129	F10	
KF11	0x012A	F11	
KF12	0x012B	F12	
KF13	0x012C	F13	
KF14	0x012D	F14	
KF15	0x012E	F15	
KF16	0x012F	F16	
KF17	0x0130	F17	
KF18	0x0131	F18	
KF19	0x0132	F19	
KF20	0x0133	F20	
Shift/KF1 through Shift/KF20			
KF1S	0x0134	Shift/F1	(Local Function Key)
KF2S	0x0135	Shift/F2	(Local Function Key)
KF3S	0x0136	Shift/F3	(Local Function Key)
KF4S	0x0137	Shift/F4	(Local Function Key)
KF5S	0x0138	Shift/F5	(Local Function Key)
KF6S	0x0139	Shift/F6	(User-Defined Key)
KF7S	0x013A	Shift/F7	(User-Defined Key)

(continued on next page)

Table 3–9 (Cont.) Standard Keyboard Functions

Function Name	Value	Explanation	
KF8S	0x013B	Shift/F8	(User-Defined Key)
KF9S	0x013C	Shift/F9	(User-Defined Key)
KF10S	0x013D	Shift/F10	(User-Defined Key)
KF11S	0x013E	Shift/F11	(User-Defined Key)
KF12S	0x013F	Shift/F12	(User-Defined Key)
KF13S	0x0140	Shift/F13	(User-Defined Key)
KF14S	0x0141	Shift/F14	(User-Defined Key)
KF15S	0x0142	Shift/F15	(User-Defined Key)
KF16S	0x0143	Shift/F16	(User-Defined Key)
KF17S	0x0144	Shift/F17	(User-Defined Key)
KF18S	0x0145	Shift/F18	(User-Defined Key)
KF19S	0x0146	Shift/F19	(User-Defined Key)
KF20S	0x0147	Shift/F20	(User-Defined Key)
Editing Keypad			
KFIND	0x0150	Find	
KINS	0x0151	Insert here	
KREMOV	0x0152	Remove	
KSEL	0x0153	Select	
KPREV	0x0154	Previous screen	
KNEXT	0x0155	Next screen	
KSESPREV	0x0156	Previous session	
KSESNEXT	0x0157	Next session	
Cursor Keypad			
KUP	0x0158	Up ⇧	
KDOWN	0x0159	Down ⇩	
KLEFT	0x015A	Left ⇦	
KRIGHT	0x015B	Right ⇨	
Numeric Keypad			
KNUM0	0x0160	0	

(continued on next page)

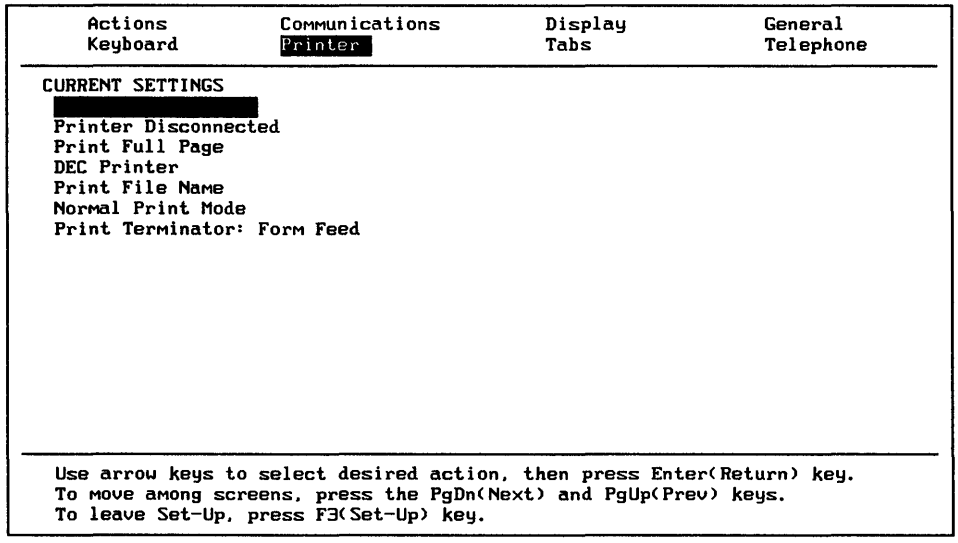
Table 3–9 (Cont.) Standard Keyboard Functions

Function Name	Value	Explanation
KNUM1	0x0161	1
KNUM2	0x0162	2
KNUM3	0x0163	3
KNUM4	0x0164	4
KNUM5	0x0165	5
KNUM6	0x0166	6
KNUM7	0x0167	7
KNUM8	0x0168	8
KNUM9	0x0169	9
KNUMMINUS	0x016A	Minus
KNUMCOMMA	0x016B	Comma
KNUMDOT	0x016C	Dot
KNUMENT	0x016D	Enter
More Miscellaneous Function Keys		
KNUMSTAR	0x016E	Star near or on numeric keypad
KUPS	0x016F	Shift/⇧
KDOWNS	0x0170	Shift/⇩
KMSDOS	0x0171	Push to DOS

Defining Printer Operations

To define printer operations, such as print size, type of printer, and print terminator, use the Printer Set-Up screen (see Figure 3–8).

Figure 3–8 Sample Printer Set-Up Screen



Note

If you are using a serial printer, set up the printer outside of SETHOST. For more information on setting up serial printers, refer to Chapter 3 of the PATHWORKS for DOS, *Client Commands Reference.*

Table 3–10 defines the printer operations that you can set using the Printer Set-Up screen.

Table 3–10 Defining Printer Operations

To Do the Following...	Choose This Setting...
Connect the printer. You must connect the printer to use print screen or to send screen text to a file.	Printer Connected
Disconnect the printer.	Printer Disconnected (default)
Print the entire screen.	Print Full Page (default)
Print only the scrolling region of the screen.	Print Scroll Region
Select a Digital printer as the output device. The SETHOST utility sends escape sequences that direct the printer to accurately represent attributes like underline and bold.	Digital Printer (default)
Select an industry-compatible printer as the output device. The SETHOST utility does not send attributes, such as underline and bold, to the printer.	Other Printer
Send screen text to a print file. The cursor flashes in the SELECTIONS column, requesting you to enter a file name. Enter the default name (PRN), the name of a printer device (for example, LPT1), or any other file name. The print file name must conform to DOS file naming conventions. The SETHOST utility creates the print file in your current directory unless you specify another location as part of the file name. To access this file, exit SETHOST and press the PRINT SCREEN key (F2) to send the text to a file. To use this feature, set the Printer Connected setting.	Print File Name
Select normal print mode for the printer. The Normal Print Mode setting sends information to the printer only when you invoke print functions from the keyboard.	Normal Print Mode (default)
Select auto print mode for the printer. The Auto Print Mode setting prints the text line when the terminal receives a line feed, form feed, or vertical tab code from a host.	Auto Print Mode

(continued on next page)

Table 3–10 (Cont.) Defining Printer Operations

To Do the Following...	Choose This Setting...
Select printer controller mode for the printer. The Printer Controller Mode setting causes the printer port to treat the connected device as a terminal while SETHOST monitors traffic. (The printer and host computer transfer data without displaying the data on the screen.)	Printer Controller Mode
Send a form-feed character at the end of print screen operations.	Print Terminator: Form Feed (default)
Send no terminator at the end of print screen operations.	Print Terminator: None

Defining Tab Settings

To define tab settings, use the Tabs Set-Up screen (see Figure 3–9).

Figure 3–9 Sample Tabs Set-Up Screen

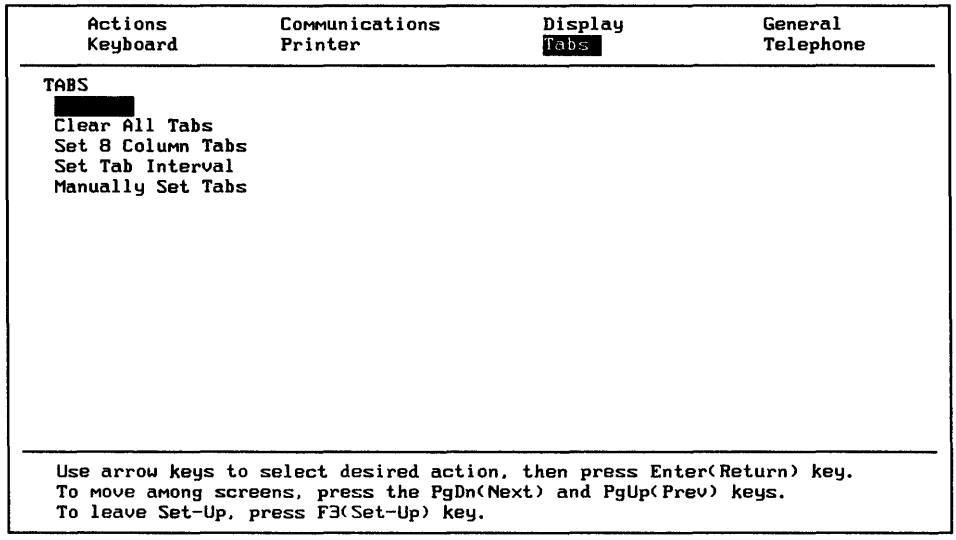


Table 3–11 lists the tab setting characteristics that you can change using the Tabs Set-Up screen.

Table 3–11 Setting Tab Characteristics

To Do the Following...	Choose This Setting...
Clear all tab settings. The SETHOST utility displays tab settings at the bottom of your screen, and indicates each setting with a capital T. When you press Return, SETHOST removes all current settings.	Clear All Tabs
Set tabs every 8 columns starting with column 9. When you press Return, SETHOST sets tabs every 8 columns, and indicates each setting with a capital T.	Set 8 Column Tabs
Set the interval between tabs. The right column displays an interval prompt with the current value. To change this value, enter a new value and press Return.	Set Tab Interval
Change individual tab stops. To set or clear tab stops, press T. When you finish modifying your tab settings, press Return.	Manually Set Tabs

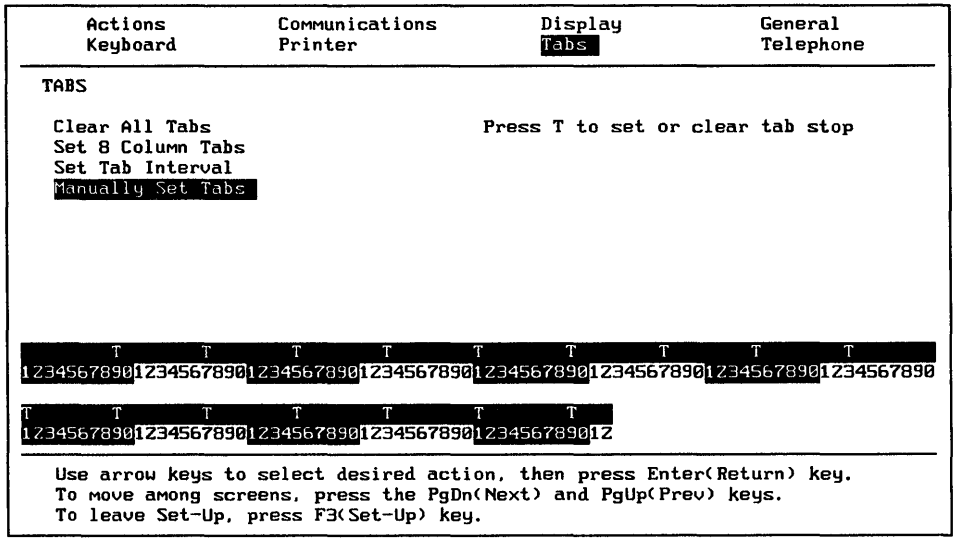
Setting Tabs Manually

To set tabs manually, follow these steps:

1. Use the ⇧ or ⇩ keys to move the video bar to the Manually Set Tabs setting (see Figure 3–10).
2. Press Return.

   A blinking cursor appears in the tabs ruler at the bottom of your Tabs Set-Up screen.

Figure 3–10 Sample Tabs Set-Up Screen Using the Manually Set Tabs Setting



3. Use → or ← to position the cursor on the desired tab stop setting.
4. Press T to set the tab.

   Pressing T again clears the tab stop. (The T key acts as a toggle switch for setting and clearing tabs.)
5. Press Return to end manual tab setting.

Determining Telephone Dialing

To determine how SETHOST accomplishes telephone dialing and to enter telephone numbers to use with a modem, use the Telephone Set-Up screen (see Figure 3–11).

Figure 3–11 Sample Telephone Set-Up Screen

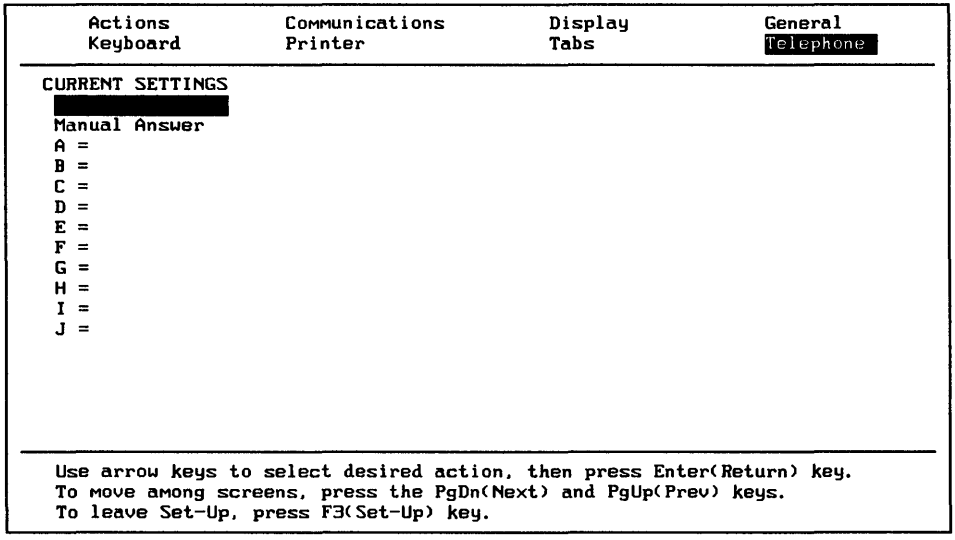


Table 3–12 defines the Telephone Set-Up selections that you can modify.

Table 3–12 Selecting Telephone Settings

To Do the Following...	Choose This Setting...
Select manual answering for the modem. The Manual Answer setting leaves the modem in talk mode after disconnecting from the host, allowing you to answer incoming calls. Normal telephone operation requires the Manual Answer setting.	Modem Answer Selection (default)
Select automatic answering for the modem. The Auto Answer setting leaves the modem in data mode after disconnecting from the host, allowing the terminal to answer incoming calls automatically.	Auto Answer
Specify up to ten telephone numbers to store for automatic dialing. To change or enter a phone number, select a letter. Press Return to move to the CURRENT SETTINGS, and type over the old number or enter a new number. Press Return again. The number appears in the SELECTIONS column, next to its corresponding letter. You can repeat this process for each letter.	A= B= C= D= E= F= G= H= I= J=
To dial a number, press Shift/F4. When SETHOST prompts you for a telephone number, enter the letter assigned to the number. If you select Hayes Modem on the Communications Set-Up screen, SETHOST adds the Hayes dialing commands to the number you selected.	

Using Telephone Selections

Before you use the Telephone Set-Up selections, select and save the following information from the Communications Set-Up screen:

- Comm Port 1: or Comm Port 2: depending on the port to which your modem is connected.

  If you specify the Hayes modem, SETHOST adds the Hayes modem pulse dialing command to the beginning of telephone numbers.

- The appropriate transmit and receive speeds.

For example, to dial in to a terminal server using a Hayes Modem on COM1:, follow these steps:

1. Type the following command line at the DOS prompt:

   ```
   C:\>SETHOST COM1: Return
   ```

2. Press F3 to enter the Set-Up screens.
3. Use the Pg Up and Pg Dn key to select the Communications Set-Up screen.
   - Select Comm Port 1: Data Leads Only.
   - Select a baud rate (for example 2400 baud).
   - Select the stop bit.
   - Select the parity bit.
4. Use the Pg Up and Pg Dn key to select the Telephone Set-Up screen.
5. Enter a dialing string for setting A. For example, enter ATDT86802 for setting A.
6. Press F3 to exit from the Set-Up screens.
7. Press Shift/F4 to dial.

   The screen displays the dialing string while connecting to the terminal server. After SETHOST makes the connection to the terminal server, the screen displays the baud rate for the connection.

For more information on telephone settings, refer to the user's guide for your modem.

Tailoring Your SETHOST Environment

To tailor SETHOST operations to your environment, define environment variables. Use the SET command to define these variables. You can specify these variables as part of a batch file, or you can enter them from the command line. Chapter 4 describes the SET command in detail.

Table 3–13 shows what you can do with the SETHOST environment variables.

Table 3–13 Environment Variables

To Do the Following...	Set This Variable...
Set the buffer size that the CTERM protocol uses. This has the same effect as the VMS Version 5.0 SET HOST/BUFFER_SIZE=*n* command. Adjusting the buffer size allows you to tune the CTERM protocol for better performance with a particular application. To set this buffer size, set the following environment variable in your AUTOEXEC.BAT file or its equivalent. Reload the CTERM protocol to activate this environment variable. For example: C:\>SET CTERMBUF=*n* C:\>CTERM Where: *n* is a value from 128 to 1010. You set the buffer size before you install the CTERM protocol. Set values less than 128 to 128 and values greater than 1010 to 1010. The default size is 512.	SET CTERMBUF=*n*
Tell SETHOST to drop DTR modem signal when it exits (terminating the modem connection).	SET NVTDTR=1
Exit SETHOST when no active network sessions are left.	SET NVTEXIT=1
Control the screen display by disabling all nonscript messages.	SET NVTMSG=1
Discontinue the user messages, such as Press F3 for Set-Up..., that SETHOST displays on the screen.	SET NVTMSG=2
Disable the reset of the video hardware by INT 10H. (This is useful if you want to use a non-standard video mode.)	SET NVTVID=1

(continued on next page)

Table 3–13 (Cont.) Environment Variables

To Do the Following...	Set This Variable...
Control all the video attributes that SETHOST uses for Monochrome mode. For more information, see SET NVTCOLOR. The first 8 characters are the foreground attributes (0-F), and the second 8 are the background attributes (0-7). **Mono Foreground Attributes** 0 Black 1 Underline 2-7 White 8 High Intensity 9 High intensity, Underline A-E Vendor specific F Black **Mono Background Attributes** 0 Black 7 White All others are vendor specific.	SET NVTMONO=*bit-pattern*
Control all the video attributes that SETHOST uses except for blink. The first 8 characters are the foreground attributes (0-F), and the second 8 are the background attributes (0-7). The attributes are in the following order: normal reverse bold reverse bold underline underline reverse underline bold underline bold reverse	SET NVTCOLOR=*bit-pattern* **Color Attributes** 0 Black 1 Blue 2 Green 3 Cyan 4 Red 5 Magenta 6 Brown 7 White 8 Dark Gray 9 Light Blue A Light Green B Light Cyan C Light Red D Light Magenta E Yellow F Intense White

(continued on next page)

Table 3–13 (Cont.) Environment Variables

To Do the Following...	Set This Variable...
Cause SETHOST not to check for the presence of an LK250 keyboard. Use this variable to prevent system suspension when users inadvertently select the LK250 keymap when they do not have an LK250 keyboard.	SET NVTLK250=0
Specify that SETHOST uses the LK250 keyboard map as its default.	SET NVTLK250=1
Prevent a SETTERM/INQUIRE from replacing your National Replacement Character Set with the DEC Multinational Character Set.	SET NVTNRC=1
Specify INT10 or a batch file name to use to switch between 80- and 132-column modes.	SET NVT80=*INT 10H videomode in hex*
Use NVT132 to switch to 132-column mode, and use NVT80 to switch back to 80 columns. Enter these values in your AUTOEXEC.BAT file, or enter them as command line parameters. Subsequently, you can use the columns setting of the SETHOST Display Set-Up screen to switch between 80- and 132-column modes.	or SET NVT132=*INT 10H videomode in hex* or SET NVT80=*batch file name* or SET NVT132=*batch file name*
Specify where to write a log file for the SETHOST session. If you do not specify a location, SETHOST does not write a log file.	SET NVTSCRIPTLOG=*dev:directory*\
Prevent SETHOST from spawning the network services (LAT, CTERM, or TELNET).	SET NVTWIN=1
If you are using a Microsoft Windows enhanced environment, this variable prevents SETHOST from disrupting windows.	

