

PCSA

---

**digital**™

VAXmate  
Standalone Guide

AA-LB67A-TH

# PCSA

---



## VAXmate Standalone Guide

**May 1988**

**Revision/Update Information:**

This is a new manual.

**Operating System and Version:**

VAX/VMS, Version 4.6 or later

MS-DOS, Version 3.3

**Software Version:**

VAXmate Standalone, Version 2.0

Order Number AA-LB67A-TH

Digital Equipment Corporation

Maynard, Massachusetts

**First Published, 1988**

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license and may be used or copied only in accordance with the terms of such license.

No responsibility is assumed for the use or reliability of software on equipment that is not supplied by Digital Equipment Corporation or its affiliated companies.

Copyright © 1988 Digital Equipment Corporation.

All Rights Reserved.  
Printed in U.S.A.

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

IBM is a registered trademark of International Business Machines Corporation.

Intel is a trademark of Intel Corporation.

MS is a registered trademark and MS-DOS is a trademark of Microsoft Corporation.

The following are trademarks of Digital Equipment Corporation:

DEC	LA75 Companion	P/OS	UNIBUS
DECmate	LN03	Professional	VAX
DECnet	LN03 Plus	Rainbow	VAXmate
DECUS	LN03R ScriptPrinter	RSTS	VAX/VMS
DECwriter	MASSBUS	RSX	VMS
DIBOL	MicroVAX	RT	VT
LA50	PDP	ThinWire	Work Processor

**digital**™

# Contents

---

<b>Preface</b>	ix
----------------	----

---

## **Part I Getting Started with Your VAXmate Standalone Workstation**

---

### **Chapter 1 The VAXmate Standalone Workstation**

---

What Your VAXmate Standalone System Can Do . . . . .	1-1
List of Hardware and Software Features . . . . .	1-2
MS-DOS Operating System . . . . .	1-3
MS-Windows User Interface . . . . .	1-4
Information System . . . . .	1-5
Before You Continue . . . . .	1-5

### **Chapter 2 Preparing the Hard Disk for Software Installation**

---

Handling Diskettes . . . . .	2-2
Initializing the Hard Disk . . . . .	2-4
Formatting the Hard Disk . . . . .	2-9

## Chapter 3 Installing the VAXmate Software

---

Guidelines for Software Installation . . . . .	3-1
Installing the Software on the Hard Disk . . . . .	3-2

## Chapter 4 Working with Your VAXmate Standalone Workstation

---

What to do Next . . . . .	4-1
Customizing Your Startup Process . . . . .	4-2
International Workstation Setup . . . . .	4-2
Modifying the AUTOEXEC.BAT File . . . . .	4-2
Modifying the CONFIG.SYS File . . . . .	4-4
Printing Files with Your VAXmate . . . . .	4-5
Using Your Workstation as a Terminal . . . . .	4-7

## Chapter 5 Problem Solving

---

Correcting Startup Problems . . . . .	5-1
Screen Remains Blank . . . . .	5-2
Screen Displays a Blinking Cursor . . . . .	5-3
Screen Displays a Number . . . . .	5-3
Preventing Hard Disk Problems . . . . .	5-5
Installation Error Messages . . . . .	5-6

## Appendix A Copying a Diskette

---

## Part II Using Your VAXmate as a Terminal: SETHOST

---

### Chapter 6 Overview of SETHOST

---

What is SETHOST? . . . . .	6-2
Accessing Remote Hosts . . . . .	6-2
Help for SETHOST . . . . .	6-3
Using SETHOST Function Keys . . . . .	6-3
Character Sets . . . . .	6-5
Starting SETHOST . . . . .	6-6
Sample Formats for Starting a Session . . . . .	6-7
Exiting SETHOST . . . . .	6-7

### Chapter 7 Using SETHOST Set-Up

---

Set-Up Configuration Files . . . . .	7-2
Using Non-Default Configuration Files . . . . .	7-2
Using Scripts . . . . .	7-3
Introduction to Set-Up Screens . . . . .	7-3
Saving and Recalling Set-Up Files . . . . .	7-5
Saving Selection Settings . . . . .	7-6
Recalling Selection Settings . . . . .	7-6
Changing Settings . . . . .	7-7
Using SETHOST Keyboard Keys . . . . .	7-8
Exiting Set-Up . . . . .	7-8
Actions Screen . . . . .	7-9
Displaying Status Information . . . . .	7-11
Receiving Characters From the Host Into a File . . . . .	7-12
Autotyping Characters to the Host . . . . .	7-13
Communications Screen . . . . .	7-14
Display Screen . . . . .	7-17
Using Video Modes . . . . .	7-20
Displaying Line Attributes . . . . .	7-21
Displaying Character Attributes . . . . .	7-21

General Screen .....	7-22
Keyboard Screen .....	7-25
Printer Screen .....	7-28
Tabs Screen .....	7-30
Setting Tabs Manually .....	7-31
Telephone Screen .....	7-32
Using Telephone Selections .....	7-33

## Chapter 8 SETHOST Script Processor

---

General Concepts .....	8-1
Starting a Script .....	8-2
Session Log Files .....	8-2
Script Language .....	8-2
Command Line .....	8-5
Strings .....	8-5
Special Characters .....	8-5
Variables .....	8-8
Error Processing .....	8-8
Timeout Handling .....	8-9
Error Handling .....	8-9
Commands .....	8-11
A Sample COM Log-In Script .....	8-84

## Figures

---

1-1 VAXmate Standalone Running Applications .....	1-3
2-1 Parts of a Diskette .....	2-2
2-2 Inserting a Diskette .....	2-3
2-3 Power-up Test Bar .....	2-4
2-4 The Ctrl/Alt/Del Keys .....	2-9
3-1 Hard Disk Directory Structure .....	3-4
4-1 Shift/Prt Sc Keys .....	4-7
5-1 Brightness-Contrast Adjustment .....	5-2

## Tables

---

4-1	AUTOEXEC.BAT Command Lines . . . . .	4-3
4-2	CONFIG.SYS Command Lines . . . . .	4-4
5-1	Powerup Test Numbers and Solutions . . . . .	5-4
5-2	Installation Error Messages . . . . .	5-6
6-1	SETHOST Function Keys . . . . .	6-3
7-1	SETHOST Set-Up Screens . . . . .	7-4
7-2	Actions Set-Up . . . . .	7-9
7-3	Communications Set-Up . . . . .	7-14
7-4	Display Set-Up . . . . .	7-17
7-5	General Set-Up . . . . .	7-22
7-6	Keyboard Set-Up . . . . .	7-25
7-7	Printer Set-Up . . . . .	7-28
7-8	Tab Set-Up . . . . .	7-30
7-9	Telephone Set-Up . . . . .	7-32
8-1	Special Characters Used in Script Commands . . . . .	8-6



## Preface

---

The Personal Computing Systems Architecture (PCSA) is an extension of DIGITAL's systems and networking architecture that merges VMS and MS-DOS environments. One of the workstations that can run in this environment is the VAXmate workstation. A VAXmate workstation can run either by itself (standalone) or connected by asynchronous communications to a host computer. Using the VAXmate as a standalone workstation is the focus of Part I of this guide. Using your workstation as a terminal is the focus of Part II.

### Manual Objectives

The *VAXmate Standalone Guide* is the introductory guide to the VAXmate standalone workstation. It describes the features of your VAXmate standalone workstation, how to install the software, and start using your system.

### Intended Reader

You do not have to be an expert in MS-Windows, MS-DOS, or the VAXmate workstation to use this guide. It will lead you to other books for topics that require more detailed information.

# Manual Organization

Part I, "Getting Started with Your VAXmate Standalone Workstation," contains:

Chapter	Contents
Chapter 1	"The VAXmate Standalone Workstation" tells what the standalone VAXmate workstation can do, introduces the MS-DOS operating system, MS-Windows, and the on-line Information System.
Chapter 2	"Preparing the Hard Disk for Software Installation" contains guidelines on handling diskettes and instructions on running FDISK and FORMAT.
Chapter 3	"Installing the VAXmate Standalone Software" contains installation instructions for the MS-DOS, MS-Windows, and Information System software.
Chapter 4	"Working With Your VAXmate Standalone Workstation" tells what to do after installation, how to customize your workstation start-up process, and gives an overview of printing and terminal emulation.
Chapter 5	"Problem Solving" contains guidance on start-up problems, preventing hard disk problems, and interpreting installation error messages.

Part II, "Using Your VAXmate as a Terminal: SETHOST," provides a guide to one method of terminal emulation in a networked environment. Part 2 contains:

Chapter	Contents
Chapter 1	"Overview of SETHOST" introduces SETHOST.
Chapter 2	"Using SETHOST Set-Up" contains information on SETHOST Set Up screens.
Chapter 3	"SETHOST Script Processor" contains a list of commands for the SETHOST Script utility.

## Conventions Used

Follow these conventions while using this guide and the VAXmate software:

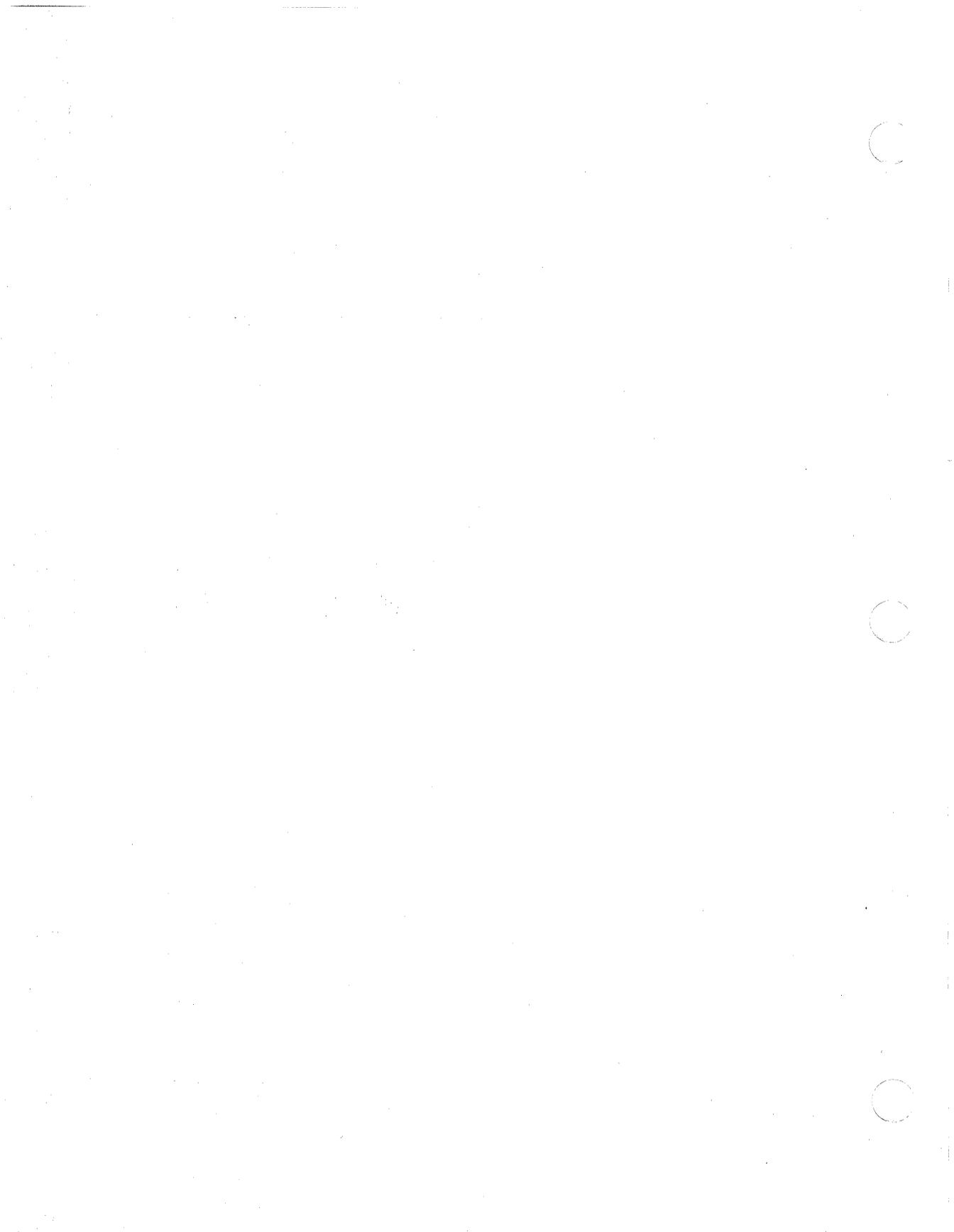
Convention	Meaning
Red type	In examples, what you type from the keyboard is shown in red.
Black type	In examples of computer dialog, what the computer shows on the screen is shown in black.
Typing	Type all spaces and punctuation marks exactly as they are printed. Press the Return key after each command line.
Case	You can type commands and parameters in uppercase or lowercase letters, or in a combination of both.
<b>Return</b>	For user input examples, press the Return key.
Ctrl/Alt/Del	Hold down the Ctrl and Alt keys while you press the Del key on the numeric keypad. The Del is also labelled Sel. You use the Ctrl/Alt/Del keys to restart the workstation.
Shift/Prt Sc	Hold down the Shift key while you press the Prt Sc key.
NOTE	A note contains important general information.
CAUTION	A caution contains information to prevent damage to equipment or software.