Table 3–14 lists values that function properly with selected video adapters.

Note

Most video hardware requires you to turn on the monitor before you turn on the personal computer if you want to display data in 132-column mode.

Table 3–14 Video Adapter Values

To Use This Video Adapter...	Set This Variable in Your AUTOEXEC.BAT File...
Paradise VGAplus, Tandy VGA, DECstation VGA, DECpc VGA	SET NVT80=3 SET NVT132=55
DECstation models 316 and 320, DECpc VGA	For monochrome systems: SET NVT80=3 SET NVT132=19 For color systems: SET NVT80=3 SET NVT132=23
Paradise Autoswitch EGA 480	SET NVT80=3 SET NVT132=WIDE.BAT
COMPAQ Avanced VGA	Then create a file named WIDE.BAT that contains the line: 13225
ATI VGA Wonder, ATI VIP, ATI EGA Wonder 800, Orchid Designer VGA, Sigma/H VGA, STB VGA Extra/EM	SET NVT80=3 SET NVT132=23
Genoa 5100, Genoa 5200	SET NVT80=3 SET NVT132=60

Note: The following European DECstations do not support a 132-column mode: DECstation 200, DECstation 220, DECstation 300, DECstation 325, DECstation 350, and DECstation 425.

(continued on next page)

Table 3–14 (Cont.) Video Adapter Values

To Use This Video Adapter...	Set This Variable in Your AUTOEXEC.BAT File...

Genoa SuperEGA Hires +	`SET NVT80=3`
	`SET NVT132=WIDE.BAT`
	Then create a file named WIDE.BAT that contains the line:
	SPEGA E25

Video 7 Vega EGA	`SET NVT80=3`
	`SET NVT132=WIDE.BAT`
	Then create a file named WIDE.BAT that contains the line:
	VEGA 41

Video 7 VRAM VGA (with color VGA monitor)	`SET NVT80=NORMAL.BAT` `SET NVT132=WIDE.BAT` Create a file named NORMAL.BAT that contains the line: ESU /3 Create file named WIDE.BAT that contains the line: ESU /41

DEGA	`SET NVT80=3`
	`SET NVT132=WIDE.BAT`
	Then create a file named WIDE.BAT that contains the line:
	DEGA E25

4

Performing Functions Using SETHOST Scripts

A **script** is a text file with commands that allows SETHOST to perform many operations automatically. For example, a script can execute an interactive session with a host computer. This chapter describes SETHOST scripts and the script language.

Note

The information and procedures in this chapter also apply to writing scripts for VT320 emulation.

What Is a Script File?

It is common to perform the same set of operations each time you use a remote host, such as connecting, logging in, reading and sending mail, and perhaps running an application.

To do this, you learn a procedure, then type the same commands and look for the same results each time you use the system. You learn that there is little variation between one session and the next.

A script does the same thing. You design it to type the required commands and look for the variations in output. A SETHOST script is a sequence of commands in a language designed specifically for describing interactive terminal sessions.

Using Script Files

You can create scripts that contain commands used to perform a function, such as automatically logging on to a host system. The scripts execute each command, simulating an interactive session on a host terminal. This is useful if you want to log into a host system automatically, or if you want to connect to a remote system to retrieve information. Using scripts is similar to using Set-Up files; but the script can include commands, where a Set-Up file includes only parameters used to configure your terminal emulator session.

The command line syntax for using script files is as follows:

SETHOST /SCRIPT=*login.scr*

The *login* variable represents the name of the script file, and the *scr* variable represents the file type. You do not have to specify the *scr* variable because it is the system default file type. The following command starts a session using a script file:

SETHOST LAT: *service-name* /SCRIPT=ALLIN1.SCR

Using a LAT connection and calling in the script file ALLIN1.SCR, this command starts a SETHOST session on a specific node.

Script Session Log Files

To create log files when you start up SETHOST, set the NVTSCRIPTLOG environment variable. Refer to Tailoring Your SETHOST Environment for information about setting this variable. When you set this variable, SETHOST stores session information in the file you specify. This is helpful when tracking down problems in scripts.

The OPEN and CLOSE commands in the script language let you store session information in log files.

Overview of the SETHOST Script Language

A script is a file containing commands. The script file is a text file you can create with any text editor. Each line in the script file can contain one command or a label, or it can be empty.

The SETHOST script language has six kinds of commands:

- Communication commands that define the environment that you set up between your computer and the communication line.

BAUD RATE	DATA BITS
XON/XOFF	NO XON/XOFF
NTS	PARITY
PORT	STOP BITS
RX BAUD	TX BAUD

- Control commands that define the sequence of operations in a script.

CASE	CASE END
END ON ERROR	EXIT EMULATOR
EXIT EMULATOR ON	EXIT SCRIPT
EXIT SCRIPT ON	GOTO
NO SKIP ON	ON ERROR
READ	SKIP
SKIP ON	

- Data commands that manipulate and monitor data going to and coming from the communication line.

BREAK	CLEAR LINE
DIAL	DTR CLEAR
DTR SET	HANG UP
PURGE TYPE	SEND
SEND PASSWORD	SEND USERNAME
TYPE UNTIL	WAIT FOR

- Display/Print commands that control your screen and printer.

DEBUG	DISPLAY
ECHO	FPRINT
NO DEBUG	NO ECHO
PRINT SCREEN	PRINTER OFF
PRINTER ON	

- File commands that perform operations on files.

CHAIN	CLOSE
LOAD	OPEN
SCRIPT	SEND FROM

- Other commands that perform miscellaneous functions.

COMMENT	RING BELL
KEYBOARD OFF	KEYBOARD ON
<LABEL>	PAUSE
PAUSE UNTIL	RETRY
SET	SYSTEM
TIMEOUT	TIMER
TIMER OFF	

Setting Up Command Lines

Each line in the script file can contain one command or a label, or it can be blank. Lines in a script file can be as long as 132 characters. Blank lines have no effect on script execution, but they make scripts more readable. Label lines have no effect on script execution unless the script processor is executing a GOTO command. All other lines are command lines.

A command line consists of a command name, such as ON ERROR or NO ECHO, followed by a required colon (:). Some commands also have optional or required operands following the colon. Two commands, CASE and ON ERROR, extend over several lines. (Refer to the command pages for specific examples.)

Command names and other keywords can contain spaces to make them more readable; they are shown in this chapter with spaces. However, SETHOST does not require the spaces. For example, ON ERROR is equivalent to ONERROR. Case also has no effect. For example, On Error and onerror are the same as ON ERROR.

Including Text Strings

Some script commands operate on text strings, and the value of a string can be part of the command. Look at this format example of a command that is described later in this chapter:

DISPLAY: *string*

When *string* is shown in a format example, it represents a string of characters that begins immediately after the colon. Unless the description states otherwise, you can represent special characters, such as control characters, in the command.

In the following example, the string is eight characters long. The first character is a space and the last character is a carriage return character (represented by four characters, two of which are angle brackets).

```
DISPLAY: Hello!<CR>
```

Including Special Characters

When descriptions for *string* do not indicate otherwise, you can include special characters in two ways:

- You can use angle brackets to enclose the decimal value of the character. For example, <3> specifies a Control-C character, and <13> specifies a carriage return character.
- You can use the symbols for special characters in Table 4–1.

Table 4–1 shows the nonprintable characters, control characters, function keys, and keypad keys you can use in script commands.

Table 4–1 Special Characters Used in Script Commands

Category	Character/Key	Script Representation
Control Characters	Control-A	<Ctrl/A>
	Control-B	<Ctrl/B>
	Control-C	<Ctrl/C>
	.	.
	.	.
	.	
	Control-Z	<Ctrl/Z>
Function Keys	F6	<F6>
	F7	<F7>
	.	.
	.	.
	.	
	F15	<F15> or <Help>
	F16	<F16> or <Do>
	.	.
	.	.
	.	
	F20	<F20>

(continued on next page)

Table 4–1 (Cont.) Special Characters Used in Script Commands

Category	Character/Key	Script Representation
Keypad Keys	PF1	<PF1>
	PF2	<PF2>
	PF3	<PF3>
	PF4	<PF4>
	Keypad 0	<KP0>
	Keypad 1	<KP1>
	.	.
	.	.
	.	.
	Keypad 9	<KP9>
	-	<KPMINUS>
	,	<KPCOMMA>
	Enter	<KPENTER>
	Find	<FIND>
	Insert Here	<INSERTHERE>
	Next	<NEXT>
	Prev	<PREV>
	Remove	<REMOVE>
	Select	<SELECT>
	Up Arrow	<UP>
	Down Arrow	<DOWN>
	Right Arrow	<RIGHT>
	Left Arrow	<LEFT>
Nonprintable Characters	Carriage Return	<RET>
	Delete	<DEL>
	Escape	<ESC>
	Form Feed	<FF>
	Line Feed	<LF>
	Tab	<TAB>

Setting Variables

Variables are named data items whose values you can set and test in a script session. The maximum number of variables in a script session is 80, including the two standard variables, Error_Number and Error_Message. However, you can use the SET command to delete variables so you can set new ones.

All variables are character strings, whether their values are numeric or alphanumeric. For example, the variable Error_Number always contains a numeric value, but the value 0 is a 1-character string, and it is not equivalent to the value 00 in a test for equality.

The SET and READ commands create, change the value of, and delete variables. The CASE command tests the value of variables.

The READ command can store up to 126 characters. However, the format of a command line determines the maximum practical size of a variable. See the CASE, READ, and SET commands.

Understanding Error Processing

All script commands return a status after executing. The variables Error_ Number and Error_Message contain the results. All values other than 0 are errors.

When an error occurs, the script processor takes immediate action, depending on the type of error and the commands you code in the script.

Use Table 4–2 to determine the script status.

Table 4–2 Script Status Messages

Message Number	Status Message
0	Command successfully completed
1	Illegal command
2	Timeout
3	Non-numeric parameter
4	Invalid parameter
5	Incompatible settings
6	Unimplemented command
7	Service not available
8	Cannot find file
9	END ON ERROR not found
10	ON ERROR not found
11	CASE END not found
12	DOS command failed
13	Print error
14	Label not found
15	Logfile already open
16	No logfile open
17	File transfer error

(continued on next page)

Table 4–2 (Cont.) Script Status Messages

Message Number	Status Message
18	Variable not found
19	Comm Port not available
20	Nested ON ERROR routines not allowed
21	Scripts nested too deeply
22	Error creating log file
23	Too many variables:symbol table overflow
24	ACL information not defined in database

Timeout Handling

Several script commands make use of a built-in script processor timer. If those commands do not complete within the number of seconds specified by the timer, a timeout results.

Timeout error conditions require special processing, because the error concerns timing rather than an obvious problem. The SETHOST script language allows you more control for timeout error conditions.

First, you can code script commands that help avoid timeout errors. The TIMER and TIMER OFF commands allow you to tailor the timeout period for a particular operation. For example, the TIMER OFF command turns off the timer completely so that a timeout error cannot occur.

The second method of controlling timeout error processing is the RETRY command. It causes a rerun of part of the script if a timeout occurs, and it retries the segment of the script the number of times specified in the command.

If a timeout error still occurs after the script segment is rerun as the result of a RETRY command, the script processor treats it as any other error.

Error Handling

The SETHOST script language allows you to set up error handlers. An **error handler** is a segment of code that executes as a result of the script processor detecting an error condition. Error handlers begin with the ON ERROR command and end with the END ON ERROR command. When execution of a script reaches an ON ERROR command, the script processor marks the location, then skips over the segment. If an error condition occurs, the script processor transfers control to the most recently marked error handler. Therefore, a script can handle errors exactly the way a situation requires. Example 4–1 displays sample script segments:

Example 4–1 Sample Script Segment

```
    ...
COMMENT:    Set up an error handler for the log-in
COMMENT:    that will call another script to log in
COMMENT:    elsewhere if there is a real timeout.
    ON ERROR:
    CASE: error number
          "2" SCRIPT: tryalt
    DEFAULT: GOTO $report error
    CASE END:
    END ON ERROR:
COMMENT:    If it times out, we will try up to
COMMENT:    5 times.
    RETRY:5
    SCRIPT: login1
COMMENT:    The called script, LOGIN1.SCR, returns an
COMMENT:    error in error number, so a timeout will
COMMENT:    run the ON ERROR segment above.
COMMENT:    For the rest of the remote session, set
COMMENT:    up an empty error handler.  If there is
COMMENT:    a timeout, the script processor will
COMMENT:    execute an implicit EXIT SCRIPT.
    ON ERROR:
    END ON ERROR:
    RETRY:0
    SEND: run time report<CR>
    SEND: logout<CR>
    EXIT SCRIPT:
```

After an error handler (ON ERROR) executes, SETHOST returns control to the statement following the one that invoked the handler. This allows SETHOST to continue executing the script if the ON ERROR segment can get past the error. The error handler can transfer control elsewhere or it can end the script and the SETHOST session, if necessary.

When a script starts executing, either because it is the initial script or because it received control in a SCRIPT or CHAIN command, there is no current error handler until execution reaches an ON ERROR command. If an error occurs, and there is no current error handler, the script processor executes an implicit EXIT SCRIPT command, which returns control to either the calling script (if there is one) or to SETHOST.

The following sections detail the error processing for each of the commands:

Sample Scripts

The following sections describe sample scripts that you can use with SETHOST. Each sample provides steps for creating a script file and includes comments about how the commands operate within the script. You can find additional sample scripts that illustrate how to make an asynchronous DECnet connection in the PATHWORKS for DOS, *Installation and Configuration (with Diskettes)*.

A Sample COM Log-In Script

The following sample creates a script file (COMLOGIN.SCR) that logs you into a VMS host using the COM1 port. This script assumes that COM1 is hardwired to a particular computer. (To use COM2, change the PORT: command to read DATA-2 instead of DATA-1).

1. Select the communications method.

   ```
   PORT:Data-1
   ```

2. Use the ECHO command so that you can see what is happening as the script executes each command.

   ```
   ECHO:
   ```

3. Now, start the log-in process. Use the WAIT FOR command. However, if the node you want to connect to is very busy, the WAIT FOR command may time out. To avoid this, first use the RETRY command to make several attempts at a connection (for this example, repeat the attempt four times).

   ```
   RETRY:4
   ```

   Note

   You can also use the TIMER command, supplying a time interval such as 60 seconds. For example: TIMER:60

4. Next, send a carriage return character to get the username prompt, and include the WAIT FOR command here.

   ```
   SEND:<CR>
   WAIT FOR:Username:
   ```

5. After the username prompt appears, you can cancel any remaining RETRY command attempts.

   ```
   RETRY:0
   ```

6. Now you can send the user name.

```
SEND:ROBERTS<CR>
```

7. Wait for the password prompt. If the node you want to connect to is very busy, the WAIT FOR command may time out. To avoid this, first use the RETRY command to make several attempts at completing the log in (for this example, repeat the attempt four times).

```
RETRY:4
WAIT FOR:Password:
```

8. The password prompt appears, so you can cancel the RETRY command attempts.

```
RETRY:0
```

9. You can now enter your password at the prompt. Since the TYPE UNTIL command does not time out, you do not need to specify a RETRY period.

```
TYPE UNTIL:<CR>
```

10. You should now be logged into the host node unless you entered your password incorrectly. Use the EXIT SCRIPT command to leave the script processor and use the terminal emulator.

```
EXIT SCRIPT:
```

In summary, the script commands in your COMLOGIN.SCR file should look like this:

```
PORT:Data-1
ECHO:

RETRY:4
SEND:<CR>
WAIT FOR:Username:

RETRY:0
SEND:ROBERTS<CR>

RETRY:4
WAIT FOR:Password:

RETRY:0
TYPE UNTIL:<CR>
EXIT SCRIPT:
```

A Sample Local Area Transport Log-In Script

The following sample creates a script file (LATLOGIN.SCR) that logs you into a VMS host using LAT.

1. Select the communications method and the service you want to connect to.

```
PORT:LAT
NTS:(service-name)
```

2. Use the ECHO command so that you can see what is happening as the script executes each command.

```
ECHO:
```

3. Now, start the log-in process. Use the WAIT FOR command. However, if the local area network (LAN) is very busy, the WAIT FOR command may time out. To avoid this, first use the RETRY command to make several attempts at a connection (for this example, repeat the attempt four times).

```
RETRY:4
```

4. Next, send a carriage return character to get the username prompt, and include the WAIT FOR command here.

```
SEND:<CR>
WAIT FOR:Username:
```

5. Once the username prompt appears, you can cancel any remaining RETRY command attempts.

```
RETRY:0
```

6. Now you can send the user name.

```
SEND:ROBERTS<CR>
```

7. Wait for the password prompt. If the node is very busy, the WAIT FOR command may time out. To avoid this, first use the RETRY command to make several attempts at completing the log-in (for this example, repeat the attempt four times).

```
RETRY:4
WAIT FOR:Password:
```

8. The password prompt appears, so you can cancel the RETRY command attempts.

```
RETRY:0
```

9. You can now enter your password at the prompt. Because the TYPE UNTIL command does not time out, you do not need to specify a RETRY period.

```
TYPE UNTIL:<CR>
```

10. You should now be logged into the host unless you entered your password incorrectly. Use the EXIT SCRIPT command to leave the script processor and stay in terminal emulation.

```
EXIT SCRIPT:
```

In summary, the script commands in your LATLOGIN.SCR file should look like this:

```
PORT:LAT
NTS:(service-name)
ECHO:

RETRY:4
SEND:<CR>
WAIT FOR:Username:

RETRY:0
SEND:ROBERTS<CR>

RETRY:4
WAIT FOR:Password:

RETRY:0
TYPE UNTIL:<CR>
EXIT SCRIPT:
```

A Sample Mail Reader Script

The following sample script logs you into a VAX system using a LAT network connection, read any new mail messages, extract that mail to a file, print the mail file, then log out.

```
COMMENT:    MAIL.SCR
COMMENT:    This script logs into a VAX system using a network
COMMENT:    LAT connection. It then reads any new mail
COMMENT:    messages, stores them in a file, prints the file, then logs out
```

```
COMMENT:    This script assumes that the user's VMS prompt has
COMMENT:    not been changed with the SET PROMPT command.  If
COMMENT:    it has, the "WAIT FOR:$ " commands must be changed.
COMMENT:    First, we log in using a separate script file.
    SCRIPT: LATLOGIN.SCR
COMMENT:    Start up MAIL.  This assumes LATLOGIN has left us
COMMENT:    at the "$ " prompt.
    SEND : MAIL <CR>
COMMENT:    The ON ERROR block handles the error condition
COMMENT:    that is raised when there are no new mail messages
COMMENT:    (when the WAIT FOR:new message command times out).
COMMENT:    Since there is no new mail, we exit from mail and log out.

    ON ERROR:
    $NO NEW MAIL:
        COMMENT:Tell the user (if he's watching)
        COMMENT:    that he has no new mail.
        DISPLAY:<CR><LF><7>You have NO new mail messages<CR><LF>
        COMMENT:    exit from mail
        SEND:EXIT<CR>
        COMMENT:    Log out and return to DOS.
        WAIT FOR:$
        SEND:LOGOUT<CR>
        EXIT EMULATOR:
    END ON ERROR:
    WAIT FOR:new message
    WAIT FOR:MAIL>
COMMENT:    If you have a LOT of new mail, you might wish to
COMMENT:    increase the TIMER value here, so that the SELECT
COMMENT:    and EXTRACT commands do not cause a timeout.
    SEND:select newmail<CR>
    WAIT FOR:MAIL>
COMMENT:    An alternative here (the following 2 lines) would
COMMENT:    be to turn PRINTER ON: and then SEND:extract/all
COMMENT:    TT:<CR>. This would print the new mail on the
COMMENT:    local printer, if there is one.
    SEND:extr/all newmail.txt <CR>
    WAIT FOR:MAIL>
COMMENT:    Leave mail.
    SEND: exit <CR>
    WAIT FOR:$
COMMENT:    Print the new mail on a printer
COMMENT:    attached to the VAX.
    SEND: print newmail.txt <CR>
    WAIT FOR:$
COMMENT:    Leave VMS and return to DOS.
    SEND:logout<CR>
    EXIT EMULATOR:
```

SETHOST Script Processor Commands

This section describes the SETHOST script processor commands in alphabetical order. Each command description includes an example of the command.