## Getting Started

---

Insert tabbed  
divider here.



## Chapter 1

---

# The VAXmate Standalone Workstation

You are the user of a VAXmate workstation, a DIGITAL personal computer that can be used either standalone or connected to a network. The purpose of this guide is to introduce you to the VAXmate standalone environment, help you to install the software, solve problems, and guide you to information about using your workstation.

### What Your VAXmate Standalone System Can Do

This chapter introduces you to the hardware and software features of the VAXmate standalone system. It includes:

- What your standalone VAXmate can do
  - List of Hardware and Software Features
  - MS-DOS Operating System
  - MS-Windows
  - On-line Information System
- Before you continue

## **List of Hardware and Software Features**

The VAXmate hardware, with the addition of the software for standalone use, provides these features:

- 1 Mbyte memory
- Built-in clock/calendar
- Integral graphics monitor with adjustable tilt screen
- 1.2 Mbyte high-density floppy diskette drive
- 20 or 40 Mbytes of hard disk drive storage
- 3 hardware expansion slots
- Extended function keyboard
- Serial Printer Port and Serial Communications Port
- Mouse
- MS-DOS operating system
- MS-Windows user interface
- On-line Information System

The VAXmate system also lets you:

- Use and print graphics characters
- Run several applications at the same time
- Run third-party applications
- Store applications on a hard disk or diskette

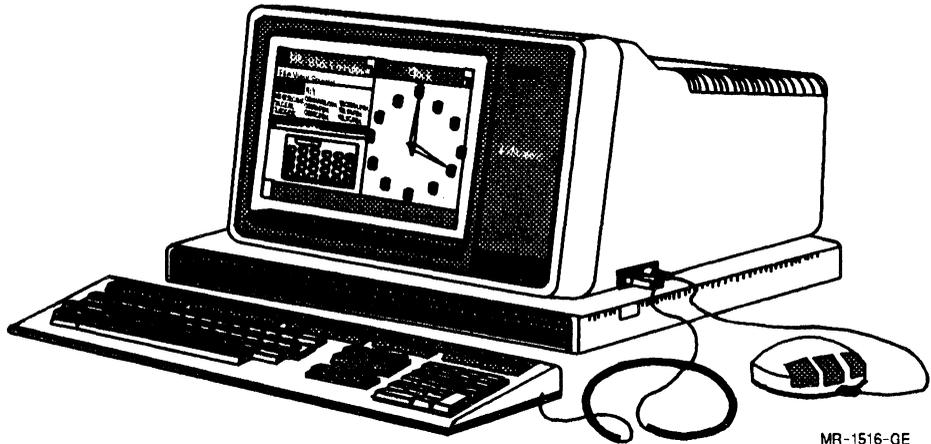
In addition, you can add the following options:

- Integral modem (only available in North America)
- 2 Mbyte memory upgrade option
- Intel 80287 Math Coprocessor

Your VAXmate can also run as a terminal in a DECnet environment. Part II of this guide describes using the SETHOST utility, which lets you use your VAXmate workstation as a terminal emulator. See the *Overview* for an introduction to the network environment.

Figure 1-1 shows the VAXmate standalone workstation running typical MS-Windows applications.

**Figure 1-1: VAXmate Standalone Running Applications**



## MS-DOS Operating System

The MS-DOS operating system provides the command line user interface, utility support, and the ability to run MS-DOS-based applications. You type MS-DOS commands at the prompt or through the MS-DOS Executive Window application to perform tasks such as:

- Creating directories and files
- Copying data
- Developing, running, and testing application programs
- Backing up and restoring information
- Printing text and graphics

To learn more about MS-DOS commands, see the *MS-DOS Reference Manual*.