BAUD RATE

Sets the rate at which characters are both received and transmitted.

Format

BAUD RATE: *speed*

Where:

speed is one of the following:

```
 50    300   2400
 75    600   3600
110   1200   4800
134   1800   9600
150   2000  19200
```

Remarks

The BAUD RATE command is invalid when you use a NETWORK, LAT, CTERM, or TELNET service in a PORT command. The RX BAUD and TX BAUD commands are exactly the same as this command. See the PORT command.

Example

This example sets the baud rate to 2400:

```
PORT: Data-1
PARITY: None
DATA BITS: 8
STOP BITS: 1
BAUD RATE: 2400
DIAL:B
```

Results

```
0 - Command Successfully Completed
4 - Invalid Parameter
5 - Incompatible Settings
```

BREAK

Transmits a break signal.

Format

BREAK: *break-length*

Where:

break-length specifies the duration of the break signal in tenths of a second,

or

break-length is one of the keywords SHORT or LONG.

SHORT specifies a length of 0.24 seconds.

LONG specifies a length of 3.5 seconds.

Remarks

The BREAK command is invalid when you use a NETWORK, LAT, CTERM, or TELNET service in a PORT command (see the PORT command).

Example

The first two examples are equivalent.

```
Break:35
BREAK: LONG
BREAK:Short
```

Results

```
0 - Command Successfully Completed
4 - Invalid Parameter
5 - Incompatible Settings
```

CASE

Selects an action depending on the contents of a variable.

Format

CASE: *variable*

"string" command
"string" command

.
.

"string" command

DEFAULT: *command*
CASE END:

Remarks

The CASE command extends over several lines. The first line contains the CASE command and the name of the controlling variable. The SETHOST script facility requires the CASE END command to mark the end of the CASE command. If there is a DEFAULT command, it must appear immediately before the CASE END command.

You must assign a value to the *variable* using the SET or READ commands before you execute the CASE command using that variable. Or, the *variable* can be one of the standard variables named Error_Number or Error_Message. CASE compares the value of each *"string"* with the value of *variable*. The comparison is exact; that is, uppercase letters are not equal to the corresponding lowercase letters.

The script processor executes the *command* paired with the first matching *"string"* (if any). The *command* can be any script command other than ON ERROR and CASE.

If there is a DEFAULT command and there is no match between any *"string"* and the *variable*, the SETHOST script processor executes the *command* associated with the DEFAULT command. If there is no DEFAULT command and there is no match between any *"string"* and the *variable*, the SETHOST script processor transfers control to the command following CASE END.

If the CASE command executes a *command* that returns control, then the SETHOST script facility transfers control to the command following CASE END.

Example

1. This example waits for the user to enter a day of the week. The script processor stores the input in the *variable* day-of-week.

```
READ: day-of-week
CASE: day-of-week
    "MO" GOTO: $readmail
    "mo" GOTO: $readmail
    "WE" EXIT SCRIPT:
    "we" EXIT SCRIPT:
DEFAULT: SCRIPT:sendfile
CASE END:
```

   If you type "MO" or "mo", the script processor transfers control to the label $readmail in the same script. If the user types "WE" or "we", the current script ends. If the user types anything else, SETHOST calls the script SENDFILE.SCR. Because string comparison is exact, the script should tell the user to type only the first two characters of the day. That is, the script should anticipate the user's input. In this example of day of the week, a numeric value of 1 to 7 would be less likely to result in a mismatch.

2. This script expects that the variable file-status was assigned a value by the script that called this one. If that was not the case, then the CASE command fails with error 11.

```
CASE: file-status
    "missing" GOTO: $create_file
    "ready" SEND FROM:output.txt
DEFAULT: EXIT SCRIPT:
CASE END:
```

   If the value in file-status is "missing," the script transfers control to the label $create_file in this script. If the value is "ready," the script sends the characters from the file OUTPUT.TXT out on the communication line. Otherwise, control returns to the script that called this one, or to SETHOST if this is not a called script.

Results

```
 0 - Command Successfully Completed
11 - CASE END not found
18 - Variable not found
```

CASE END

Terminates the multiple-line CASE command.

Format

CASE END:

Example

This example shows a CASE END command terminating a CASE command:

```
CASE: file-status
    "missing" GOTO: $create file
    "ready" SEND FROM:output.txt
DEFAULT: EXIT SCRIPT:
CASE END:
```

Results

```
 0 - Command Successfully Completed
```

CHAIN

Opens a new script file and transfers control to the first command in the file.

Format

CHAIN: *file-name*

Remarks

Control transfers unconditionally to the first command in the script file named *file-name*.

Control does not return to the script that executes the CHAIN command even if the chained script ends with an EXIT SCRIPT command.

See the description of the SCRIPT and EXIT SCRIPT commands.

The default extension of the script file is SCR.

Example

1. Both of the following CHAIN commands transfer control to CONNECT1.SCR:

```
CHAIN: CONNECT1.SCR
CHAIN: connect1
```

2. The following example shows three scripts. The first script calls the second (SCRIPT2) with a SCRIPT command. SCRIPT2 transfers control to SCRIPT3 with a CHAIN command. The last command in SCRIPT3 is EXIT SCRIPT. It causes control to transfer to the line following the outstanding SCRIPT command.

Script 1

```
COMMENT:  Start of SCRIPT1
...
SCRIPT: script2
COMMENT:  Control comes here from SCRIPT3
...
COMMENT:  End of SCRIPT1
```

Script 2

```
COMMENT:  Start of SCRIPT2
...
COMMENT:  We prepared some variables. Now, we give up
COMMENT:    control to SCRIPT3.
CHAIN: script3
COMMENT:  Control cannot get here if the CHAIN command succeeded.
COMMENT:  End of SCRIPT2
```

Script 3

```
COMMENT:  Start of SCRIPT3
...
COMMENT:  We did the important work, so now we return.  We do not
COMMENT:  know or care where control is transferred to.
EXIT SCRIPT:
COMMENT:  End of SCRIPT3
```

Results

```
0 - Command Successfully Completed
8 - Cannot find file
```

CLEAR LINE

Clears all data from the communication line buffer.

Format

CLEAR LINE:

Remarks

The CLEAR LINE command discards all characters received but not displayed on the screen or interpreted by the script processor.

Example

In this example, the script allows the user to type until the string "//BYE" is typed. Then, the script discards all undisplayed characters received from the communication line, sends a LOGOUT command, and returns control to the script that called it.

```
TYPE UNTIL://BYE
CLEAR LINE:
NO ECHO:
SEND:LOGOUT<CR>
EXIT SCRIPT:
```

Results

```
0 - Command Successfully Completed
```

CLOSE

Ends the capture of data by the log file opened with the OPEN command.

Format

CLOSE:

Remarks

The command closes the file and makes it available.

See the OPEN command.

Example

In this example, the script opens a log file, which then captures the output to the screen until the user enters Ctrl/C. The log file then closes and the script continues.

```
OPEN:log1.log
TYPE UNTIL:<3>
CLOSE:
```

Results

```
 0 - Command Successfully Completed
16 - No log file open
```

COMMENT

Ignores the remainder of the command line.

Format

COMMENT: *comment-text*

Remarks

Use the COMMENT command to make your scripts more understandable to others.

Example

The following script shows the use of the COMMENT command:

```
...
ON ERROR:
COMMENT:      If it is not one of these errors,
COMMENT:      I do not know what is wrong, so quit.
CASE: error number
  "15" GOTO: $still open
  "4"  GOTO: $strange name
  DEFAULT: EXIT EMULATOR:
CASE END:
COMMENT:      We cannot get here.
END ON ERROR:
```

Results

```
0 - Command Successfully Completed
```

DATA BITS

Specifies the number of data bits per character in communication line text.

Format

DATA BITS: 7

or

DATA BITS: 8

Remarks

The DATA BITS command is invalid when you use a NETWORK, LAT, CTERM, or TELNET service in a PORT command.

See the PORT command.

Example

The following script segment specifies 8 as the number of data bits per character:

```
PORT: Data-1
PARITY: None
DATA BITS: 8
STOP BITS: 1
BAUD RATE: 2400
DIAL:B
```

Results

```
0 - Command Successfully Completed
4 - Invalid Parameter
5 - Incompatible Settings
```

DEBUG

Displays each line of the script before it is executed.

Format

DEBUG:

Remarks

Use the NO DEBUG command to stop the display.

Use the DEBUG command only in scripts you are developing. It allows you to trace the execution of the script so that you can see the details of its execution. Normally, you would remove DEBUG commands when you are satisfied that the script is performing correctly.

Example

In this example, if the NO ECHO command is active, you would not see any of the text received from the communication line. But the DEBUG command shows you commands as they execute, so you can see whether both SEND commands execute or just one.

```
DEBUG:
NO SKIP ON:hello
SEND:hello back<CR>
SEND:logout<CR>
NO DEBUG:
```

Results

```
0 - Command Successfully Completed
```

DIAL

Turns on the DTR signal and dials a phone number.

Format

DIAL: *dial-string*

Remarks

The DIAL command turns on the DTR signal between the computer and the modem, then sends a *dial-string* to the modem.

The DIAL command compresses *dial-string* before sending it by removing spaces, tab characters, and hyphens (-).

The DIAL command is equivalent to a DTR SET command followed by a SEND command that sends a compressed *dial-string* to the modem.

If you need to send embedded spaces, tab characters, or hyphens in the *dial-string*, use the SEND command instead of the DIAL command.

If *dial-string* is a single letter from A to J, the DIAL command sends the corresponding Set-Up phone number string to the modem.

The DIAL command is invalid when you use a NETWORK, LAT, CTERM, or TELNET service in a PORT command.

See the PORT command.

Example

This example shows a common combination of communication parameters.

```
PORT: Data-1
PARITY: None
DATA BITS: 8
STOP BITS: 1
BAUD RATE: 2400
DIAL: 1 617 555-1212


COMMENT:    This script dials a computer, turns DECnet on, and
COMMENT:    establishes a DDCMP connection.
    ON ERROR:
        COMMENT:    This ON ERROR segment is empty because we
        COMMENT:    do not care if the NCP command fails. (It
        COMMENT:    could fail because DECnet is not installed.)
    END ON ERROR:
COMMENT:    SETHOST cannot use the COM1 port if DECnet is using it,
COMMENT:    so tell DECnet to go away.
    SYSTEM:NCP SET LINE STATE OFF
COMMENT:    Set the communication parameters correctly.
    PORT: Data-1
    BAUD RATE: 2400
$dialit:
COMMENT:    Dial the number.
COMMENT:    Assume it's a Scholar in 'verbose' mode.
    DIAL:<Ctrl/B>T1-800-555-1212!
    ON ERROR:
        DISPLAY: <Ctrl/G>Other end did not answer!<CR><LF>
        PAUSE: 0:0:5
        EXIT EMULATOR:
    END ON ERROR:
    WAIT FOR:Attached
    ON ERROR:
    END ON ERROR:
COMMENT:    The called script (LOGIN.SCR) logs you in or
COMMENT:    executes an EXIT EMULATOR command.
COMMENT:    If it returns, we are at the "$ " prompt.
    SCRIPT: login
COMMENT:    The following VMS DCL command will turn the
COMMENT:    terminal line into a DECnet (DDCMP) line.
    SEND:set terminal/switch=DECnet/protocol=DDCMP/automatic<CR>
COMMENT:    2400 seconds is 4 minutes.  VMS will time out
COMMENT:    if it sees no response in 4 minutes, so there
COMMENT:    is no need to wait any longer than that.
    TIMER: 2400
    ON ERROR:
        GOTO: $other end did not start DECnet
    END ON ERROR:
```

```
COMMENT:    VMS will send the escape sequence below to tell
COMMENT:    us that it is trying to start up DECnet.  If we
COMMENT:    do not see it, we go to the above ON ERROR segment.
    WAIT FOR:<ESC>[&p
COMMENT:    If we get here, we must have seen the escape sequence.
COMMENT:    So, we use NCP to turn the line state back on.
COMMENT:    If the NCP command fails, we assume it's because
COMMENT:    DECnet is not installed, so we have a special routine
COMMENT:    that installs DECnet.
    ON ERROR:
        GOTO: $install DECnet and retry setting line state on
    END ON ERROR:
    SYSTEM:NCP set line state on
    EXIT EMULATOR:
$other end did not start DECnet:
    DISPLAY:Other end did not start DECnet in time.<CR><LF>
    DISPLAY:Would you like to hang up and try again?
    READ: answer
    CASE: answer
        "Y" GOTO: $retry
        "y" GOTO: $retry
    DEFAULT: EXIT EMULATOR:
    CASE END:
$retry:
COMMENT:    Hang up the phone and start over.
COMMENT:    Leave DTR off for 4 seconds to make sure
COMMENT:    it really hung up.
    DTR CLEAR:
    PAUSE: 0:0:4
    DTR SET:
    GOTO: $dialit
$install DECnet and retry setting line state on:
COMMENT:    Note the assumption that the DECnet directory
COMMENT:    is on drive C and in directory \DECnet.
    ON ERROR:
        COMMENT:    If anything fails, just give up.
        EXIT EMULATOR:
    END ON ERROR:
    SYSTEM:SCH
    SYSTEM:DNP C:\DECnet
    SYSTEM:NCP set line state on
    EXIT EMULATOR:
```

Results

```
0 - Command Successfully Completed
4 - Invalid Parameter
5 - Incompatible Settings
```

DISPLAY

Writes a string of characters to your screen.

Format

DISPLAY: *string*

Remarks

The DISPLAY command sends the *string* to the screen with no formatting other than that contained in the *string* itself. Spaces and tab characters that follow the colon (:) are part of the *string*. The *string* can contain control characters to perform screen formatting.

The DISPLAY command does not affect the contents of the log file and does not send anything to the communication line.

Example

This example shows the result of executing DISPLAY commands on the format of the screen.

```
COMMENT:    Note: The first line clears the screen
    DISPLAY:<ESC>[2J
    DISPLAY:Abcd
    DISPLAY:Efghi
    DISPLAY: Jklmno
    DISPLAY:<CR><LF>Pqrst
```

The asterisk (*) shows the position of the cursor on the screen after the DISPLAY commands execute:

```
AbcdEfghi Jklmno
Pqrst*
```

Results

```
0 - Command Successfully Completed
```

DTR CLEAR

Turns off the DTR signal between the computer and modem.

Format

DTR CLEAR:

Remarks

The DTR CLEAR command turns off the DTR signal between the computer and modem. When the DTR signal is off, most modems hang up.

The DTR CLEAR command is invalid when you use a NETWORK, LAT, CTERM, or TELNET service in a PORT command (see the PORT command).

Example

This script segment shows the end of a session on a remote host and the beginning of a second one.

```
...
SEND:logout<CR>
DTR CLEAR:
DIAL:J
...
```

Results

```
0 - Command Successfully Completed
5 - Incompatible Settings
```

DTR SET

Turns on the DTR signal between the computer and modem.

Format

DTR SET:

Remarks

The DTR SET command is invalid when you use a NETWORK, LAT, CTERM, or TELNET service in a PORT command. See the PORT command.

Example

1. The following script segment sets up a Hayes-compatible modem to answer a call. It turns off the timer, then waits until the caller connects and sends a HELLO message. It then sets the timer to allow timeouts, and continues.

```
...
DTR SET:
SEND:ATS0=1<CR>
TIMER OFF:
WAIT FOR:HELLO
TIMER: 15
...
```

2. This example uses a Scholar modem in verbose mode.

```
...
DTR SET:
SEND:<Ctrl/B>
WAIT FOR:Ready
SEND:T18005551212!
...
```

Results

```
0 - Command Successfully Completed
5 - Incompatible Settings
```

ECHO

Sends all characters that the script processor receives through the communication line to the screen.

Format

ECHO:

Remarks

Until an ECHO command executes in a script session, the only characters that appear on the screen are those that the DEBUG and DISPLAY commands display in the script. That is, when a script session starts, the default is the NO ECHO command.

The NO ECHO command reverses the action of the ECHO command. That is, it stops the display of characters on the screen except for those that the DEBUG and DISPLAY commands display.

Example

This script segment finishes logging in to a host. The user sees nothing, because the default is the NO ECHO command. After the log in, the script uses the ECHO command to let the user continue interactively.

```
...
WAIT FOR:Password:
SEND:mypassword<CR>
WAIT FOR:$
DISPLAY:Connected and logged in<CR>
ECHO:
...
```

Results

```
0 - Command Successfully Completed
```

END ON ERROR

Marks the end of an ON ERROR segment.

Format

END ON ERROR:

Remarks

The ON ERROR and END ON ERROR commands mark the beginning and end of a segment of script code that the script processor executes when it detects an error.

The ON ERROR and END ON ERROR commands must occur in pairs.

See the ON ERROR command.

Example

This script segment shows a sample error handling section.

```
ON ERROR:
CASE: error number
      "2" SCRIPT: tryalt
DEFAULT: GOTO $report error
CASE END:
END ON ERROR:
```

Results

```
 0 - Command Successfully Completed
10 - ON ERROR not found
```

EXIT EMULATOR

Ends processing of both the current script and SETHOST.

Format

EXIT EMULATOR:

Remarks

The EXIT EMULATOR command disconnects any links established during the session, closes all files opened during the session, and ends the execution of SETHOST.

Example

This script segment gives the user the choice of ending the script and returning to SETHOST or ending both the script and the SETHOST session:

```
...
$get answer:
DISPLAY:<CR><LF>Type C to continue, Q to leave SETHOST:
READ: action
CASE: action
  "C" EXIT SCRIPT:
  "Q" EXIT EMULATOR:
DEFAULT: GOTO:$get answer
CASE END:
```

Results

Not applicable.

EXIT EMULATOR ON

Ends processing of both the current script and SETHOST if the script processor receives a specific string through the communication line.

Format

EXIT EMULATOR ON: *string*

Remarks

The EXIT EMULATOR ON command waits until the script processor receives *string* through the communication line or until the command times out. If the script processor detects *string*, it ends the script and session in the same way as the EXIT EMULATOR command. See the EXIT EMULATOR command.

If the EXIT EMULATOR ON command times out, the timeout error occurs. See the RETRY and ON ERROR commands and Understanding Error Processing for more information about error processing.

Example

In this example, the script tried to connect to a remote node. If SETHOST detects the string "not reachable", the SETHOST session ends. Otherwise, the EXIT EMULATOR ON command times out. The empty ON ERROR segment causes control to pass to the SEND command, and the script continues.

```
...
ON ERROR:
END ON ERROR:
EXIT EMULATOR ON:not reachable
SEND:<CR>
...
```

Results

```
2 - Timeout
```

EXIT SCRIPT

Ends processing for the current script.

Format

EXIT SCRIPT:

Remarks

If the script processor executed a SCRIPT command during the current session and it is still outstanding when the EXIT SCRIPT command executes, control returns to the line following the most recently executed SCRIPT command.

If there are no outstanding SCRIPT commands, the script processor terminates, and the user can continue the terminal session in SETHOST.

See the SCRIPT command.