## **MS-Windows User Interface**

MS-Windows provides a graphical user interface to the operating system in which MS-DOS-based applications can run. You can display several applications at once because MS-Windows provides a separate area of the screen called a *window* for each application. You control Windows applications with the mouse and/or the keyboard. MS-Windows features:

- Pull-down menus - to select commands or options
- Dialog boxes - to facilitate user input
- Icons - to identify, select, and run available applications

Some MS-Window applications, called *desktop applications*, are designed to help simplify your daily work. These applications include:

- Calculator - to perform calculations
- Calendar - to keep track of appointments
- Cardfile - to sort and catalog information
- Clipboard - to hold data for later use
- Clock - to keep time
- Notepad - to write memos, edit files

In addition, the MS-DOS Executive application functions as the file management tool providing an alternative to the traditional MS-DOS command line structure. From it you can start MS-DOS commands and programs using MS-Windows menus and dialog boxes.

To learn how to get started using MS-Windows, see the *VAXmate System Handbook*. For more advanced user information, see the *MS-Windows User's Guide*.

## Information System

Your VAXmate standalone software includes an on-line Information System. This MS-Windows application provides easy-to-use documentation that you can access through menu selections. The Information System contains tutorial and reference information for the VAXmate system. You can read text, print parts of the information for later use, return to specific topics using "bookmarks," and search for topics in the Information System using the on-line index.

Of particular interest to you are the following menu selections:

Type of Topic	Menu Selections
For new users	Getting Started with Your VAXmate Workstation
For general information	Desktop Application, MS-Windows, Standard Applications
For advanced users	Disks and Diskettes, MS-DOS Operating System, MS-DOS Operating System Commands

See your *VAXmate System Handbook* to learn more about the Information System and how to use it.

## Before You Continue

The *VAXmate Expansion Box Installation Guide and Owner's Manual* provides instructions on how to install your expansion box and any hardware options. See that book to make sure your VAXmate hardware is fully installed and all connections are secure.

Next, check to see that your VAXmate standalone software kit includes the following diskettes:

- PCSA VM S/A Install V2.0
- PCSA VM MS-DOS V3.3
- PCSA VM Info. System V2.0
- PCSA VM MS-Windows V1.03 (three diskettes)

You use these diskettes in Chapters 2 and 3. Always keep these diskettes in a safe place as you only have one copy of each. You can make copies of your diskettes. See Appendix A for instructions.



## Chapter 2

---

# Preparing the Hard Disk for Software Installation

This chapter covers:

- Guidelines on handling diskettes
- Instructions on how to prepare the workstation disk drive for software installation

Before you prepare the hard disk, run the extended self-test described in the *VAXmate Expansion Box Installation Guide and Owner's Manual*. Make sure the extended self-test recognizes that the expansion box and hard drive exist. If they are not recognized by this test, and they are there, see the *VAXmate Expansion Box and Installation Guide and Owner's Manual* for help.

### NOTE

You must run the extended self-test every time you connect or reconnect the Expansion Box to the workstation or add options.

When the extended self-test is successful, you are ready to prepare your hard disk by:

1. Running the FDISK utility, which initializes and partitions the drive
2. Running the FORMAT utility with the /S switch, which formats and places part of the MS-DOS operating system on the drive

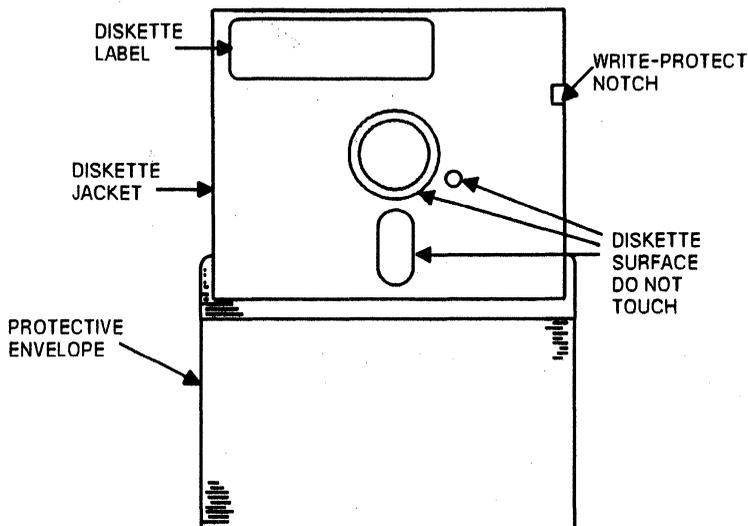
## Handling Diskettes

To run the FDISK and FORMAT utilities, use the MS-DOS diskette that comes with your VAXmate standalone software kit. To prevent damage, follow these simple rules for handling diskettes:

- Store them in the envelope provided.
- Keep the diskettes away from magnets.
- Keep fingers away from the diskette surface area.
- Do not put them on top of your VAXmate or leave them in sunlight or they warp.
- Do not remove a diskette if the drive light is on. The light indicates that the disk head is in contact with the diskette. Removing the diskette when the heads are in contact can corrupt the diskette and possibly ruin the read heads.

Figure 2-1 shows the parts of a diskette.