Example

The example shows how scripts can call one another and pass information in script variables.

```
COMMENT:  ------------------------------------------------
COMMENT:  Start of FIRST.SCR
COMMENT:  ------------------------------------------------
...
COMMENT:  Get the host name
$get host:
    DISPLAY:Enter host name or QUIT:
    READ: host name
    CASE: host name
        "QUIT" EXIT EMULATOR:
        "quit" EXIT EMULATOR:
    CASE END:
$get method:
COMMENT:    Call script depending on method
    DISPLAY:Type M to use modem, D to use DECnet:
    READ: method
    CASE: method
        "M" COMMENT: SCRIPT: MODEM
        "D" COMMENT: SCRIPT: DECNET
    DEFAULT: GOTO: $get method
    CASE END:
```

```
COMMENT:  If host was invalid for method, give user a chance
COMMENT:  to try again. If host was down, quit.
    CASE: result
        "noconnect" GOTO: $get host
        "nohost" EXIT SCRIPT:
    CASE END:
...
COMMENT:  ------------------------------------------------
COMMENT:  Start of MODEM.SCR
COMMENT:  ------------------------------------------------
...
    CASE: host name
        "WUMBLE" SCRIPT: CALLW
        "WUMBL2" SCRIPT: CALLW2
        "OFFICE" SCRIPT: CALLO
    DEFAULT: SET: host name nohost
    CASE END:
    EXIT SCRIPT:
COMMENT:  ------------------------------------------------
COMMENT:  Start of DECNET.SCR
COMMENT:  ------------------------------------------------
...
    CASE: host name
        "SERGE"  SCRIPT: DSERG
        "INFO" SCRIPT: DINFO
        "WUMBLE" SCRIPT: DWUMB
    DEFAULT: SET: host name nohost
    CASE END:
    EXIT SCRIPT:
```

Results

None.

EXIT SCRIPT ON

Ends processing of the current script if the script processor receives a specific string of characters through the communication line.

Format

EXIT SCRIPT ON: *string*

Remarks

The EXIT SCRIPT ON command waits until *string* is received over the communication line or the command times out. If the script processor detects *string*, it ends processing in the same way as the EXIT SCRIPT command. If the EXIT SCRIPT ON command times out, the timeout error condition occurs.

Example

This example displays a script using the EXIT SCRIPT ON command:

```
$restart:
COMMENT: Set up empty error handler so script falls through if report times out.
    ON ERROR:
    END ON ERROR:
COMMENT: Run the program on the host.
    SEND:run report1<CR>
COMMENT:  Return to calling script when report completes.
    EXIT SCRIPT ON:end of report
COMMENT: If we get here, something is wrong because the report is taking
COMMENT: too long. Try it again. Set up an error handler to quit SETHOST
COMMENT: if we cannot recover.
 ON ERROR:
        EXIT EMULATOR:
    END ON ERROR:
COMMENT:  Send a Control-C and wait for the $.  If we get it, start over.
    SEND:<3>
    WAIT FOR:$
    GOTO: $restart
```

Results

```
0 - Command Successfully Completed
2 - Timeout
```

FPRINT

Prints a file on the local printer.

Format

FPRINT: *file-name*

Example

This script segment creates a file using the output of a program on the host, then prints the file on the local printer.

```
PRINTER ON: sched.txt
SEND:run showschedule<CR>
WAIT FOR:$
PRINTER OFF:
FPRINT: sched.txt
```

Results

```
 0 - Command Successfully Completed
 8 - Cannot find file
13 - Print Error
```

GOTO

Transfers control to a labeled line in the current script.

Format

GOTO: *label*

Remarks

The GOTO command transfers control to the first occurrence of a line in the current script that contains *label.*

A *label* must begin with a dollar sign character ($). The script processor removes and ignores embedded spaces and tab characters before matching *label* with labels in the script (see <LABEL>).

Example

This script segment shows the use of GOTO commands to control a complex procedure.

```
...
COMMENT:  Get the host name
$get host:
    DISPLAY:Enter host name or QUIT:
    READ: host name
    CASE: host name
        "QUIT" EXIT EMULATOR:
        "quit" EXIT EMULATOR:
    CASE END:
$get method:
COMMENT:    Call script depending on method
    DISPLAY:Type M to use modem, D to use DECnet:
    READ: method
    CASE: method
        "M" COMMENT: SCRIPT: MODEM
        "D" COMMENT: SCRIPT: DECNET
    DEFAULT: GOTO: $get method
    CASE END:
```

```
COMMENT:  If host was invalid for method, give user a chance
COMMENT:  to try again. If host was down, quit.
    CASE: result
        "noconnect" GOTO: $get host
        "nohost" EXIT SCRIPT:
    CASE END:
...
```

Results

```
 0 - Command Successfully Completed
14 - Label not found
```

HANG UP

Turns off the DTR signal between the computer and modem.

Format

HANG UP:

Remarks

The HANG UP command is the same as the DTR CLEAR command.

The HANG UP command is invalid when you use a NETWORK, LAT, CTERM, or TELNET service in a PORT command.

See the PORT command.

Example

This script segment shows the end of a modem session and the beginning of a new one.

```
...
SEND:logout<CR>
HANG UP:
DIAL:J
...
```

Results

```
0 - Command Successfully Completed
5 - Incompatible Settings
```

KEYBOARD OFF

Causes SETHOST to ignore Ctrl/C and Ctrl/F10 keystrokes from the keyboard during script processing.

Format

KEYBOARD OFF:

Remarks

The KEYBOARD OFF command causes SETHOST to ignore input of Ctrl/C and Ctrl/F10 from the keyboard. This prevents the user from executing an EXIT SCRIPT command from the keyboard.

If the keyboard is off when the script processor executes a READ or TYPE UNTIL command, SETHOST turns the keyboard on temporarily during execution of the command.

If the user enters Ctrl/C or Ctrl/F10 while the KEYBOARD OFF command is in effect, SETHOST can recognize the keystroke later if the script executes a KEYBOARD ON command, because SETHOST stores keystrokes temporarily in a type ahead buffer and could still be there when keyboard input is again accepted.

Example

This script segment sends the contents of a file after disabling the keyboard. After the script sends the file, the user can end the session from the keyboard.

```
...
KEYBOARD OFF:
SEND FROM: data.txt
KEYBOARD ON:
...
```

Results

```
0 - Command Successfully Completed
```

KEYBOARD ON

Causes SETHOST to accept keystrokes again during script processing.

Format

KEYBOARD ON:

Remarks

The KEYBOARD ON command enables keyboard input. This is the default. If the keyboard is off when the script processor executes a READ or TYPE UNTIL command, SETHOST temporarily turns the keyboard on during execution of the command.

Example

This script segment sends the contents of a file after disabling the keyboard. After the script sends the file, the user can end the session from the keyboard.

```
...
KEYBOARD OFF:
SEND FROM: data.txt
KEYBOARD ON:
...
```

Results

```
0 - Command Successfully Completed
```

<LABEL>

Names the line it appears on.

Format

label:

Remarks

The *label* identifies the line it occupies as a possible destination for GOTO commands in the same script.

A *label* must begin with a dollar sign character ($). The script processor removes and ignores embedded spaces and tab characters before matching a label in a GOTO command with labels in the script.

The *label* must be the only thing on the line, and it must be followed by a colon.

The same *label* may appear on more than one line in a script. However, the GOTO command always refers to the first occurrence of a label in the script, so later occurrences of the same label have no effect on execution of the script.

Example

These are all valid labels:

```
$get file:
$Thisisalonglabelandveryhardtoread:
$DO IT AGAIN:
$READ:
```

Results

Does not change Error_Number or Error_Message.

LOAD

Reads a SETHOST Set-Up file and makes the settings in it the current settings.

Format

LOAD: [*file-name*]

Remarks

The *file-name* is optional. If it does not appear, the LOAD command looks for SETHOST.DAT as the default. If it does not find SETHOST.DAT or the specified file, the LOAD command ends with an error.

The effect of the LOAD command is equivalent to recalling Set-Up parameters in SETHOST.

Note

The SETHOST utility closes any active network sessions before it loads the new Set-Up file.

Example

This script segment loads a Set-Up file, then dials the first telephone number stored in it.

```
LOAD: modem1.dat
DIAL:A
WAIT FOR:Attached
...
```

Results

```
0 - Command Successfully Completed
8 - Cannot find file
```

NO DEBUG

Stops displaying script lines before execution.

Format

NO DEBUG:

Remarks

The NO DEBUG command reverses the effect of a DEBUG command.

The NO DEBUG command is the default when SETHOST begins.

Example

In this example, if the NO ECHO command is active, you would not see any of the text received from the communication line. But the DEBUG command shows you commands as they execute, so you can see whether both SEND commands execute or just one.

```
DEBUG:
NO SKIP ON:hello
SEND:hello back<CR>
SEND:logout<CR>
NO DEBUG:
```

Results

```
0 - Command Successfully Completed
```

NO ECHO

Turns off the display of characters received through the communication line.

Format

NO ECHO:

Remarks

The NO ECHO command reverses the action of ECHO. That is, it stops the display of characters on the screen except for those that the DEBUG and DISPLAY commands display.

Example

This script segment finishes logging in to a host. The user sees nothing, because the default is the NO ECHO command. After the log in, the script uses the ECHO command to let the user continue interactively.

```
...
WAIT FOR:Password:
SEND:mypassword<CR>
WAIT FOR:$
DISPLAY:Connected and logged in<CR>
ECHO:
...
```

Results

```
0 - Command Successfully Completed
```

NO SKIP ON

Executes the next line in the script if the script processor receives a specific string of characters through the communication line.

Format

NO SKIP ON: *string*

Remarks

The NO SKIP ON command waits until *string* is received over the or until the command times out. If the script processor detects *string*, it transfers control to the next line. If the NO SKIP ON command times out, the script processor transfers control to the line following the next line. A timeout error cannot occur because timeout is a normal result of the NO SKIP ON command.

See the SKIP ON command for more examples.

Example

This example runs a program on the remote host after making sure that the data file is there. First, it sends a command to find the file. If it receives the string "no files found" from the host, it creates a copy of the file from a back-up, then runs the program. Otherwise, it skips the line with the "copy" command and just runs the program.

```
SEND:directory myfile.dat<CR>
NO SKIP ON:no files found
SEND:copy myfile.bak myfile.dat<CR>
SEND:run dailywork<CR>
```

Results

```
0 - Command Successfully Completed
```

NO XON/XOFF

Causes the script processor not to use the XON/XOFF protocol.

Format

NO XON/XOFF:

Remarks

When a script session starts, the default setting is XON/XOFF: 64. Use the NO XON/OFF command if the script processor ignores XON and XOFF characters.

See the XON/XOFF command.

Example

This script segment causes the script processor to ignore all XON and XOFF characters.

```
NO XON/XOFF:
```

Results

```
0 - Command Successfully Completed
5 - Incompatible Settings
```

NTS

Specifies a Network Terminal Service that the script session connects to.

Format

NTS: *name* [*variable-name*]

Where:

name	Specifies a LAT service name, a DECnet node name, or a TELNET host name.
variable-name	Specifies a variable name that contains the password.

You can send a password for a LAT service connection, or a connection to a non-LAT service, such as a password protected modem pool.

Remarks

If there is an NTS command, there must also be a PORT command or a SETHOST Set-Up parameter that specifies a NETWORK, LAT, CTERM, or TELNET service in a PORT command.

When the NTS command executes, it ends with an error if the PORT command in effect does not specify a NETWORK, LAT, CTERM, or TELNET service (see the PORT command).

The NTS command is optional if the SETHOST Set-Up parameters specify a LAT, CTERM, TELNET, or NET service and a valid service name. If the PORT command specifies a LAT service, the NTS *service* must be a LAT service. If the PORT command specifies a CTERM service, the NTS *service* must be a valid DECnet node name. If the PORT command specifies a TELNET service, the NTS *service* must be a valid host name.

Example

1. The script segment specifies LAT as the network service and LAN_SVC_1 as the LAT service to connect to.

```
PORT: LAT
NTS: LAN_SVC_1
```

2. This script segment specifies CTERM as the network service and hisvax as the DECnet node name to connect to.

```
PORT: cterm
NTS: hisvax
```

Results

0 - Command Successfully Completed
4 - Invalid Parameter
5 - Incompatible Settings
18 - Variable not found

ON ERROR

Marks the beginning of an error processing segment of script code.

Format

ON ERROR:

Remarks

The ON ERROR and END ON ERROR commands mark the beginning and end of a segment of script code that the script processor executes when it detects an error.

The ON ERROR and END ON ERROR commands must occur in pairs.

The ON ERROR command marks the next line as the first line that the script processor executes when an error occurs in the current script.

Execution of another ON ERROR command sets a new location for error handling.

Error handling is defined when the script processor executes an ON ERROR command in the current script.

The script processor ignores the lines between the ON ERROR and END ON ERROR commands unless they execute as the result of an error, and control transfers to the line following the END ON ERROR command.

When the ON ERROR segment executes as the result of an error condition and execution reaches the END ON ERROR command, control returns to the command following that which caused the error.

The ON ERROR segment can contain GOTO commands to avoid return to the command following that which caused the error.

If a command in an ON ERROR segment ends with an error, the script processor executes an implicit EXIT SCRIPT command.

It is an error to transfer control to a line within an ON ERROR segment from outside the segment. The results are undefined.

The ON ERROR segment can contain SCRIPT commands to call other scripts as subroutines. If control returns from the called script, the ON ERROR segment continues.

See Understanding Error Processing for information about error processing.

See the RETRY command for information on timeout error handling.

Example

This script segment shows a sample error handling section.

```
ON ERROR:
CASE: error number
      "2" SCRIPT: tryalt
DEFAULT: GOTO $report error
CASE END:
END ON ERROR:
```

Results

```
 0 - Command Successfully Completed
 9 - END ON ERROR not found
```

OPEN

Prepares a log file to capture all text received from the communication line that is sent to the screen.

Format

OPEN: *file-name*

Remarks

If the script processor does not find the file, the OPEN command creates it.

If the script processor finds the file, the OPEN command opens it at the end of file (EOF) position. That is, the OPEN command appends the new text to the end of the existing file.

After the OPEN command executes, it also stores all the characters that the communication line sends that are sent to the screen in the log file.

A script session can open no more than one log file at a time. However, it can execute a CLOSE command and then another OPEN command to continue logging to a different file.

The default extension for log files is LOG. So, if *file-name* does not have an extension, the OPEN command looks for or creates a file named *file-name*.LOG.

This log file is separate from the session log file that the script processor automatically creates. Do not use a *file-name* that conflicts with the session log file.

Example

This script allows the user to start and stop logging of the remote session by pressing Ctrl/P repeatedly. Each time logging begins, the OPEN command appends the text that the communication line sends to the file REMOTE.TXT.

```
...
    DISPLAY:<CR><LF> Press Ctrl/P to toggle session logging
$loop:
    TYPE UNTIL:<Ctrl/P>
    OPEN: remote.txt
    DISPLAY:Session logging ON<CR><LF>
    TYPE UNTIL:<Ctrl/P>
    CLOSE:
    DISPLAY:Session logging OFF<CR><LF>
    GOTO: $loop
```

Results

```
 0 - Command Successfully Completed
 4 - Invalid Parameter
15 - Logfile already open
```

PARITY

Sets the type of parity checking on the communication line.

Format

PARITY: *parity-type*

Where:

parity-type is one of the following:

NONE
EVEN
EVEN NO CHECK
ODD
ODD NO CHECK
MARK
SPACE

Remarks

The PARITY command specifies the type of parity checking that SETHOST performs on the communication line.

The PARITY command is invalid when you use a NETWORK, LAT, CTERM, or TELNET service in a PORT command (see the PORT command).

Example

1. In this example, the first PARITY command starts checking for ODD parity. The second turns off parity checking.

```
PARITY: odd

PARITY: none
```

2. The following example shows how parity checking is set as part of setting other communication parameters:

```
PORT: Data-1
PARITY: None
DATA BITS: 8
STOP BITS: 1
BAUD RATE: 2400
DIAL:B
```

Results

```
0 - Command Successfully Completed
4 - Invalid Parameter
5 - Incompatible Settings
```

PAUSE

Suspends processing for a specified time.

Format

PAUSE: *length-of-pause*

Remarks

The PAUSE command suspends script processing for the specified length of time, after which execution continues with the next line.

The format of *length-of-pause* is *hh:mm:ss*, where *hh*, *mm*, and *ss* represent hours, minutes, and seconds. You can omit numbers on the right, in which case the script processor treats them as zeros, as in the examples that follow.

Example

1. These examples explain the format of the *length-of-pause* value.

```
COMMENT: Suspend processing for 1 hour, 12 minutes, 32 seconds
PAUSE: 1:12:32

COMMENT: Suspend processing for 9 hours, 16 minutes
PAUSE: 9:16

COMMENT: Suspend processing for 13 minutes, 2 seconds
PAUSE: 0:13:2

COMMENT: Suspend processing for 5 hours
PAUSE: 5

COMMENT: Suspend processing for 5 seconds
PAUSE: 0:0:5
```

2. This example shows a script segment that periodically checks for new mail on a remote system and prints it out on the local printer.

```
$check for mail:
COMMENT:  Call LOGIN.SCR to log in to host
     SCRIPT: login
COMMENT:  Send the MAIL command
     SEND:mail<CR>
COMMENT:  If the string "message" appears, go read the mail.
COMMENT:  Otherwise, go to $Nomail to get out of MAIL.
     SKIP ON:message
     GOTO: $Nomail
COMMENT:  I have mail, so turn on the printer, extract all the
COMMENT:  messages to the terminal (comm line),
COMMENT:  and turn the printer off.
     PRINTER ON:
     SEND:extract/all sys$output<CR>
     PRINTER OFF:
$Nomail:
COMMENT:  Finished.  Get out of MAIL and then log out.
     SEND:exit<CR>
     SEND:logout<CR>
COMMENT:  Wait one hour, then check mail again.
     PAUSE: 1:00:00
     GOTO: $check for mail
```

3. This script segment shows a common use for the PAUSE command, timing output so the user can read it.

```
...
DISPLAY:              INSTRUCTIONS<CR><LF>
DISPLAY: <CR><LF>
DISPLAY: After you have logged in, the system<CR><LF>
DISPLAY: will ask for your name, employee number,<CR><LF>
DISPLAY: and department.  If the information you<CR><LF>
DISPLAY: enter is valid, the system presents a<CR><LF>
DISPLAY: menu that gives you choices for the rest<CR><LF>
DISPLAY: of the session. <CR><LF>
PAUSE: 0:0:10
DISPLAY: <CR><LF>
DISPLAY: After you select the session option, you<CR><LF>
DISPLAY: will see a set of instructions like this<CR><LF>
DISPLAY: one.  You can refer to the instructions<CR><LF>
DISPLAY: at any time by typing HELP. <CR><LF>
PAUSE: 0:0:10
...
```

Results

```
0 - Command Successfully Completed
4 - Invalid Parameter
```

PAUSE UNTIL

Suspends processing until a specified date and time arrives.

Format

PAUSE UNTIL: *DD-MMM-YYYY:HH:MM:SS*

Remarks

To delay the execution of a script until a specific date and time, use the PAUSE UNTIL command. For example, you could start a mail collection/printing script that has a PAUSE UNTIL command to periodically check for mail and print it while you are away from your workstation.

The PAUSE UNTIL command uses the local personal computer systems date and time, not the date and time of the remote host system.

You are not required to specify a date when you use this command. If you do not specify a date, the command defaults to the current personal computer system date. However, you must specify a specific time in hours and minutes when you use this command.

Example

This example shows a script segment that waits until a specific date and time to check for new mail on a remote system and then prints it out on the local printer.

```
PAUSE UNTIL: 31-MAR-1989:12:30:00
$check for mail:
COMMENT:  Call LOGIN.SCR to log in to host
     SCRIPT: login
COMMENT:  Send the MAIL command
     SEND:mail<CR>
COMMENT:  If the string "message" appears, go read the mail.
COMMENT:  Otherwise, go to $Nomail to get out of MAIL.
     SKIP ON:message
     GOTO: $Nomail
COMMENT:  I have mail, so turn on the printer, extract all the
COMMENT:  messages to the terminal (comm line),
COMMENT:  and turn the printer off.
     PRINTER ON:
     SEND:extract/all sys$output<CR>
     PRINTER OFF:
$Nomail:
COMMENT:  Finished.  Get out of MAIL and then log out.
     SEND:exit<CR>
     SEND:logout<CR>
COMMENT:  Wait until tomorrow at the same time, then check mail again.
     PAUSE UNTIL: 01-APR-1989:12:30:00
     GOTO: $check for mail
```

Results

```
 0 - Command Completed Successfully
 4 - Invalid Parameter
```

PORT

Tells the script processor how to communicate with the host computer.

Format

PORT: *method*

Where:

method is one of the following:

DATA-1
MODEM-1
DATA-2
MODEM-2
INTEGRAL-2
CTERM
LAT
TELNET
NETWORK

Remarks

The PORT command tells the script processor how to communicate with the host computer. If there is no PORT command, SETHOST uses the Set-Up defaults. Table 4–3 lists the connection methods and their associated settings.