**Figure 2-1: Parts of a Diskette**



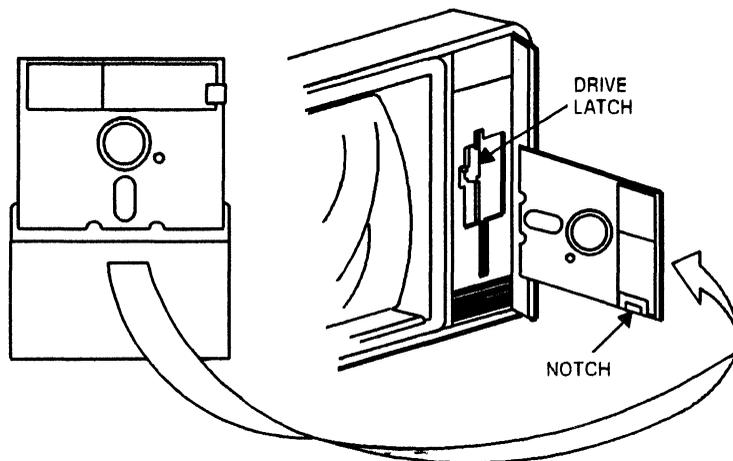
MR-1431-GE

When inserting diskettes in the diskette drive:

- Make sure the write-protect notch is facing down.
- If the notch has a self-adhesive tab covering it, do not remove the tab. If there is no tab, add one if the contents on the diskette cannot be easily replaced.
- Make sure you firmly lock the diskette drive. To lock the drive, turn the latch clockwise.

Figure 2-2 shows how to insert a diskette.

**Figure 2-2: Inserting a Diskette**



MR-1432-GE

## Initializing the Hard Disk

To initialize your hard disk and set up startup partitions, you need to run the FDISK utility.

### CAUTION

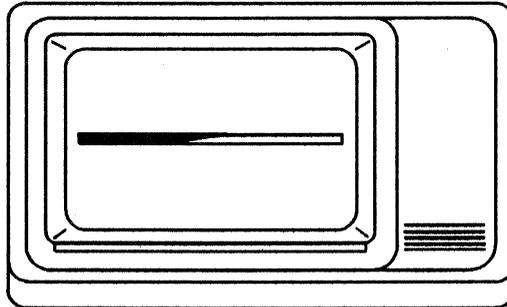
These utilities erase all data from the hard disk. If there is already data on the hard disk, run these utilities after backing up the disk. For more information, see the *VAXmate Standalone Release Notes*. If you are installing the expansion box for the first time, there is no data on the hard disk.

To run FDISK:

1. Remove the diskette labeled PCSA VM MS-DOS V3.3 from its envelope.
2. Insert the diskette into the diskette drive and lock the diskette drive door. (Turn the latch clockwise.)
3. Turn the workstation on.

The system starts a powerup self-test every time you turn it on. The test program draws a bar across the screen and slowly fills this bar during the testing. Figure 2-3 shows the powerup test bar.

**Figure 2-3: Power-up Test Bar**



MR-1433-GE

4. Did the screen display a number or remain blank? Did the system complete the test and then display a blinking cursor? If any of these occurred, the system might have a problem. Go directly to Chapter 5, Problem Solving, for help.

5. Enter the date and time in the format shown in the following example. Press the Return key after each entry.

**NOTE**

You must use the US format here. Later, you can set the workstation to follow country-specific conventions. For more information see the *MS-DOS Reference Manual* and the *International Features Guide*.

Enter the date in the format mm-dd-yy (month, day, year). Enter the time in the format hh-mm-ss (hours, minutes, seconds). The last digit is in 100ths of a second. You need not enter this figure.

For example:

```
Current date is Mon 1-01-2085
Enter new date (mm-dd-yy): 1-01-1988 
Current time is 2:54:12.22
Enter new time: 1:15 
```

**NOTE**

If the expansion box becomes disconnected, always reenter the date and time. Entering the date and time keeps the dates of files and directories you create current.

6. At the A:\> prompt, type:

```
A:\> FDISK 
```

The Hard Disk Setup Program menu displays on your workstation screen.

When the program instructs you to enter a menu selection, use the number keys on the top row of the keyboard.

When you enter a selection and press the Return key, you cannot stop the command execution. The only exception is main menu item 2, **Initialize the drive**, which gives you a chance to change your mind.

**NOTE**

The following example shows the DEC RD31 drive type.

## *Preparing the Hard Disk for Software Installation*

Hard Disk Setup Program Version 2.0  
(C) Copyright 1988 by Digital Equipment Corporation

Drive Type is DEC RD31

Hard Disk Partitions:  
No partitions defined.

Startup priority: floppy disk, hard disk, network.

1. Set drive type.
2. Initialize the drive. CAUTION!  
Erases all data on the disk!
3. Add a partition.
4. Delete a partition.
5. Set the startup partition.
6. Prepare the drive for shipping.
7. Select network startup priority.
8. Exit from this program.

Enter your selection by typing a number followed  
by Return or by pressing the indicated function key.

7. **Select the Set drive type option from the FDISK main menu by typing:**

1

The FDISK utility displays the following menu:

Hard Disk Setup Program Version 2.0  
(C) Copyright 1988 by Digital Equipment Corporation

How do you want to set the drive type?

1. Set type to DEC RD31 (20 MByte).
2. Set type to DEC RD32 (40 Mbyte).
3. Enter a parameter table.
5. Return to main menu without changing type.

Enter your selection:

Type a number to select your drive type, depending on what  
drive type you have.

8. **Select the Initialize the drive option from the FDISK main menu by typing:**

2

### **NOTE**

If you want to stop the initialization process, press  
the Return key to redisplay the FDISK startup menu.

9. Now press F10. The program displays a message similar to the following:

```
Initialization is in progress...
```

```
    Initializing cylinder 001 of 613 cylinders.
```

This process takes a few minutes, then the program returns you to the main menu.

10. Select the **Add a partition** option from the FDISK main menu by typing:

```
3 Return
```

The FDISK utility displays an informational screen.

You can create up to four partitions on a drive. A partition is MS-DOS reserved space, which you indicate as a percentage of the whole drive (100%). MS-DOS allows a disk partition up to 32 Mbytes in size. The partitions can be any percentage (size) value as long as the total of all reserved space does not exceed 100 (or 32 Mbyte size).

The following table gives recommendations for partitions according to drive type.

Drive Type	Mbytes	# Partitions/Partition %
RD31	20 Mbyte	1/100%
RD32	40 Mbyte	2/78%, 21%

For example, if you have an RD32, and two partitions are created to fill the whole disk, the values can be 78% for one partition and 21% for the remaining partition. Note that the total does not exceed 100% and the total does not equal exactly 100% due to mathematical rounding. You do, however, have 100% utilization of the drive.

See the *MS-DOS Reference Manual* for more information on FDISK.

## *Preparing the Hard Disk for Software Installation*

11. Create one partition on the hard disk drive by typing:

### **NOTE**

The following example assumes you have the DEC RD31 drive type.

100

This entry creates one partition, C:, which is 100% of the drive. The program displays a note indicating that you must run **FORMAT** on this new partition before you can use it. (You do not do have to do this yet.)

12. Press any key to continue.
13. Select the **Set the startup partition** option from the FDISK main menu by typing:

5

The FDISK utility displays another menu. The startup partition is the partition from which the MS-DOS operating system is loaded into memory.

14. Select the MS-DOS C partition from this menu as the startup partition by typing:

1

The FDISK utility displays the FDISK main menu and lists partition C: as the startup partition.

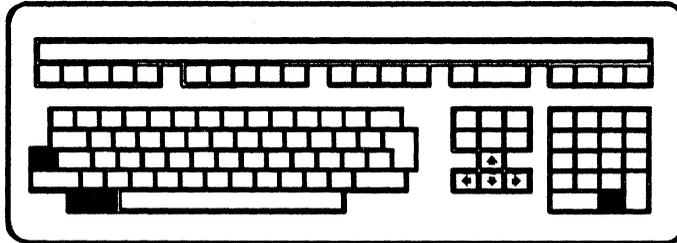
15. Select the **Exit from this program** option from the FDISK main menu by typing:

8

When you exit the program, the FDISK utility displays a message to restart the system by pressing the **Ctrl/Alt/Del** keys.

16. Press the Ctrl/Alt/Del keys. Figure 2-4 shows the location of the Ctrl/Alt/Del keys.

**Figure 2-4: The Ctrl/Alt/Del Keys**



MR-1435-GE

The program also informs you that you must run **FORMAT** on all created partitions.

The workstation displays the startup screen after a few seconds.

17. Enter the time and date again, if incorrect.

## **Formatting the Hard Disk**

After initializing the disk you are ready to format partition C:.

Formatting with the **/S** switch places a part of the MS-DOS operating system on the startup partition (C:). If you later choose to create more than one partition, you must format each partition.

### **NOTE**

The MS-DOS diskette is still in drive A:.

1. Type:

```
A:\>FORMAT C: /S 
```

The **FORMAT** utility displays the following message.

```
CAUTION! FORMAT ERASES ALL DATA ON THE DISK!  
Type Ctrl/C to quit
```

```
CAUTION, ALL DATA ON NONREMOVABLE DISK  
DRIVE C: WILL BE LOST!  
Proceed with Format (Y/N)?
```

## *Preparing the Hard Disk for Software Installation*

### 2. Type:

Y

If you have a used hard disk, the FORMAT utility displays a message indicating that your disk requires a media scan. If this is the case, proceed by typing:

Y

#### **NOTE**

New systems do not display this message. The media scan takes approximately one minute per Mbyte.

### 3. When formatting is completed, the utility indicates how many bytes are available on your hard disk. For example:

```
Format complete
System transferred
```

```
  21274624 bytes total disk space
    75776 bytes used by system
  21198848 bytes available on disk
```

The numbers your VAXmate workstation displays can differ from those shown in the example.

### 4. Remove the MS-DOS diskette from the diskette drive. Do not turn off the workstation, but go to Chapter 3 for software installation instructions.

## Chapter 3

---

# Installing the VAXmate Software

You have run the extended self-test and prepared the hard disk. Now you are ready to install the software.

This chapter covers:

- Guidelines for software installation
- Complete instructions on VAXmate standalone software installation

### Guidelines for Software Installation

To install the MS-DOS operating system, the Information System, and MS-Windows use these diskettes:

- PCSA VM S/A Install Diskette V2.0
- PCSA VM MS-DOS V3.3
- PCSA VM Information System V2.0
- PCSA VM MS-Windows V1.03 (three diskettes)

When installing the software, keep in mind the following:

- It takes some time to install the software on the hard disk.
- You must monitor the installation and follow instructions to insert diskettes.
- Wait for the drive light to go out before you remove one diskette to insert another.

## *Installing the VAXmate Software*

- If you insert the wrong diskette, the screen prompts you to insert the correct diskette. If you do not answer the prompt, the system repeats it until you respond.

Each component is optional, so you can bypass any part of the installation by typing N and pressing the Return key. For new workstations, DIGITAL recommends that you install all the components.

## **Installing the Software on the Hard Disk**

To start the installation process, use the PCSA VM S/A Install diskette V2.0.

1. Insert the diskette in the diskette drive.
2. Press the Ctrl/Alt/Del keys to restart the workstation.

After the power-up test, the system displays the keyboard selection screen, which lists the following choices:

English (US)  
Canadian  
German  
Spanish  
French  
Norwegian  
Swedish  
English (UK)  
Danish  
Finnish  
Italian  
Swiss (German)  
Swiss (French)

3. Using the arrow keys, select the kind of keyboard you want to use and press the Return key. The English (US) selection is the default.