Table 4–3 Connection Methods and Settings

To Use This Method...	Choose This Setting...
Port 1, data leads only	DATA-1
Port 1, full modem control	MODEM-1
Port 2, data leads only	DATA-2
Port 2, full modem control	MODEM-2
Port 2, integral modem	INTEGRAL-2
DECnet CTERM protocol	CTERM
LAT protocol	LAT

(continued on next page)

Table 4–3 (Cont.) Connection Methods and Settings

To Use This Method...	Choose This Setting...
For DECnet, SETHOST tries the LAT protocol first, but if that fails, then tries the CTERM protocol. For TCP/IP, SETHOST tries only the TELNET protocol.	NETWORK
TELNET protocol	TELNET

See the NTS command.

Example

These examples show the use of the PORT command.

1. This script segment requests a network service connection.

```
PORT: NETWORK
```

2. This script segment requests an asynchronous connection using data leads only.

```
PORT: Data-1
```

Results

```
 0 - Command Successfully Completed
 4 - Invalid Parameter
19 - Comm Port not available
```

PRINT SCREEN

Prints the current screen on the local printer.

Format

PRINT SCREEN: [*file-name*]

Where:

The *file-name* is the name that you specified for the Print File Name choice of the Printer Set-Up screen.

Remarks

If there is a *file-name*, the PRINT SCREEN command writes the contents of the screen to the named file instead of the printer.

If there is a *file-name* and the file already exists, this command appends the screen contents to the end of the file. Otherwise, it creates a new file with the name specified in *file-name*.

Example

The first PRINT SCREEN command example sends the contents of the screen to the local printer. The second example uses a file named SCRN1.

```
PRINT SCREEN:

PRINT SCREEN: scrn1
```

Results

```
 0 - Command Successfully Completed
 4 - Invalid Parameter
13 - Print Error
```

PRINTER OFF

Turns off echo printing.

Format

PRINTER OFF:

Remarks

Reverses the action of the PRINTER ON command.

The PRINTER OFF command is the default when the script session starts.

See the PRINTER ON command.

Example

The following example turns off echo printing:

```
PRINTER OFF:
```

Results

```
0 - Command Successfully Completed
```

PRINTER ON

Turns on echo printing.

Format

PRINTER ON: [*file-name*]

Remarks

The PRINTER ON command sends all text received from the communication line to the local printer.

The PRINTER ON command does not affect the appearance of the screen.

If there is a *file-name*, the PRINTER ON command directs the output to the named file instead of the printer.

If there is a *file-name* and the file already exists, the PRINTER ON command appends the output to the end of the file. Otherwise, the PRINTER ON command creates a new file with the name specified in *file-name*.

Example

The first PRINTER ON command example starts sending output to the local printer. The second example uses a file named SCRN1.

```
PRINTER ON:

PRINTER ON: scrn1
```

Results

```
0 - Command Successfully Completed
4 - Invalid Parameter
```

PURGE TYPE

Clears the contents of the type ahead buffer.

Format

PURGE TYPE:

Remarks

The SETHOST utility stores keyboard input temporarily in the type ahead buffer until the script processor is ready to use it. The PURGE TYPE command clears the contents of the type ahead buffer. Use this command when you want the script to ignore previous keystrokes from the user.

Example

This script segment logs the user in to a remote system. The empty ON ERROR segment causes the EXIT SCRIPT ON command to fall through if it does not exit as a result of detecting the $ prompt after logging in. The user may have anticipated a successful log in and typed some commands; so the PURGE TYPE command clears the user's input. The script then tells the user that the log in failed, and it transfers control to another part of the script to try the log in again.

```
...
ON ERROR:
END ON ERROR:
EXIT SCRIPT ON:$
PURGE TYPE:
DISPLAY:Log in timed out.  Retrying log in.
GOTO: $start login
```

Results

```
0 - Command Successfully Completed
```

READ

Accepts input from the keyboard and stores it in a variable.

Format

READ: *variable-name*

Remarks

If the variable *variable-name* already exists, the READ command replaces its current value.

If the variable *variable-name* does not exist, the READ command creates it and assigns a value to it, unless the maximum number of variables would be exceeded by adding the new one. In that case, the command ends with an error.

The READ command accepts input from the keyboard until the user presses Return. The SETHOST utility stores the characters the user types before the Return in the *variable-name*.

The user can type any number of characters before pressing Return. However, the READ command stores no more than the first 74 characters, and the script processor ignores the rest.

Note

To see the input on the screen, turn the ECHO command on before executing the READ command.

Example

In this example, the script displays a menu on the screen and tells the user to type a letter to choose the next action. The READ command stores the user's response in the variable USER_ACTION. The CASE command then calls a script to perform the user's choice. If the user types an invalid response, the DEFAULT command transfers control back to the menu display.

```
$get user input again:
...
COMMENT:    Turn on ECHO, just in case it is off.
    ECHO:
COMMENT:    Now get the user's choice.
    DISPLAY:Type a letter from A to N and press RETURN:
    READ: user action
    CASE: user action
         "A" SCRIPT: readfl
         "a" SCRIPT: readfl
         "B" SCRIPT: copyfl
         "b" SCRIPT: copyfl
         ...
         "N" SCRIPT: nextnd
         "n" SCRIPT: nextnd
    DEFAULT: GOTO: $get user input again
    CASE END:
```

Results

```
 0 - Command Successfully Completed
 4 - Invalid Parameter
18 - Variable not found
```

RETRY

Causes control to transfer to the next line in the script when a timeout error occurs.

Format

RETRY: *counter*

Remarks

Each script has a *retry counter*, which keeps track of the number of times that a segment of code can be rerun when a timeout error occurs. The script processor automatically sets the script's retry counter to 0 when it starts the first script, just as if the first command in the script were RETRY: 0. The RETRY command stores the number *counter*, which must not be negative, in the script's retry counter. A script's retry counter is also set to 0 when the script is called with a SCRIPT or CHAIN command.

Example

1. This example sets the script's retry counter to 3, which means that unless another RETRY command executes, each time (up to the third) that a timeout error occurs, the script processor decrements the retry counter and transfers control to the line following the RETRY command. If a fourth timeout error occurs before another RETRY command executes, the ON ERROR segment executes, if there is one; and if there is not an ON ERROR segment, an automatic EXIT SCRIPT command executes.

   ```
   RETRY: 3
   ```

2. This example sets the script's retry counter to 0, therefore reruns do not occur when timeout errors occur.

   ```
   RETRY: 0
   ```

Results

```
0 - Command Successfully Completed
4 - Invalid Parameter
```

RING BELL

Causes the workstation to beep one time.

Format

RING BELL:

Remarks

Use the RING BELL command to signal events by making the workstation beep. Control the frequency of the beeps by using the PAUSE command along with the RING BELL command. This is a useful command to prompt for user input, signal error conditions, or alert a user when a lengthy script has completed.

Example

This script segment causes the workstation to beep once every 5 seconds for 5 times.

```
SCRIPT: EXAMPLE.SCR
COMMENT: This script takes a long time to complete.  When it does...
COMMENT: Ring the bell once every 5 seconds for five times.
RING BELL:
PAUSE:0:0:5
RING BELL:
PAUSE:0:0:5
RING BELL:
PAUSE:0:0:5
RING BELL:
PAUSE:0:0:5
RING BELL:
PAUSE:0:0:5
```

Results

```
0 - Command Successfully Completed
```

RX BAUD

Sets the rate at which characters are both received and transmitted.

Format

RX BAUD: *speed*

Where:

speed is one of the following:

50	300	2400
75	600	3600
110	1200	4800
134	1800	9600
150	2000	19200

Remarks

The RX BAUD command is exactly the same as the BAUD RATE command.

The RX BAUD command is invalid when you use a NETWORK, LAT, CTERM, or TELNET service in a PORT command. See the PORT command.

Example

This script segment sets the baud rate to 2400.

```
PORT: Data-1
PARITY: None
DATA BITS: 8
STOP BITS: 1
RX BAUD: 2400
DIAL:B
```

Results

```
0 - Command Successfully Completed
4 - Invalid Parameter
5 - Incompatible Settings
```

SCRIPT

Calls another script file as a subroutine.

Format

SCRIPT: *script-file*

Remarks

The SCRIPT command causes the script processor to open the file named by *script-file* and transfer control to the first command in that file. That script is termed the *called script*.

If *script-file* does not have an extension, the script processor looks for a file named *script-file*.SCR.

Execution of the current (or *calling*) script stops. However, the script processor keeps the script file open and keeps track of the location of the SCRIPT command when it executes. The SCRIPT command remains outstanding until the script processor reaches an EXIT SCRIPT command in the called script. Then, the script processor returns control to the line immediately following the SCRIPT command in the original script. The script processor also saves the following information about the current script:

- Retry counter
- Location of current RETRY command
- Location of current ON ERROR segment

Control returns when an explicit or implicit EXIT SCRIPT command executes. The script processor executes an implicit EXIT SCRIPT command if:

- An error occurs in a script and there is neither an active retry counter (see the RETRY command) nor a current ON ERROR segment. (See Understanding Error Processing for full information about error processing.)
- Control reached the end of the current script file.

A called script can call another script by executing a SCRIPT command, causing another SCRIPT command to be outstanding. This is called *nesting*. Scripts can be nested 10 deep. When a script executes an EXIT SCRIPT command, control returns to the line following the most recently executed SCRIPT command, which is then no longer outstanding.

There is no restriction on which scripts a script can call. A script can call itself with a SCRIPT command, or a called script can call the original script with a SCRIPT command.

When control returns from the called script, the variables error_number and error_message contain the result of executing the called script. So, if an error occurred in the called script and it was not handled there, the calling script can handle the error after control returns.

See the CHAIN command, which transfers control to another script but does not set up a return location. Only SCRIPT commands set up return locations and become outstanding. CHAIN command executions do not increase the nesting depth.

Example

The following example shows three scripts. The first script calls the second (SCRIPT2) with a SCRIPT command. SCRIPT2 transfers control to SCRIPT3 with a CHAIN command. The last command in SCRIPT3 is EXIT SCRIPT. It causes control to transfer to the line following the outstanding SCRIPT command.

Script 1

```
COMMENT:  Start of SCRIPT1
...
SCRIPT: script2
COMMENT:  Control comes here from SCRIPT3
...
COMMENT:  End of SCRIPT1
```

Script 2

```
COMMENT:  Start of SCRIPT2
...
COMMENT:  We prepared some variables. Now, we give up
COMMENT:    control to SCRIPT3.
CHAIN: script3
COMMENT:  Control cannot get here if the CHAIN command succeeded.
COMMENT:  End of SCRIPT2
```

Script 3

```
COMMENT:  Start of SCRIPT3
...
COMMENT:  We did the important work, so now we return.  We do not
COMMENT:  know or care where control is transferred to.
EXIT SCRIPT:
COMMENT:  End of SCRIPT3
```

More examples of the SCRIPT command appear in Understanding Error Processing.

Results

```
0 - Command Successfully Completed
8 - Cannot find file
```

When control returns to the line following the SCRIPT command, it may be the result of an error in the called script. So, if the current ON ERROR segment executes immediately after a SCRIPT command, it may be the result of the SCRIPT command failing or of an error in the execution of a command in the called script.

SEND

Sends a string of characters through the communication line.

Format

SEND: *string*

Remarks

The script processor examines the *string* for special character translation and then sends it through the communication line.

Example

This script segment logs into a LAT service. It waits for the "Username:" prompt, sends the user name (ending the line with a carriage return character), then waits for the user to type the password.

```
...
ECHO:
PORT: lat
NTS: mysystem
WAIT FOR:Username:
SEND:fried<CR>
WAIT FOR:Password:
TYPE UNTIL:<CR>
...
```

Results

```
0 - Command Successfully Completed
2 - Timeout
```

SEND FROM

Sends the contents of a file through the communication line.

Format

SEND FROM: *file-name*

Remarks

The SEND FROM command opens the file named by *file-name* and sends the contents through the communication line. There is no default for any part of *file-name*. The script processor does not interpret the contents of the file. That is, the script processor treats the contents of the file as a stream of bytes. If you are using this command with a LAT or asynchronous (COM1) connection, the script processor may drop or corrupt characters due to line noise or other factors.

Example

This script segment creates a file on the remote host. It uses SEND FROM commands to transfer the contents of two local files, and it includes titles with SEND commands.

```
...
SEND:create sched.txt<CR>
SEND:Beginning of weekly report:<CR><CR>
SEND FROM: week.txt
SEND:Beginning of monthly report:<CR><CR>
SEND FROM: month.txt
SEND:<Ctrl/Z>
...
```

Results

```
0 - Command Successfully Completed
8 - Cannot find file
```

SEND PASSWORD

Checks the DECnet database for the password and passes it to the remote host for verification. You can use the SEND PASSWORD command as part of a script to log on to a remote host.

Format

SEND PASSWORD: *node-name*

Remarks

Use the NCP DEFINE NODE command to associate your password with a specific node. For example, to associate the ALPHA with a remote node named NAC, issue the following command:

```
NCP> DEFINE NODE NAC PASSWORD ALPHA
```

As part of your log-on script, you would subsequently send a SEND PASSWORD: NAC command. This command is commonly used in association with the SEND USERNAME command. The SEND PASSWORD command provides a security feature because the password resides in the DECnet database where it cannot be displayed.

Example

This example shows a script segment that automatically sends your password to a remote system. This script also employs the SEND USERNAME command.

```
COMMENT: This script sends a username and password to a remote node named NAC.
WAIT FOR:Username:
SEND USERNAME:NAC
WAIT FOR:Password:
SEND PASSWORD:NAC
```

Results

```
 0 - Command Completed Successfully
 4 - Invalid Parameter
24 - ACL Information Not Defined in Database
```

SEND USERNAME

Checks the DECnet database for the username and passes it to the remote host for verification. You can use the SEND USERNAME command as part of a script to log on to a remote host.

Format

SEND USERNAME: [*node-name*]

Remarks

Use the NCP DEFINE NODE command to associate your username with a specific node. For example, to associate the username ALEX with a remote node named NAC, issue the following command:

```
NCP>DEFINE NODE NAC USER ALEX
```

As part of your log-on script, you would subsequently send a SEND USERNAME: NAC command. This command is commonly used in association with the SEND PASSWORD command.

Example

This example shows a script segment that automatically sends your username to a remote system. This script also employs the SEND PASSWORD command.

```
COMMENT: This script sends a username and password to a remote node named NAC.
WAIT FOR: Username:
SEND USERNAME:NAC
WAIT FOR: Password
SEND PASSWORD:NAC
```

Results

```
 0 - Command Completed Successfully
 4 - Invalid Parameter
24 - ACL Information Not Defined in Database
```

SET

Assigns a string to a variable or deletes a variable.

Format

SET: *variable-name* [*string*]

Remarks

With no *string*:

- If *variable-name* exists as a variable, the SET command deletes the variable.
- If *variable-name* does not exist as a variable, the SET command ends with an error.

With *string*:

- If *variable-name* exists as a variable, the SET command replaces its current value with *string*.
- If *variable-name* does not exist as a variable, but the maximum number of variables has been reached, the SET command ends with an error. (The maximum number of variables is 80, including the two standard variables, Error_Number and Error_Message.)
- If *variable-name* does not exist as a variable, the SET command adds it with *string* as its value.
- The script processor does not interpret special character symbols in the SET command *string*. For example, if the command specifies <CR> for *string*, it stores the four characters, not a single carriage return character.
- The standard variables Error_Number and Error_Message can be changed or deleted with the SET command. However, there is no effect because the SET command itself places values into these variables when it ends.

Example

This script segment looks at a variable that was set elsewhere and calls other scripts depending on the value. Then, it deletes the variable. The second CASE command looks at the contents of a variable that the called script returned. If the value is neither "ERROR" nor "WARNING," the script sets the variable to "OK" and returns to the calling script.

```
...
CASE: action
    "A" SCRIPT: proca
    "B" SCRIPT: procb
    "C" SCRIPT: procc
CASE END:
SET: action
CASE: my-result
    "ERROR" EXIT EMULATOR:
    "WARNING" EXIT SCRIPT:
CASE END:
SET: my-result OK
EXIT SCRIPT:
```

Results

```
 0 - Command Successfully Completed
18 - Variable not found
```

SKIP

Passes over lines in the script without executing them.

Format

SKIP: *lines*

Remarks

The SKIP command transfers control forward in the script, ignoring the number of lines specified by *lines*.

The value of *lines* must not be negative.

If a SKIP command transfers control to the line following the last line in the script, the script processor executes an implicit EXIT SCRIPT command.

If a SKIP command attempts to transfer control to any line past the line following the last line in the script, the command ends with an error.

Example

The two script segments do the same basic things. They wait for the string "ok" and run a program on the remote system when it appears. If the scripts do not detect the "ok" and a timeout occurs, both scripts send a logout command to the remote system, then hang up the phone and leave SETHOST.

One difference between the script that uses the SKIP command and the other, which uses ON ERROR segments, is the number of lines of code. The SKIP command saves four lines.

A more significant difference is that the first script segment does not change the active ON ERROR segment. That is, when it finishes, an ON ERROR segment that was active at the beginning is still active. The second script segment, on the other hand, sets up an ON ERROR segment to handle the timeout, then sets up an empty error handler to deactivate the first one. The result is that the original ON ERROR segment, if there was one, would not get control if an error were to occur.

```
...
NO SKIP ON:ok
SKIP: 3
SEND:logout<CR>
HANG UP:
EXIT EMULATOR:
SEND:run myjob<CR>
...

...
ON ERROR:
    SEND:logout<CR>
    HANG UP:
    EXIT EMULATOR:
END ON ERROR:
WAIT FOR:ok
ON ERROR:
END ON ERROR:
SEND:run myjob<CR>
...
```

Results

```
 0 - Command Successfully Completed
 4 - Invalid Parameter
```

SKIP ON

Passes over the next line in the script if a specific string of characters is received.

Format

SKIP ON: *string*

Remarks

The SKIP ON command waits until the script processor receives *string* through the communication line or until the command times out.

If the script processor detects the *string*, it passes over the next line in the script and transfers control to the line that follows it.

If the SKIP ON command times out, the timeout error condition occurs.

If the timeout error condition occurs and control is not transferred elsewhere as a result of an ON ERROR segment or RETRY command, then control passes to the next line, and there is no skip.

See the RETRY and ON ERROR commands and Understanding Error Processing for more information about error processing.

Example

This script segment periodically checks for new mail on a remote system and prints it out on the local printer.

```
$check for mail:
COMMENT:  Call LOGIN.SCR to log in to host
     SCRIPT: login
COMMENT:  Send the MAIL command
     SEND:mail<CR>
COMMENT:  If the string "message" appears, go read the mail.
COMMENT:  Otherwise, go to $Nomail to get out of MAIL.
     SKIP ON:message
     GOTO: $Nomail
COMMENT:  I have mail, so turn on the printer, extract all the
COMMENT:  messages to the terminal (comm line),
COMMENT:  and turn the printer off.
     PRINTER ON:
     SEND:extract/all sys$output<CR>
     PRINTER OFF:
$Nomail:

COMMENT:  Finished.  Get out of MAIL and then log out.
     SEND:exit<CR>
     SEND:logout<CR>
COMMENT:  Wait one hour, then check mail again.
     PAUSE: 1:00:00
     GOTO: $check for mail
```

Results

```
 0 - Command Successfully Completed
 2 - Timeout
```

STOP BITS

Sets the number of stop bits to 1 or 2.

Format

STOP BITS: 1

or

STOP BITS: 2

Remarks

The STOP BITS command is invalid when you use a NETWORK, LAT, CTERM, or TELNET service in a PORT command.

See the PORT command.

Example

The following example sets the number of stop bits to 1:

```
PORT: Data-1
PARITY: None
DATA BITS: 8
STOP BITS: 1
BAUD RATE: 2400
DIAL:B
```

Results

```
0 - Command Successfully Completed
4 - Invalid Parameter
5 - Incompatible Settings
```

SYSTEM

Executes a DOS system command.

Format

SYSTEM: [*command*]

Remarks

If there is no *command*, the DOS prompt appears. The user can return to the script by typing the DOS EXIT command.