### **NOTE**

This selection is only valid for the installation process. After the installation, the keyboard reverts to the default (US).

A message similar to the following message displays on your screen. The file name in quotation marks varies depending on the keyboard you selected.

Keyboard Map Loader Version 1.03  
(C) Copyright 1988 by Digital Equipment Corporation

The default keyboard map "STDUS.KEY" has been  
successfully loaded.

4. Begin the installation of the software by typing:

A:\> Install

The program displays the message:

VAXmate Software Installation Utility Version 2.0  
(C) Copyright 1988 by Digital Equipment Corporation  
Beginning installation of PCSA VM Operating Environment V2.0  
Drive for software installation [C:]?

5. Press the Return key to indicate that you want the software installed on drive C:.
6. The system asks if you want to install PCSA VM MS-DOS V3.3. Type:

Y

The system prompts you to insert the diskette labeled PCSA VM MS-DOS V3.3.

7. Remove the install diskette and insert the MS-DOS diskette. Press the Return key.

The system automatically copies the files to the hard disk drive. As it copies the software, a message is displayed on the screen indicating the destination directory for the files.

When all the files are copied onto the hard disk, the screen prompts you for the next diskette.

8. The system asks if you want to install the VAXmate Information System. Type:

Y

The system prompts you to insert the diskette labeled VAXmate Info System V2.0. Insert the diskette and press the Return key.

9. The system asks if you want to install MS-Windows. Type:

Y

## Installing the VAXmate Software

The system prompts you to insert the diskette labeled VAXmate Windows V1.03 1 of 3. Insert the diskette and press the Return key. When prompted, follow the instructions to insert diskettes 2 of 3 and 3 of 3.

10. After installing the Windows software, the system prompts you to insert the PCSA VM S/A Install V2.0 diskette. Insert the diskette and press the Return key.

The program copies the AUTOEXEC.BAT, CONFIG.SYS and SETHOST files on the hard disk. For information on these files, see Chapter 4.

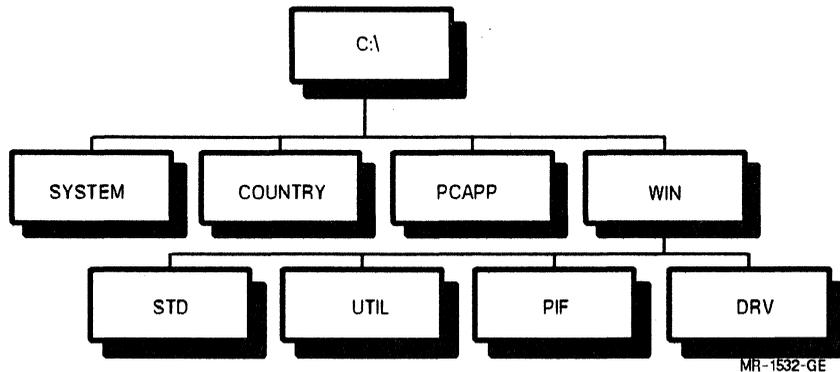
11. When installation is complete, the system displays the message:

Installation of PCSA VM Operating Environment V2.0 completed.

12. Remove the last diskette and store all the diskettes in their covers in a safe place.

During installation, the system automatically sets up directories on the hard disk. Figure 3-1 shows the hard disk directory structure. The C:\ directory is called the root directory.

**Figure 3-1: Hard Disk Directory Structure**



Your VAXmate software is now installed. Read Chapter 4 to learn about working with your VAXmate standalone workstation.

## Chapter 4

---

# Working with Your VAXmate Standalone Workstation

After you have prepared the hard disk and installed the VAXmate software, you can start using your workstation. This chapters covers:

- What to do next
- Customizing your startup process
- Printing files with your VAXmate
- Using your workstation as a terminal

### What to do Next

To start using your workstation:

1. Restart the system by pressing the Ctrl/Alt/Del keys.

#### NOTE

Make sure there is no diskette in drive A.

2. Enter the date and time again, if incorrect.
3. Press any key to start MS-Windows.

MS-Windows starts automatically from the hard disk drive. The system displays the MS-Windows startup window and then the MS-DOS Executive window.

4. To start the on-line Information System, use the mouse to click on the INFO.EXE program file in the PCAPP directory.

You can select topics in the Information System to learn more about MS-Windows and the VAXmate workstation.

If you are using MS-Windows for the first time, see the *VAXmate System Handbook* for information about using the mouse, menu commands, and the Information System.

## Customizing Your Startup Process

The VAXmate standalone software includes an AUTOEXEC.BAT file and a CONFIG.SYS file. Both of these files contain commands that the MS-DOS operating system reads at start-up. This section explains how to change these files to suit your start-up preferences.

## International Workstation Setup

If you want your workstation to start with your preferred country-specific features, you need to modify the AUTOEXEC.BAT and CONFIG.SYS files. To set up your preferred keyboard, character set, and graphics font, add the DECKEYB, FONT, and GRAFTBL commands to the AUTOEXEC.BAT file. To set up your preferred country format, add the LOUNTRY command to the CONFIG.SYS file. See the *International Features Guide* and the *MS-DOS Reference Manual* for more information.

## Modifying the AUTOEXEC.BAT File

When you start the VAXmate standalone workstation, the operating system searches the hard disk for a file named AUTOEXEC.BAT. When the file is found, the operating system runs it automatically. The default file provided with your software kit contains the lines:

```
echo off
prompt $p$q
path c:\;c:\system;c:\country;c:\pcapp;c:\win\std\;
c:\win\util;c:\win\pif;c:\win\drv
graphics /R
echo *****
echo To start PCSA VM MS-Windows
pause
win
```

Table 4-1 explains these command lines.