If there is a *command*, the script processor passes it to DOS for execution.

Any output from the DOS command appears on the screen.

Example

The following example sets the line state off so that SETHOST can use the COM1 port.

```
SYSTEM:NCP SET LINE STATE OFF
```

Results

```
 0 - Command Successfully Completed
12 - DOS command failed
```

TIMEOUT

Forces an immediate timeout error condition.

Format

TIMEOUT:

Remarks

If the script processor does not transfer control elsewhere as a result of an ON ERROR segment or RETRY command, then it executes an implicit EXIT SCRIPT command, and the script ends.

See the RETRY and ON ERROR commands and Understanding Error Processing for more information about error processing.

Example

This is an example of a counted loop. The commands between the RETRY and TIMEOUT commands execute 10 times, after which control transfers to the ON ERROR segment, if one is active. If there is no active ON ERROR segment, an implicit EXIT SCRIPT command executes.

```
...
RETRY: 9
...
TIMEOUT:
...
```

Results

```
2 - Timeout
```

Note

The result of the TIMEOUT command is always an error condition. Therefore, the standard variable Error_Number cannot have a 0 value.

TIMER

Specifies the timeout period.

Format

TIMER: *seconds*

Remarks

The TIMER command specifies the number of *seconds* in a timeout period.

Until a TIMER command executes in a script session, the default timeout period is 15 seconds.

See the TIMER OFF command.

Example

The following script segment sets up a Hayes-compatible modem to answer a call. It turns the timer off, then waits until the caller connects and sends a HELLO message. It then sets the timer to allow timeouts and continues.

```
...
DTR SET:
SEND:ATS0=1<CR>
TIMER OFF:
WAIT FOR: HELLO
TIMER: 15
...
```

Results

```
0 - Command Successfully Completed
4 - Invalid Parameter
```

TIMER OFF

Disables timeouts.

Format

TIMER OFF:

Remarks

The TIMER OFF command turns off the timeout period, which was set by the TIMER command or by default when the script session started.

While the timeout period is turned off, no command can time out. As a result, some commands, such as WAIT FOR, could wait indefinitely. However, the TIMEOUT command can force a timeout error.

Use the TIMER command to re-enable timeouts.

Example

The following script segment sets up a Hayes-compatible modem to answer a call. It turns the timer off, then waits until the caller connects and sends a HELLO message. It then sets the timer to allow timeouts and continues.

```
...
DTR SET:
SEND:ATS0=1<CR>
TIMER OFF:
WAIT FOR: HELLO
TIMER: 15
...
```

Results

```
0 - Command Successfully Completed
```

TX BAUD

Sets the rate at which characters are both received and transmitted.

Format

TX BAUD: *speed*

Where:

speed is one of the following:

50	300	2400
75	600	3600
110	1200	4800
134	1800	9600
150	2000	19200

Remarks

The TX BAUD command is exactly the same as the BAUD RATE command.

The TX BAUD command is invalid when you use a NETWORK, LAT, CTERM, or TELNET service in a PORT command. See the PORT command.

Example

This script segment sets the baud rate to 2400.

```
PORT: Data-1
PARITY: None
DATA BITS: 8
STOP BITS: 1
RX BAUD: 2400
DIAL:B
```

Results

```
0 - Command Successfully Completed
4 - Invalid Parameter
5 - Incompatible Settings
```

TYPE UNTIL

Sends keyboard input through the communication line until the user types a specific string.

Format

TYPE UNTIL: *string*

Remarks

The TYPE UNTIL command accepts input from the keyboard and sends it through the communication line.

When the script processor detects the characters in *string* in the input, it stops sending keyboard input and executes the next command in the script.

Control keys, such as Ctrl/C and Ctrl/F10 have no effect on SETHOST or the script processor while a TYPE UNTIL command executes. The TYPE UNTIL command sends the generated characters through the communication line.

If keyboard input is disabled by a KEYBOARD OFF command, the TYPE UNTIL command enables it while the command executes. When the command ends, the script processor disables the keyboard again.

The TYPE UNTIL command does not time out. It waits indefinitely for *string* in the input.

Example

This script allows the user to start and stop logging of the remote session by pressing Ctrl/P repeatedly. Each time logging begins, the OPEN command appends the text that the communication line sends to the file REMOTE.TXT.

```
...
    DISPLAY:<CR><LF> Press Ctrl/P to toggle session logging
$loop:
    TYPE UNTIL:<Ctrl/P>
    OPEN: remote.txt
    DISPLAY:Session logging ON<CR><LF>
    TYPE UNTIL:<Ctrl/P>
    CLOSE:
    DISPLAY:Session logging OFF<CR><LF>
    GOTO: $loop
```

Results

```
0 - Command Successfully Completed
```

WAIT FOR

Suspends processing until the script processor receives a specific string of characters over the communication line.

Format

WAIT FOR: *string*

Remarks

The WAIT FOR command waits until the script processor receives *string* over the communications or until the command times out.

If the script processor detects the *string*, the WAIT FOR command ends and execution continues with the next line.

If the WAIT FOR command times out, the timeout error condition occurs.

Example

This script segment logs into a LAT service. It waits for the "Username:" prompt, sends the user name (ending the line with a carriage return character), then waits for the user to type the password.

```
...
ECHO:
PORT: lat
NTS: mysystem
WAIT FOR:Username:
SEND:fried<CR>
WAIT FOR:Password:
TYPE UNTIL:<CR>
...
```

Results

```
0 - Command Successfully Completed
2 - Timeout
```

XON/XOFF

Causes the script processor to use the XON/XOFF protocol.

Format

XON/XOFF: *buffer-limit*

Where:

buffer-limit is one of the following values:

64
256
512
1024

Remarks

The XON/XOFF command turns on or off use of the XON/OFF protocol and sets the number of characters that triggers XOFF.

The *buffer-limit* specifies the number of characters in the buffer at or above which SETHOST sends an XOFF character to stop transmission.

Example

This example sets the buffer limit to 64:

```
XON/XOFF:64
```

Results

```
0 - Command Successfully Completed
```

A

The SETHOST Default Settings

SETHOST has many of its parameters already set to particular default values. Refer to Table A–1 for a description of the default values associated with each Set-Up screen.

Table A–1 Default Settings

Set-Up Screen	Default Parameters
Communications Screen	Comm Port 1: Data Leads Only Disconnect, 2-second delay XOFF at 64 Speed = 9600 8 data bits—No Parity 1 stop bit Transmit = Receive
Display Screen	80 Columns Auto Wrap Off Dark Background Block Cursor Visible Cursor Color Background:Blue Reverse:Magenta Bold:Cyan Foreground:White Underline:

(continued on next page)

Table A–1 (Cont.) Default Settings

Set-Up Screen	Default Parameters
General Screen Default	On-Line Local Echo Off No New Line Multinational Mode DEC Multinational Character Set Slow (Video Bios) VT300 7-Bit Controls User-Defined Keys Unlocked User Features Unlocked
Keyboard Screen	Numeric Keypad Normal Cursor Keys Margin Bell Warning Bell Break Auto Answerback Answerback:Not Concealed Answerback String IBM XT Keymap For LK250: Digital LK250 Normal Auto Repeat Normal Keyclick Keyboard Dialect: United States
Printer Screen	Printer Disconnected Print Full Page DEC Printer Print File Name Normal Print Mode Print Terminator:Form Feed

B

SETHOST Error Messages and Codes

This appendix lists the error messages and codes you can encounter when using SETHOST. Errors may occur during:

- LAT connection
- LAT sessions
- CTERM connection
- CTERM sessions
- Network operations
- TCP/IP connection
- TELNET connection

For Connections Using the Local Area Transport (LAT) Protocol

You can encounter these messages at the following times:

- During initial connect
- During a session

During Initial Connect

The following messages can appear during initial contact of a LAT session:

Service not known

Explanation: The personal computer's LAT service table does not contain that name.

User Action: Add the node to service table using LATCP. See the *User's Handbook* for more information on adding a node.)

Start session error

Explanation: The personal computer's LAT driver has depleted all its resources. Causes:

- No more virtual circuit blocks
- No more sessions available
- No more sessions available on this circuit

Unable to allocate SCB from LAT

Explanation: NVTWIN is set to 1 and SETHOST cannot allocate a session control block from LAT.

User Action: Refer to the PATHWORKS for DOS, *Client Commands Reference* for information on how to add session control blocks (SCBs) to the LAT driver.

During a Session

You can encounter the following LAT messages during your SETHOST session.

These messages are numbered. The first 2 digits represent the error message number and the last 2 digits represent the connection status. The 08 in the following error messages represents a broken connection.

LAT Reason: 0108

Explanation: LAT received a Stop Slot message from the remote system causing LAT to break the connection. This error is a host problem, caused by the host dropping the connection.

User Action: Retry the connection.

LAT Reason: 0208

Explanation: LAT received a Stop message from the remote system causing LAT to break the connection. This error is a host problem, caused by the host dropping the connection.

User Action: Retry the connection.

LAT Reason: 0308

Explanation: LAT reached the retransmit limit and broke the connection. If the default is used, that means SETHOST tried 8 times to send a message and got no response from the remote system. It may also be a network hardware problem.

User Action: Increase the retransmit count for LAT with LATCP. See the *Client Commands Reference* for information on the LATCP command. If that does not work, it is probably a hardware problem, see your system administrator.

LAT Reason: 0408

Explanation: LAT received a message containing an illegal slot and broke the connection.

User Action: Retry the connection. If that does not work, tell your system adminstrator.

LAT Reason: 0508

Explanation: LAT received an illegal message and broke the connection.

User Action: Retry the connection. If that does not work, tell your system adminstrator.

For Connections Using the CTERM Protocol over DECnet

You can encounter these messages at the following times:

- During initial connect
- During a session

This section refers to CTERM as a driver. CTERM is actually a Terminate and Stay Resident (TSR) program.

During Initial Connect

The following messages can appear during the initial contact of a CTERM session:

Unrecognized node name

Explanation: DECnet does not recognize the node name (DECnet error 49). This is the most common error.

User Action: Use NCP to set up the node name.

You can receive DECnet error messages during initial contact. See Table B–1 for the list of DECnet reason codes and the meanings associated with each code.

During a Session

You can encounter the following messages during your SETHOST session.

These messages are numbered. The first two digits represent the connection status and the last two digits represent the error message number. In the following error messages, a leading 80 means the connection has broken and a leading C0 means a DECnet error caused the connection break.

CTERM reason: 8001

Explanation: The CTERM driver received an unknown protocol message from the remote system and broke the connection.

User Action: Retry the connection.

CTERM reason: 8002

Explanation: The remote system violated the CTERM protocol and broke the connection.

User Action: Retry the connection.

CTERM reason: C003

Network Reason: See Table B–1 for the list of reason codes you can encounter during a SETHOST session.

Explanation: The CTERM driver did not process the CTERM initiate message and broke the connection.

User Action: See Table B–1 for a list of the meanings associated with each reason code. This error is a network error, the Network error list provides the reason.

CTERM reason: C004

Network Reason: See Table B–1 for the list of reason codes you can encounter during a SETHOST session.

Explanation: Connection was broken while the CTERM driver was receiving a message from the remote system.

User Action: See Table B–1 for a list of the meanings associated with each reason code. This error is a network error, the Network error list provides the reason.

CTERM reason: C005

Network Reason: See Table B–1 for the list of reason codes you may encounter during a SETHOST session.

Explanation: Connection was broken while the CTERM driver was sending a message to the remote system.

User Action: See Table B–1 for a list of the meanings associated with each reason code. This error is a network error, the Network error list provides the reason.

CTERM reason: C006

Network Reason: See Table B–1 for the list of reason codes you can encounter during a SETHOST session.

Explanation: Connection was broken when the CTERM driver checked for a message from the remote system.

User Action: See Table B–1 for a list of the meanings associated with each reason code. This error is a network error, the Network error list provides the reason.

CTERM reason: 8007

Explanation: The remote system broke the connection because it does not support the CTERM protocol.

User Action: Try to connect to another remote system.

CTERM reason: 8008

Explanation: The remote system broke the connection because it does not support the correct CTERM protocol version.

User Action: Try to connect to another remote system.

CTERM reason: C009

Network Reason: See Table B–1 for the list of reason codes you can encounter during a SETHOST session.

Explanation: CTERM broke the connection because the CTERM driver did not receive a BIND request message from the remote system.

User Action: See Table B–1 for a list of the meanings associated with each reason code. This error is a network error, the Network error list provides the reason.

CTERM reason: C00A

Network Reason: See Network Reason Codes for CTERM for the list of reason codes you can encounter during a SETHOST session.

Explanation: CTERM broke the connection because the CTERM driver could not send a BIND request message to the remote system.

User Action: See Table B–1 for a list of the meanings associated with each reason code. This error is a network error, the Network error list provides the reason.

CTERM reason: 800B

Explanation: CTERM broke the connection because there are no more CTERM sessions available. The maximum number of sessions the CTERM driver is built for is 32. This error occurs when you are using windows or are in a task-switching environment and have 32 simultaneous SETHOST CTERM sessions active.

User Action: You cannot have more than 32 CTERM sessions. Log out of one or more.

CTERM reason: 800C

Explanation: CTERM broke the connection because the session handle passed to the CTERM driver did not match an existing session.

User Action: This is a serious error in your local system. If the problem occurs, see your system administrator. The cure is to exit to DOS, close up all open files, and restart your personal computer.

CTERM reason: 800D

Explanation: CTERM broke the connection because the CTERM driver did not have enough memory to complete the operation.

User Action: There is no cure for this error. Have your system administrator file a software performance report. This error is rare.

CTERM reason: C00E

Explanation: Connection is broken.

User Action: See Table B–1 for a list of the meanings associated with each reason code. This error is a network error, the Network error list provides the reason.

Network Reason Codes for CTERM

Table B–1 lists the network reason codes and the meanings associated with each code.

Table B–1 Network Reason Codes and Meanings

Reason Code	Meaning
0	Connection rejected by object
1	Insufficient network resources
2	Unrecognized node name
3	Network object is unknown at remote node
4	Unrecognized object
5	Invalid object name format
6	Object too busy
7	Argument list too long
8	Execute format error
9	Bad file number
10	Invalid node name format
11	Local node shutting down
12	Not enough core
13	Permission denied
14	Bad address
15	Block device required
16	Mount device busy
17	File exists
18	Cross-device disk
19	No such device

(continued on next page)

Table B–1 (Cont.) Network Reason Codes and Meanings

Reason Code	Meaning
20	Not a directory
21	Is a directory
22	Invalid argument
23	File table overflow
24	Too many open files
25	Not a typewriter
26	Text file busy
27	File too large
28	No space left on device
29	Illegal seek
30	Read-only file system
31	Too many links
32	No node resources for new logical link
33	No user resources for new logical link
34	Access control rejected
35	Operation would block
36	Bad account information
37	Operation already in progress
38	No response from object
39	Node unreachable
40	Message too long
41	Protocol wrong type for socket
42	Protocol not available
43	Connect image data field too long
44	Socket type not supported
45	Operation not supported on socket
46	Protocol family not supported
47	Address family not supported by protocol family

(continued on next page)

Table B–1 (Cont.) Network Reason Codes and Meanings

Reason Code	Meaning
48	Address already in use
49	Cannot assign requested address
50	Network is down
51	Network is unreachable
52	Network dropped connection on reset
53	Software caused connection abort
54	Connection reset by peer
55	No buffer space available
56	Socket is already connected
57	Socket is not connected
58	Cannot send after socket shutdown
59	Too many references: cannot splice
60	Connection timed out
61	Connection refused
62	Too many levels of symbolic links
63	File name is too long
64	Host is down
65	Host is unreachable
66	Directory not empty
67	Too many processes
68	Too many users
69	Disk quota exceeded

Note

Some of these reason codes do not apply to CTERM, but the network software could return them. They are included for completeness.

For Connections Using the TELNET Protocol over TCP/IP

You can encounter the following codes when using a TCP/IP connection. Refer to the PATHWORKS for DOS, *TCP/IP User's Reference* for more information about TCP/IP Network Messages.

Bridge Architecture Protocol Interface Error Codes for TELNET Connections

Table B–2 lists the Bridge Architecture Protocol Interface (BAPI) error codes and the meanings associated with each value.

Table B–2 BAPI Session ID and Error Parameter Values

Value	Meaning
Session ID Parameter Values	
00h	External Session Management
01	Session 1
02	Session 2
nn	Session n, nn <FFh
Error Parameter Values	
00h	Success
01	No characters written
02	No characters read
03	No such session
04	Clearinghouse name not found
05	No response from host
06	No more sessions available
07	Session aborted
08	Invalid clearinghouse name
09	Not supported
10	Internal (general) network error
11	Out of memory
12	Invalid Internet Protocol (IP) address

C

SETHOST Character Sets

This appendix includes figures and tables that illustrate the international character sets available during terminal emulation. You can select or change character sets from the General Set-Up menu.

The character set files that SETHOST uses are stored in subdirectories based on your path. These are ASCII text files, containing two 256-character tables.

The first figure is the DECIBM table. The DEC character is indexed into this table to find the corresponding IBM character (which is then displayed on the screen).

The second figure is the IBMDEC table. The IBM character is indexed into this table to find the corresponding DEC character to send to the remote system.

Because SETHOST uses the IBM ROM character set and not a graphic character set for displaying characters, you are limited to certain characters. When you select a character from the character set file, SETHOST tries to determine the closest match to the character you want to display.

Note

If you need to conserve disk space, store only the character set files that you plan to use. If you are using only the ASCII and DEC Multinational characters, those are the defaults that SETHOST uses if it cannot find the character set file that you selected.

Note the following information when using the character set translation files:

- When you install PATHWORKS for DOS, the install program copies all of the Character Translation Files (*.CHR).
- These files are required for both the Personal Computer SETHOST, NFT, and PCMAIL support of character translations.

- You can delete these files (except for the country file desired) from your disk to conserve disk space.
- To perform 8-bit character translation, SETHOST uses the ISO.CHR and DECM.CHR character translation.

SETHOST Character Sets

To configure SETHOST for use with international keyboards, follow these steps:

1. Follow your system or keyboard instructions for setting up international mapping. This usually consists of running a DOS KEYB program that replaces the resident keyboard program. For example, with MS-DOS V3.2 you would use KEYBFR.COM to map your keyboard to a French keyboard.
2. Determine which Digital character set you need to use to access the remote system.
3. Press F3 to get into the Set-Up menu.
4. Use Pg Up or Pg Dn to get to the General menu.
5. Use ⇧ or ⇩ to move to the line indicating *Multinational Mode* or *NRC Mode*. This line lets you select between 8-bit or 7-bit character sets. The next line allows you to select between either a list of DEC 8-bit character sets or DEC 7-bit character sets.
6. Use ⇨ to move to the SELECTIONS column.
7. Use the ⇧ and ⇩ to highlight the character set that you want. Select the highlighted character set by pressing Return.
8. Save this as the new default by using the Save Set-Up Parameters option on the Actions menu.

Illustrations Figure C–1 through Figure C–5 show the character sets you can use with SETHOST. The special graphics character set (also known as VT100 line drawing) and the DEC Technical Character Set (TCS) are built into SETHOST.

Figure C–1 DEC Special Graphics Character Set

ROW	BITS B7 B6 B5 B4 B3 B2 B1	COLUMN 0		1		2		3		4		5		6		7	
		0 0 0		0 0 1		0 1 0		0 1 1		1 0 0		1 0 1		1 1 0		1 1 1	
0	0 0 0 0	NUL	0 0 0	DLE	20 16 10	SP	40 32 20	0	60 48 30	@	100 64 40	P	120 80 50	♦	140 96 60	— SCAN 3	160 112 70
1	0 0 0 1	SOH	1 1 1	DC1 (XON)	21 17 11	!	41 33 21	1	61 49 31	A	101 65 41	Q	121 81 51	␤	141 97 61	— SCAN 5	161 113 71
2	0 0 1 0	STX	2 2 2	DC2	22 18 12	"	42 34 22	2	62 50 32	B	102 66 42	R	122 82 52	␋	142 98 62	— SCAN 7	162 114 72
3	0 0 1 1	ETX	3 3 3	DC3 (XOFF)	23 19 13	#	43 35 23	3	63 51 33	C	103 67 43	S	123 83 53	␌	143 99 63	— SCAN 9	163 115 73
4	0 1 0 0	EOT	4 4 4	DC4	24 20 14	$	44 36 24	4	64 52 34	D	104 68 44	T	124 84 54	␍	144 100 64	├	164 116 74
5	0 1 0 1	ENQ	5 5 5	NAK	25 21 15	%	45 37 25	5	65 53 35	E	105 69 45	U	125 85 55	␊	145 101 65	┤	165 117 75
6	0 1 1 0	ACK	6 6 6	SYN	26 22 16	&	46 38 26	6	66 54 36	F	106 70 46	V	126 86 56	0	146 102 66	⊥	166 118 76
7	0 1 1 1	BEL	7 7 7	ETB	27 23 17	'	47 39 27	7	67 55 37	G	107 71 47	W	127 87 57	±	147 103 67	┬	167 119 77
8	1 0 0 0	BS	10 8 8	CAN	30 24 18	(	50 40 28	8	70 56 38	H	110 72 48	X	130 88 58	␤	150 104 68	│	170 120 78
9	1 0 0 1	HT	11 9 9	EM	31 25 19	)	51 41 29	9	71 57 39	I	111 73 49	Y	131 89 59	␋	151 105 69	≤	171 121 79
10	1 0 1 0	LF	12 10 A	SUB	32 26 1A	*	52 42 2A	:	72 58 3A	J	112 74 4A	Z	132 90 5A	┘	152 106 6A	≥	172 122 7A
11	1 0 1 1	VT	13 11 B	ESC	33 27 1B	+	53 43 2B	;	73 59 3B	K	113 75 4B	[	133 91 5B	┐	153 107 6B	π	173 123 7B
12	1 1 0 0	FF	14 12 C	FS	34 28 1C	,	54 44 2C	<	74 60 3C	L	114 76 4C	\	134 92 5C	┌	154 108 6C	≠	174 124 7C
13	1 1 0 1	CR	15 13 D	GS	35 29 1D	-	55 45 2D	=	75 61 3D	M	115 77 4D	]	135 93 5D	└	155 109 6D	£	175 125 7D
14	1 1 1 0	SO	16 14 E	FS	36 30 1E	.	56 46 2E	>	76 62 3E	N	116 78 4E	^	136 94 5E	┼	156 110 6E	·	176 126 7E
15	1 1 1 1	SI	17 15 F	US	37 31 1F	/	57 47 2F	?	77 63 3F	O	117 79 4F	(BLANK)	137 95 5F	— SCAN 1	157 111 6F	DEL	177 127 7F

C0 CODES

GL CODES (DEC SPECIAL GRAPHICS)

KEY

CHARACTER	ESC	33	OCTAL
		27	DECIMAL
		18	HEX

LJO-0005-PM

Figure C–2 ISO Latin-1 Character Set (Left Half)

B8 B7 B6 B5 BITS B4 B3 B2 B1	ROW	COLUMN 0		COLUMN 1		COLUMN 2		COLUMN 3		COLUMN 4		COLUMN 5		COLUMN 6		COLUMN 7	
0 0 0 0	0	NUL	0 0 0	DLE	20 16 10	SP	40 32 20	0	60 48 30	@	100 64 40	P	120 80 50	`	140 96 60	p	160 112 70
0 0 0 1	1	SOH	1 1 1	DC1	21 17 11	!	41 33 21	1	61 49 31	A	101 65 41	Q	121 81 51	a	141 97 61	q	161 113 71
0 0 1 0	2	STX	2 2 2	DC2	22 18 12	"	42 34 22	2	62 50 32	B	102 66 42	R	122 82 52	b	142 98 62	r	162 114 72
0 0 1 1	3	ETX	3 3 3	DC3	23 19 13	#	43 35 23	3	63 51 33	C	103 67 43	S	123 83 53	c	143 99 63	s	163 115 73
0 1 0 0	4	EOT	4 4 4	DC4	24 20 14	$	44 36 24	4	64 52 34	D	104 68 44	T	124 84 54	d	144 100 64	t	164 116 74
0 1 0 1	5	ENQ	5 5 5	NAK	25 21 15	%	45 37 25	5	65 53 35	E	105 69 45	U	125 85 55	e	145 101 65	u	165 117 75
0 1 1 0	6	ACK	6 6 6	SYN	26 22 16	&	46 38 26	6	66 54 36	F	106 70 46	V	126 86 56	f	146 102 66	v	166 118 76
0 1 1 1	7	BEL	7 7 7	ETB	27 23 17	'	47 39 27	7	67 55 37	G	107 71 47	W	127 87 57	g	147 103 67	w	167 119 77
1 0 0 0	8	BS	10 8 8	CAN	30 24 18	(	50 40 28	8	70 56 38	H	110 72 48	X	130 88 58	h	150 104 68	x	170 120 78
1 0 0 1	9	HT	11 9 9	EM	31 25 19	)	51 41 29	9	71 57 39	I	111 73 49	Y	131 89 59	i	151 105 69	y	171 121 79
1 0 1 0	10	LF	12 10 A	SUB	32 26 1A	*	52 42 2A	:	72 58 3A	J	112 74 4A	Z	132 90 5A	j	152 106 6A	z	172 122 7A
1 0 1 1	11	VT	13 11 B	ESC	33 27 1B	+	53 43 2B	;	73 59 3B	K	113 75 4B	[	133 91 5B	k	153 107 6B	{	173 123 7B
1 1 0 0	12	FF	14 12 C	FS	34 28 1C	,	54 44 2C	<	74 60 3C	L	114 76 4C	\	134 92 5C	l	154 108 6C	|	174 124 7C
1 1 0 1	13	CR	15 13 D	GS	35 29 1D	-	55 45 2D	=	75 61 3D	M	115 77 4D	]	135 93 5D	m	155 109 6D	}	175 125 7D
1 1 1 0	14	SO	16 14 E	RS	36 30 1E	.	56 46 2E	>	76 62 3E	N	116 78 4E	^	136 94 5E	n	156 110 6E	~	176 126 7E
1 1 1 1	15	SI	17 15 F	US	37 31 1F	/	57 47 2F	?	77 63 3F	O	117 79 4F	_	137 95 5F	o	157 111 6F	DEL	177 127 7F

C0 CODES

KEY

ASCII CHARACTERS

ESC	1/11	COLUMN ROW
	33	OCTAL
	27	DECIMAL
	18	HEX

HIGHLIGHTS THE NONDISPLAYABLE CHARACTERS

LJO-0003-PM

Figure C–3 ISO Latin-1 Character Set (Right Half)

8		9		10		11		12		13		14		15		BITS B4 B3 B2 B1	ROW
	200 128 80	DCS	220 144 90	NBSP	240 160 A0	°	260 176 B0	À	300 192 C0	Ð	320 208 D0	à	340 224 E0	ð	360 240 F0	0 0 0 0	0
	201 129 81	PU1	221 145 91	¡	241 161 A1	±	261 177 B1	Á	301 193 C1	Ñ	321 209 D1	á	341 225 E1	ñ	361 241 F1	0 0 0 1	1
	202 130 82	PU2	222 146 92	¢	242 162 A2	²	262 178 B2	Â	302 194 C2	Ò	322 210 D2	â	342 226 E2	ò	362 242 F2	0 0 1 0	2
	203 131 83	STS	223 147 93	£	243 163 A3	³	263 179 B3	Ã	303 195 C3	Ó	323 211 D3	ã	343 227 E3	ó	363 243 F3	0 0 1 1	3
IND	204 132 84	CCH	224 148 94	¤	244 164 A4	´	264 180 B4	Ä	304 196 C4	Ô	324 212 D4	ä	344 228 E4	ô	364 244 F4	0 1 0 0	4
NEL	205 133 85	MW	225 149 95	¥	245 165 A5	µ	265 181 B5	Å	305 197 C5	Õ	325 213 D5	å	345 229 E5	õ	365 245 F5	0 1 0 1	5
SSA	206 134 86	SPA	226 150 96	¦	246 166 A6	¶	266 182 B6	Æ	306 198 C6	Ö	326 214 D6	æ	346 230 E6	ö	366 246 F6	0 1 1 0	6
ESA	207 135 87	EPA	227 151 97	§	247 167 A7	·	267 183 B7	Ç	307 199 C7	×	327 215 D7	ç	347 231 E7	÷	367 247 F7	0 1 1 1	7
HTS	210 136 88		230 152 98	¨	250 168 A8	¸	270 184 B8	È	310 200 C8	Ø	330 216 D8	è	350 232 E8	ø	370 248 F8	1 0 0 0	8
HTJ	211 137 89		231 153 99	©	251 169 A9	¹	271 185 B9	É	311 201 C9	Ù	331 217 D9	é	351 233 E9	ù	371 249 F9	1 0 0 1	9
VTS	212 138 8A		232 154 9A	ª	252 170 AA	º	272 186 BA	Ê	312 202 CA	Ú	332 218 DA	ê	352 234 EA	ú	372 250 FA	1 0 1 0	10
PLD	213 139 8B	CSI	233 155 9B	«	253 171 AB	»	273 187 BB	Ë	313 203 CB	Û	333 219 DB	ë	353 235 EB	û	373 251 FB	1 0 1 1	11
PLU	214 140 8C	ST	234 156 9C	¬	254 172 AC	¼	274 188 BC	Ì	314 204 CC	Ü	334 220 DC	ì	354 236 EC	ü	374 252 FC	1 1 0 0	12
RI	215 141 8D	OSC	235 157 9D		255 173 AD	½	275 189 BD	Í	315 205 CD	Ý	335 221 DD	í	355 237 ED	ý	375 253 FD	1 1 0 1	13
SS2	216 142 8E	PM	236 158 9E	®	256 174 AE	¾	276 190 BE	Î	316 206 CE	Þ	336 222 DE	î	356 238 EE	þ	376 254 FE	1 1 1 0	14
SS3	217 143 8F	APC	237 159 9F	¯	257 175 AF	¿	277 191 BF	Ï	317 207 CF	ß	337 223 DF	ï	357 239 EF	ÿ	377 255 FF	1 1 1 1	15

C1 CODES — GR CODES

HIGHLIGHTS THE NONDISPLAYABLE CHARACTERS

LJO-0004-PM

Figure C–4 DEC Multinational Character Set (Left Half)

ROW	COLUMN 0		COLUMN 1		COLUMN 2		COLUMN 3		COLUMN 4		COLUMN 5		COLUMN 6		COLUMN 7	
BITS B8 B7 B6 B5 B4 B3 B2 B1	0 0 0 0		0 0 0 1		0 0 1 0		0 0 1 1		0 1 0 0		0 1 0 1		0 1 1 0		0 1 1 1	
0	0 0 0 0	NUL 0 0 0	DLE 20 16 10		SP 40 32 20		0 60 48 30		@ 100 64 40		P 120 80 50		` 140 96 60		p 160 112 70	
1	0 0 0 1	SOH 1 1 1	DC1 21 17 11		! 41 33 21		1 61 49 31		A 101 65 41		Q 121 81 51		a 141 97 61		q 161 113 71	
2	0 0 1 0	STX 2 2 2	DC2 22 18 12		" 42 34 22		2 62 50 32		B 102 66 42		R 122 82 52		b 142 98 62		r 162 114 72	
3	0 0 1 1	ETX 3 3 3	DC3 23 19 13		# 43 35 23		3 63 51 33		C 103 67 43		S 123 83 53		c 143 99 63		s 163 115 73	
4	0 1 0 0	EOT 4 4 4	DC4 24 20 14		$ 44 36 24		4 64 52 34		D 104 68 44		T 124 84 54		d 144 100 64		t 164 116 74	
5	0 1 0 1	ENQ 5 5 5	NAK 25 21 15		% 45 37 25		5 65 53 35		E 105 69 45		U 125 85 55		e 145 101 65		u 165 117 75	
6	0 1 1 0	ACK 6 6 6	SYN 26 22 16		& 46 38 26		6 66 54 36		F 106 70 46		V 126 86 56		f 146 102 66		v 166 118 76	
7	0 1 1 1	BEL 7 7 7	ETB 27 23 17		' 47 39 27		7 67 55 37		G 107 71 47		W 127 87 57		g 147 103 67		w 167 119 77	
8	1 0 0 0	BS 10 8 8	CAN 30 24 18		(50 40 28		8 70 56 38		H 110 72 48		X 130 88 58		h 150 104 68		x 170 120 78	
9	1 0 0 1	HT 11 9 9	EM 31 25 19		) 51 41 29		9 71 57 39		I 111 73 49		Y 131 89 59		i 151 105 69		y 171 121 79	
10	1 0 1 0	LF 12 10 A	SUB 32 26 1A		* 52 42 2A		: 72 58 3A		J 112 74 4A		Z 132 90 5A		j 152 106 6A		z 172 122 7A	
11	1 0 1 1	VT 13 11 B	ESC 33 27 1B		+ 53 43 2B		; 73 59 3B		K 113 75 4B		[133 91 5B		k 153 107 6B		{ 173 123 7B	
12	1 1 0 0	FF 14 12 C	FS 34 28 1C		, 54 44 2C		< 74 60 3C		L 114 76 4C		\ 134 92 5C		l 154 108 6C		| 174 124 7C	
13	1 1 0 1	CR 15 13 D	GS 35 29 1D		- 55 45 2D		= 75 61 3D		M 115 77 4D		] 135 93 5D		m 155 109 6D		} 175 125 7D	
14	1 1 1 0	SO 16 14 E	RS 36 30 1E		. 56 46 2E		> 76 62 3E		N 116 78 4E		^ 136 94 5E		n 156 110 6E		~ 176 126 7E	
15	1 1 1 1	SI 17 15 F	US 37 31 1F		/ 57 47 2F		? 77 63 3F		O 117 79 4F		_ 137 95 5F		o 157 111 6F		DEL 177 127 7F	

C0 CODES

GL CODES (ASCII GRAPHICS)

KEY

ASCII CHARACTERS

ESC	1/11	COLUMN ROW
	33	OCTAL
	27	DECIMAL
	18	HEX

LJO-0001-PM

Figure C–5 DEC Multinational Character Set (Right Half)

8		9		10		11		12		13		14		15		COLUMN BITS B8 B7 B6 B5 B4 B3 B2 B1	ROW
1 0 0 0		1 0 0 1		1 0 1 0		1 0 1 1		1 1 0 0		1 1 0 1		1 1 1 0		1 1 1 1			
	200 128 80	DCS	220 144 90		240 160 A0	º	260 176 B0	À	300 192 C0		320 208 D0	à	340 224 E0		360 240 F0	0 0 0 0	0
	201 129 81	PU1	221 145 91	¡	241 161 A1	±	261 177 B1	Á	301 193 C1	Ñ	321 209 D1	á	341 225 E1	ñ	361 241 F1	0 0 0 1	1
	202 130 82	PU2	222 146 92	¢	242 162 A2	²	262 178 B2	Â	302 194 C2	Ò	322 210 D2	â	342 226 E2	ò	362 242 F2	0 0 1 0	2
	203 131 83	STS	223 147 93	£	243 163 A3	³	263 179 B3	Ã	303 195 C3	Ó	323 211 D3	ã	343 227 E3	ó	363 243 F3	0 0 1 1	3
IND	204 132 84	CCH	224 148 94		244 164 A4		264 180 B4	Ä	304 196 C4	Ô	324 212 D4	ä	344 228 E4	ô	364 244 F4	0 1 0 0	4
NEL	205 133 85	MW	225 149 95	¥	245 165 A5	µ	265 181 B5	Å	305 197 C5	Õ	325 213 D5	å	345 229 E5	õ	365 245 F5	0 1 0 1	5
SSA	206 134 86	SPA	226 150 96		246 166 A6	¶	266 182 B6	Æ	306 198 C6	Ö	326 214 D6	æ	346 230 E6	ö	366 246 F6	0 1 1 0	6
ESA	207 135 87	EPA	227 151 97	§	247 167 A7	·	267 183 B7	Ç	307 199 C7	Œ	327 215 D7	ç	347 231 E7	œ	367 247 F7	0 1 1 1	7
HTS	210 136 88		230 152 98	¤	250 168 A8		270 184 B8	È	310 200 C8	Ø	330 216 D8	è	350 232 E8	ø	370 248 F8	1 0 0 0	8
HTJ	211 137 89		231 153 99	©	251 169 A9	¹	271 185 B9	É	311 201 C9	Ù	331 217 D9	é	351 233 E9	ù	371 249 F9	1 0 0 1	9
VTS	212 138 8A		232 154 9A	ª	252 170 AA	º	272 186 BA	Ê	312 202 CA	Ú	332 218 DA	ê	352 234 EA	ú	372 250 FA	1 0 1 0	10
PLD	213 139 8B	CSI	233 155 9B	«	253 171 AB	»	273 187 BB	Ë	313 203 CB	Û	333 219 DB	ë	353 235 EB	û	373 251 FB	1 0 1 1	11
PLU	214 140 8C	ST	234 156 9C		254 172 AC	¼	274 188 BC	Ì	314 204 CC	Ü	334 220 DC	ì	354 236 EC	ü	374 252 FC	1 1 0 0	12
RI	215 141 8D	OSC	235 157 9D		255 173 AD	½	275 189 BD	Í	315 205 CD	Ÿ	335 221 DD	í	355 237 ED	ÿ	375 253 FD	1 1 0 1	13
SS2	216 142 8E	PM	236 158 9E		256 174 AE		276 190 BE	Î	316 206 CE		336 222 DE	î	356 238 EE		376 254 FE	1 1 1 0	14
SS3	217 143 8F	APC	237 159 9F		257 175 AF	¿	277 191 BF	Ï	317 207 CF	ß	337 223 DF	ï	357 239 EF		377 255 FF	1 1 1 1	15

C1 CODES

GR CODES
(DEC SUPPLEMENTAL GRAPHICS)

LJO-0002-PM

Table C–1 and Table C–2 list the keys definitions that are different in the national replacement character sets. The numbers in the table headings represent the key location.

Table C–1 National Replacement Character Sets

Character Set	2/3	4/0	5/11	5/12	5/13	5/14
ASCII	#	@	[	\	]	^
British	£	@	[	\	]	^
Dutch	£	¾	ÿ	½	|	^
Finnish	#	@	Ä	Ö	Å	Ü
French	£	à	°	ç	§	^
French Canadian	#	à	â	ç	ê	î
German	#	§	Ä	Ö	Ü	^
Italian	£	§	°	ç	é	î
Norwegian/Danish	#	@	Æ	Ø	Å	^
Portuguese	#	@	Ã	Ç	Õ	^
Spanish	£	§	¡	Ñ	¿	^
Swedish	#	É	Ä	Ö	Å	Ü
Swiss	ù	à	é	ç	ê	î

Table C–2 National Replacement Character Sets (continued)

Character Set	5/15	6/0	7/11	7/12	7/13	7/14
ASCII	-	~	{	|	}	~
British	-	~	{	|	}	~
Dutch	-	~	..	f	¼	’
Finnish	-	è	ä	ö	å	ü
French	-	‘	è	ú	é	..
French Canadian	-	ô	é	ù	è	û
German	-	‘	ä	ö	ü	ß
Italian	-	ù	à	ò	è	ì

(continued on next page)

Table C–2 (Cont.) National Replacement Character Sets (continued)

Character Set	5/15	6/0	7/11	7/12	7/13	7/14
Norwegian/Danish	-	‘	æ	ø	å	~
Portuguese	-	‘	ã	ç	õ	~
Spanish	-	‘	‘	°	ñ	ç
Swedish	-	é	ä	ö	å	ü
Swiss	è	ô	ä	ö	ü	û

Multinational Characters and VT100 Emulation

If you select VT100 emulation and you are using ALL-IN-1 or WPS-PLUS software, DEC multinational characters are not displayed. In order to get WPS-PLUS software to use the multinational character set (MCS) and to properly display the MCS characters in the on-line help screens, you need to define the following symbols on your VMS system:

DEFINE KOA$TERMINAL_*xxxx* "OUTPUT_SETHOST"
DEFINE KOA$TERMINAL_MCS_SETHOST "Y"

Where:

xxxx is the terminal name (such as RTA1, VTA32, and so on).

Code Page Support

Table C–3 lists the code pages that SETHOST supports.

Table C–3 SETHOST Code Page Support

Code Page	File
437 United States (same as STD ROM)	CODE437.SRC
865 Nordic (same as STD2 ROM)	CODE865.SRC
850 Multilingual	CODE850.SRC
860 Portuguese	CODE860.SRC
863 Canadian-French	CODE863.SRC

These code page files contain the character set translation files that SETHOST uses to go between DEC character sets and a personal computer character set. You can use the BREAKSRC.EXE file on the kit to extract the contents of these .SRC files (for example, BREAKSRC CODE437.SRC). The .SCR files are in the \DECNET\SOURCE directory.

For each character set listed in the General screen in Set-Up there is an external text file that provides two translation tables to translate between a DEC character set and a personal computer character set. SETHOST looks for these files on the path. The NFT and PCMAIL utilities also use these files.

Each code page has its own directory that contains 15 character set files. These directories appear as top level directories on the server.

Table C–4 SETHOST Code Page Directory Names

Code Page	Directory Name
437 United States	437USA
865 Nordic	865NOR
850 Multilingual	850MUL
860 Portuguese	860POR
863 Canadian-French	863CNF

Each of the code page directories contain the following character set files:

- ASCII.CHR
- BRITISH.CHR
- CANADIAN.CHR
- DECM.CHR
- DUTCH.CHR
- FINNISH.CHR
- FRENCH.CHR
- GERMAN.CHR
- ISO.CHR
- ITALIAN.CHR
- NORDAN.CHR
- PORTUGES.CHR
- SPANISH.CHR
- SWEDISH.CHR
- SWISS.CHR

Glossary

PATHWORKS Glossary

The terms that appeared in the text of this book in **boldface** are explained in this glossary. Additional computer-related terms are also explained here.

access (v.)

To use a resource, such as a printer, directory, or disk drive.

account (n.)

A set of information on a computer system that allows users access to a multiuser or networked computer. It includes the user's name, often a password, other identifiers, a list of services and privileges the user is allowed, and files belonging to the user.

alphanumeric (adj.)

Pertaining to the characters A through Z and zero (0) through nine (9).

application (n.)

A program used for a particular kind of work, such as word processing or database management.

area (n.)

In networking, a group of interrelated nodes.

ASCII (n.)

American Standard Code for Information Interchange. A set of 8-bit, binary numbers representing the alphabet, punctuation, numerals, and other symbols used to represent text.

Also, a file that is in binary format. See also *binary*.

asynchronous communication (n.)

The method of transmitting data one character at a time over a serial interface. Asynchronous communication can work locally or through a modem. Timing between bits is constant; timing between characters is variable. (Also called start-stop transmission.)

attached device (n.)

In Token Ring architecture, a processor, printer, or controller that is connected to and can communicate with the Token Ring network.

binary (adj.)

Pertaining to a numbering system that uses a base of 2; it uses only two digits, 1 and 0.

Also, a file type that is in binary format. See also *ASCII*.

client (n.)

A personal computer or workstation, connected to the network with PATHWORKS, that can access resources on a server. A client can have DOS, OS/2, or Macintosh software.

Also, hardware or software that receives resources from a server. See also *server*.

coexistence client (n.)

A personal computer with both Novell NetWare client software and PATHWORKS for DOS client software installed. From a coexistence client, users can access services on both NetWare and PATHWORKS servers. In addition, users have access to local PATHWORKS services such as PC DECwindows, SEDT, Mail, and terminal emulation.

command (n.)

An instruction issued to a computer operating system or application.

command line (n.)

That area of the screen in which commands are entered and displayed.

communication device (n.)

In LAN Manager, a COM or LPT connection on a personal computer. Communication devices include modems, image scanners, and serial printers.

configuration (n.)

The set of hardware, hardware options, and software on a computer or network.

configure (v.)

To select, install, and customize hardware and software for a computer or network.

CPU (n.)

Central processing unit. The main unit of a computer that contains the circuits controlling interpretation and execution of instructions. The CPU includes the main storage, arithmetic unit, and special registers.

CTERM (n.)

Digital Command Terminal. A network protocol that provides local and wide area network services to computers for VT terminal emulation. CTERM is one of the possible protocols used in the SETHOST and VT320 terminal emulator utilities. See also *SETHOST, LAT, NET,* and *Telnet.*

data link (n.)

The interface between the computer's network controller and higher software levels. The type of data link used depends on the computer's network controller.

DCL (n.)

Digital Command Language. The standard command interface to Digital's major operating systems, such as VMS.

DECnet (n.)

Digital networking software that runs on server and client nodes in both local area and wide area networks. With DECnet, different types of computers that have different operating systems can be connected, and users can access information and services on a remote computer.

DECnet is a networking protocol and transport. See also *TCP/IP.*

DECnet link (n.)

A virtual or logical connection between a client and a server or between two nodes in the network.

DECnet node database (n.)

The file that contains information about the network nodes with which a computer communicates.

DELNI (n.)

Digital Ethernet Local Network Interconnect. A local network device that provides eight transceiver cable connections. Up to eight devices can connect to the Ethernet network with a DELNI device.

DELQA (n.)

Digital Ethernet Q-bus Network Adapter. A high-performance IEEE 802.3/Ethernet communications controller for computers based on Q-bus hardware. The DELQA controller allows the MicroVAX computer to connect to the Ethernet network.

DEMPR (n.)

Digital Ethernet Multiport Repeater. A multiport repeater that provides eight ThinWire Ethernet drops (ports for connecting coaxial cable) from a single standard Ethernet connection.

DEPCA (n.)

Digital Ethernet Personal Computer Adapter. An Ethernet controller used to connect personal computers to the network as clients.

DESPR (n.)

Digital Ethernet Single Port Repeater. A repeater device with one port for connecting ThinWire coaxial cable to a standard baseband Ethernet.

DESTA (n.)

Digital Ethernet Station Adapter. An IEEE 802.3/Ethernet-compliant transceiver connecting intelligent devices that use transceiver cable to the ThinWire Ethernet network.

destination (n.)

The drive, file, or media to which a user is copying or moving information. (Also called target.) See also *source*.

device (n.)

A hardware component that performs a specific function. A keyboard is an input device; a printer is an output device; a terminal is an input/output device. See also *logical device*.

end node (n.)

A network node that sends and receives network messages but cannot route packets intended for other nodes.

Ethernet address (n.)

An alphanumeric string, six bytes in length, that identifies a node on the Ethernet. The string is six pairs of hexadecimal digits, separated by hyphens (for example, AA–00–04–00–91–27).

Ethernet controller (n.)

A network controller for the transmission and reception of data between a workstation or server and the Ethernet network. For example, a DEPCA is an Ethernet controller for a personal computer that is connected to the network. See also *network controller.*

FAL (n.)

File Access Listener. A DECnet utility that runs on your personal computer and monitors the network for requests from other users for your files.

group code (n.)

A number or set of numbers used by the LAT or LAST protocol to identify network resources and to control access to those resources. Group codes can be used to assign resources to a specific set of users and to balance the load between computers offering identical services. (Also called group code number.)

H4005 (n.)

A device or transceiver that provides the interface between standard baseband Ethernet and network nodes.

hexadecimal (adj.)

Pertaining to a number system using base 16. Hexadecimal notation includes the numbers 0 through 9 and the letters A through F. In base 10 (decimal notation), F equals 15.

host system (n.)

A computer, such as a server, that provides services to clients.

Internet (n.)

A group of networks that includes regional networks and local networks at universities and commercial institutions. See also *DECnet* and *TCP/IP.*

keyboard mapping (n.)

The definition of keys stored in a file and used by an application. The file associates the key you press with the application's interpretation of that key.

keycode (n.)

The hexadecimal number assigned to a key for identification, regardless of how the key is defined or mapped.

keysym files (n.)

Files containing keyboard mapping information.

LAD (n.)

Local area disk. Digital's virtual disk software on a local area network. LAD provides high-performance disk services to DOS and OS/2 clients connecting to a VMS server. See also *virtual disk*.

LAN (n.)

Local area network. A self-contained network that offers a high-speed, reliable communication channel. LANs span a limited distance, such as a building or cluster of buildings, but can be connected to WANs with bridge devices.

LAST (n.)

Local Area System Transport. The network protocol used by the virtual disk server to send and receive data between computers. LAST provides LAN services to LAD drives.

LAT (n.)

Local Area Transport. A character-oriented transport protocol that operates on a LAN to permit communication between nodes and other devices such as terminals, printers, and modems. See also *LAN* and *SETHOST*.

LATCP (n.)

LAT Control Program. A utility that allows the management of LAT services from the client.

LAT node (n.)

A computer that has LAT software and can offer services, access services, or both. A LAT node can be a personal computer, a terminal server, or a service node. See also *service node* and *terminal server*.

LAT service (n.)

Any service offered on the LAT; a terminal service is the most common type of LAT service.

link (n.)

The logical network connection to a computer or to an application.

load (v.)

To bring software into memory. See also *downline load.*

lobes (n.)

In Token Ring architecture, the cabling between a device and the Token Ring wiring concentrator. See also *Multistation Access Unit.*

local area disk (n.)

See *LAD.*

Local Area Transport (n.)

See *LAT.*

log file (n.)

A text file that contains messages describing events that occur during operation. Log files are updated frequently during operation and are useful for tracing system operation and errors. Log files are created by file servers, X servers, and many applications and utilities.

logical (adj.)

Nonphysical. For example, logical can refer to a name in the software that represents a hardware device. (Also called logical name.) See also *logical device.*

logical device (n.)

A software name that identifies a hardware device for use by an application or program.

log on (v.)

To enter a user name and a password that identify the user and start the session. (Also called log in.)

LPT1, LPT2, LPT3 (n.)

The default logical device names for local parallel printers. LPT1 is the default logical identification for the client local printer port.

MAU (n.)

See *Multistation Access Unit.*

microprocessor (n.)

A CPU contained on a single integrated-circuit microchip.

modem (n.)

Shortened form of Modulator/Demodulator. A device that converts computer signals into signals that can be sent and received over a telephone line.

Multistation Access Unit (MAU) (n.)

In Token Ring architecture, the wiring concentrator that attaches a device to the Token Ring LAN.

NAUN (n.)

See *Nearest Active Upstream Neighbor.*

NCP (n.)

See *Network Control Program.*

NDIS (n.)

The Network Device Interface Specification written jointly by Microsoft Corporation and 3Com Corporation. By supporting NDIS, Digital enables any personal computer vendor's Ethernet controller to work under PATHWORKS for DOS software.

Nearest Active Upstream Neighbor (NAUN) (n.)

In Token Ring architecture, the station on the Token Ring that is just before your station location. This previous station is upstream from the perspective of token flow direction.

network (n.)

A group of servers, clients, and devices that are connected to each other by communications lines to share information and resources.

network adapter (n.)

See *network controller.*

network controller (n.)

A combination of hardware, firmware, and software that controls the transmission and reception of data between a workstation or server and the network. For example, a DEPCA is an Ethernet network controller that connects a personal computer to the network. (Also called network adapter.)

Network Control Program (NCP) (n.)

A DECnet utility used to monitor, manage, and configure network nodes.

Network File Transfer (NFT) (n.)

A DECnet utility used to transfer files on a DECnet network.

network path (n.)

A means of identifying and locating services on the network. A network path consists of the server name and service name.

network topology (n.)

The configuration of wires, cables, and nodes in a network.

NFT (n.)

See *Network File Transfer*.

node (n.)

An individual computer, such as a server or client, that can communicate with other computers in a network.

node address (n.)

A unique numerical identification of a node in a network. A node address includes the area and node number.

node name (n.)

A name uniquely identifying a node within a network. The node name must be alphanumeric and contain at least one alphabetic character.

In DECnet, a valid node name is one to six characters in length. An example of a DECnet node name is SERVR7.

In TCP/IP, a valid node name is one to sixteen characters in length, separated from its domain specification by a period. An example of a valid TCP/IP node name (including a domain specification) is alberteinstein.princeton.edu.

node number (n.)

A number uniquely identifying a specific node in the area.

parallel (adj.)

In data transmissions, pertaining to a method of information transfer in which all bits in a character are transmitted simultaneously, rather than sequentially, on different lines or channels. See also *serial*.

parallel port (n.)

The hardware component used to connect a client to a device that uses parallel data transmission, such as a parallel printer.

parallel printer (n.)

A printer that has a parallel data communications interface. See also *parallel port*.

parameter (n.)

One or more variables that are passed to a program or command before execution. A parameter can be a file specification, option, or device name.

In the following example, filename.txt and LPT1: are parameters of the NET PRINT command:

```
NET PRINT filename.txt LPT1:
```

See also *qualifier*.

password (n.)

A string of characters that uniquely confirms the identity of a user to the system. See also *user name*.

personal computer (n.)

See *client*.

preferred service (n.)

A service you configure in the LAT service table on your node, using LATCP. LATCP lets you select the service announcements you receive from among the many services offered on the network. See also *service announcement*.

printer driver file (n.)

A file used to tailor general printing functions for a specific printer.

printer service (n.)

The availability of a printer that is connected to a server. From the client, users run network commands to access a printer service and print files. A file server makes a printer service available to clients. See also *shared printer*.

print queue (n.)

A list of files waiting to print.

prompt (n.)

A request to the user from the software for information or an input signal.

protocol (n.)

A set of rules that governs the format and timing of messages sent and received over a communication link. For example, DECnet and TCP/IP are network protocols.

protocol stack (n.)

See *protocol.*

qualifier (n.)

A portion of a command that modifies the action by setting or selecting one of several options. For example, in the following command, /COPIES is a qualifier with a value of 3.

```
NET PRINT filename.txt LPT1: /COPIES=3
```

See also *parameter.*

RAM (n.)

Random access memory. Memory from which information can be read and in which new information can be temporarily stored. See also *ROM.*

random access memory (n.)

See *RAM.*

read only memory (n.)

See *ROM.*

read-write access (n.)

The privilege to copy (read) or save to (write) a file, application, or disk area.

remote (adj.)

Stored on or connected to a server or other computer and available to a client over the network only. Remote is the opposite of local. See also *local.*

remote adapter name (n.)

A unique name given to a NETBIOS application on a WAN, or to a computer that is not a workstation on the LAN, such as a VAX file server. See also *adapter name.*

remote printer (n.)

A printer connected to a server on the network. See also *local printer.*

ROM (n.)

Read only memory. Memory from which information can be read, but to which new information cannot be written. See also *RAM.*

router (n.)

A server or a node that can send and receive data packets and direct data packets to other nodes.

SCB (n.)

See *session control block.*

script (n.)

A text file containing a series of commands that are carried out automatically.

segment (n.)

In an Ethernet network, a length of coaxial cable made up of one or more cable sections connected with barrel connectors or T-connectors.

serial (adj.)

In data transmission, pertaining to a method of data transfer in which bits of information are sent one at a time on a single channel. In the PATHWORKS environment, serial transmission is always asynchronous. See also *asynchronous communication* and *parallel.*

serial port (n.)

The hardware component that connects the personal computer or workstation to a serial communication device, such as a modem, terminal, or serial printer. See also *serial.*

serial printer (n.)

A printer that has a serial communications interface. See also *serial* and *serial port.*

server (n.)

A computer running PATHWORKS software that offers file, printer, or disk services to clients. See also *client.*

session (n.)

The logical link between a client or terminal and a server.

session control block (SCB) (n.)

A system data structure that an application must provide to the LAT whenever the application wants to create a session. The session control block contains all the interrupt and exception vectors known to the system.

SETHOST (n.)

A terminal emulator. With SETHOST, clients can establish terminal sessions to host systems and act as terminals connected to a VAX computer.

slot (n.)

A packet of data containing the information for a single session.

special characters (n.)

Any characters other than A through Z, 0 through 9, _, and $. Special characters include characters used in languages other than English, punctuation marks, and some monetary units. (Also called multinational characters.)

synchronous communication (n.)

A method of transmitting data using a timing signal. The timing signal synchronizes the transmitter and the receiver, eliminating the need for stop bits and providing efficiency in data transfer.

T-connector (n.)

Hardware that connects ThinWire Ethernet cable to devices in a network.

TCP/IP (n.)

Transmission Control Protocol/Internet Protocol. A set of protocols that governs the transport of information between computers and networks of dissimilar types. The Internet is a group of networks that includes regional networks and local networks at universities and commercial institutions. TCP/IP is an alternative to the DECnet network transport. See also *DECnet*.

terminal emulator (n.)

A program that lets you use the client as if it were a terminal connected to a host computer. The SETHOST utility is a terminal emulator.

terminate-and-stay-resident (TSR) (n.)

A program that stays in memory, running in the background, even after the user closes the application.

ThinWire (adj.)

Pertaining to a Digital Ethernet coaxial cable that is thin, flexible, and IEEE 802.3/Ethernet compliant. ThinWire is used for connecting LANs.

token passing (n.)

Passing of a free token around a Token Ring LAN until the token is captured by a station wanting to transmit. A token is passed in one direction only, downstream.

Token Ring (n.)

A combination of hardware, software, and firmware that controls the transmission of data between clients and servers on a network. The Digital Token Ring product consists of the DEC TRN Controller (DEQRA); device driver software; and firmware containing diagnostics, installation, and IVP (installation verification procedure).

Token Ring network (n.)

A LAN architecture in which a single control token is passed around the network ring to control communication between the stations. This ensures that only one station is transmitting on the ring at any time. See also *DECnet* and *TCP/IP*.

Transmission Control Protocol/Internet Protocol (n.)

See *TCP/IP*.

transport (n.)

Network software that routes user data to its destination and controls the flow of data.

TSR (n.)

See *terminate-and-stay-resident.*

unshielded twisted-pair cable (n.)

Ordinary phone wire, now used for IEEE 802.3/Ethernet and asynchronous network communications. See also *shielded twisted-pair cable.*

utility (n.)

A general-purpose program included in a system to perform common tasks.

virtual (adj.)

Having the attributes, but not the actual form, of something. For example, a virtual disk is space on a VMS disk that functions as if it were a DOS disk.

VT320 terminal emulator (n.)

In Microsoft Windows, an application that lets you use your computer as if it were a terminal connected to a host computer. See also *terminal emulation*.

WAN (n.)

Wide area network. Two or more standard or extended LANs that are joined by DECnet routers, gateways, or Packet System Interface (PSI) software.

wide area network (n.)

See *WAN*.

workstation (n.)

See *client*.

wrap (v.)

To continue a long text line or command on the next line of the screen.

Index

A

B

C

D

E

F

G

H

I

K

L

M

R

S

T

U

V

W

X

Reader's Comments

PATHWORKS for DOS
SETHOST Terminal Emulation
Guide
AA–PAFCC–TK

Your comments and suggestions help us improve the quality of our publications.

Please rate the manual in the following categories:	Excellent	Good	Fair	Poor
Accuracy (product works as described)	☐	☐	☐	☐
Completeness (enough information)	☐	☐	☐	☐
Clarity (easy to understand)	☐	☐	☐	☐
Organization (structure of subject matter)	☐	☐	☐	☐
Figures (useful)	☐	☐	☐	☐
Examples (useful)	☐	☐	☐	☐
Table of contents (ability to find topic)	☐	☐	☐	☐
Index (ability to find topic)	☐	☐	☐	☐
Page design (overall appearance)	☐	☐	☐	☐
Print quality	☐	☐	☐	☐

What I like best about this manual: _____

_____

What I like least about this manual: _____

_____

Additional comments or suggestions: _____

_____

_____

I found the following errors in this manual:

Page	Description
_____	_____
_____	_____
_____	_____

For which tasks did you use this manual?

☐ Installation
☐ Maintenance
☐ Marketing
☐ Operation/Use
☐ Programming
☐ System Management
☐ Training
☐ Other (please specify) _____

Name/Title _____

Company _____

Address _____

_____

Phone _____ Date _____

Do Not Tear - Fold Here and Tape

digital

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

DIGITAL EQUIPMENT CORPORATION
CORPORATE USER PUBLICATIONS
PKO3–1/D30
129 PARKER STREET
MAYNARD, MA 01754–9975

Do Not Tear - Fold Here and Tape

digital

Order Number: AA-PAFCC-TK
Printed in the U.S.A.