**Table 4-1: AUTOEXEC.BAT Command Lines**

Command Line	What it Does...
Echo ...	Turns on or off the display of command lines in the AUTOEXEC.BAT file
Prompt ...	Changes the system prompt to your preference
Path ...	Sets the directories the system uses to search for files and commands
Graphics /R ...	Loads the graphics program for printing graphics files from MS-Windows.
Pause ...	Suspends the batch file execution until you press any key
Win ...	Starts MS-Windows

See the *MS-DOS Reference Manual* for more information on these commands.

To customize your workstation start-up you can change or delete the command lines in the AUTOEXEC.BAT file. For example, you can change the operating system prompt, prompt for the date and time, or start the system in MS-DOS instead of in MS-Windows.

To customize your AUTOEXEC.BAT file to start your workstation in MS-DOS, edit the file using the editor or word processor you prefer. The AUTOEXEC.BAT file is located in your root directory (C:!).

The following example uses Notepad. For more information on how to use Notepad, see the *MS-Windows User's Guide*.

1. In the MS-DOS Executive window click on the PCAPP directory.
2. Double click on the Notepad application file (NOTEPAD.EXE)
3. Select the Open command from the pull-down File menu.  
A dialog box is displayed.
4. Click on the File name box and type:

```
C:\> AUTOEXEC.BAT 
```

The AUTOEXEC.BAT file displays in the Notepad window.

5. Edit the file to delete the following lines:

```
echo
pause
win
```

6. Save the file by clicking on the Save command in the File menu.
7. Close Notepad and then Windows.

The next time you start the workstation, it starts in MS-DOS.

For more information on the AUTOEXEC.BAT file, see the *MS-DOS Reference Manual*.

## Modifying the CONFIG.SYS File

The CONFIG.SYS file contains workstation-specific settings, commands that configure your workstation hardware at startup. By changing this file you can configure your workstation with a minimum of effort. The default CONFIG.SYS file provided with your software kit contains the following lines:

```
buffers=30
files=16
lastdrive=N
device=c:\system\mdrive.sys /E*
```

Table 4-2 explains these command lines.

**Table 4-2: CONFIG.SYS Command Lines**

Command Line	What it Does...
Buffers = ...	Sets the maximum number of buffers available for the MS-DOS file system
Files = ...	Sets the maximum number of simultaneously open files that MS-DOS can access
Lastdrive = ...	Sets the last logical drive letter MS-DOS can use
Device = ...	Installs a memory drive in extended memory, if applicable

You can change the maximum number of buffers available for the hard disk with the "buffers = ..." command line. For higher performance with some applications you may want to increase the number of buffers. Check your application software documentation for recommendations.

You can change the maximum number of files. For example, some applications require more than 16 simultaneously open files or the application stops. Check the documentation that comes with the application and edit the "files = ..." line.

The "device = ..." command line sets up memory drives to optimize the memory utilization. If you have installed the 2 Mbyte memory upgrade, for example, this line sets up a memory drive to utilize that extended memory. You can also configure this line to optimize performance in executing applications. If you do not want a memory drive, you can delete the "device = ..." line entirely. For more information on memory drives, see the *MS-DOS Reference Manual*.

MS-DOS has a default list of drive letters. In order to install some software applications, you may need to edit the "lastdrive = ..." line to extend the maximum number of drives available.

To change the CONFIG.SYS file, use Notepad, or any other editor or word processor.

For more information on the CONFIG.SYS file, see the *MS-Windows User's Guide* and the *MS-DOS Reference Manual*.

## **Printing Files with Your VAXmate**

You can use your VAXmate standalone workstation to print files from MS-Windows or the MS-DOS command line to a printer. The following are supported printers:

- LA50
- LA75 Companion
- LN03R Script Printer (MS-Windows only)
- LN03 Plus (with or without ISO/PC cartridge)
- LA210
- LJ250 (MS-DOS only)
- IBM Proprinter
- IBM Graphics Printer

To install your printer, follow the instructions in the printer hardware documentation. Connect the printer to one of the printer ports and start the workstation.

**NOTE**

You can connect the printer cable to the Serial Printer Port (SPP) or Serial Communications Port on your workstation. If you have an adapter, you can use SPP (the default printer port).

The following table contains guidelines to help you configure your printer so that you can print files. For complete information on adding a new printer, or changing your default printer settings, see the *Printing Guide*.

To print from ...	Using printer port ...	You must ...
MS-Windows	SPP	Use the Control Panel application to add printer files.
MS-Windows	Serial Com Port	1. Use the MS-DOS MODE command to redirect printer logical devices to the com port. 2. Use the Control Panel application to add printer files.
MS-DOS	SPP or Serial Com Port	Use the MODE command to set up all printer functions.

For information on the Control Panel and MODE command, see the *Printing Guide*.

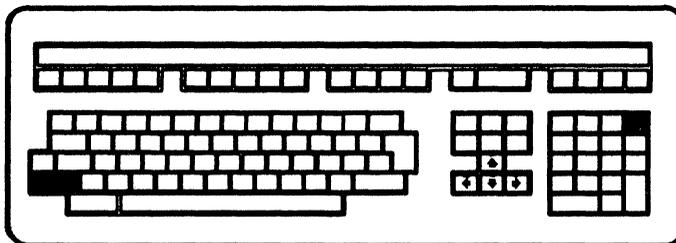
After connecting and configuring your printer, you can use one of the following commands to print files:

- MS-Windows Print command
- MS-DOS PRINT or COPY commands
- An application print command

For information on using the MS-Windows Print command, see the *MS-Windows User's Guide*. For information on the MS-DOS PRINT and COPY commands, see the *MS-DOS Reference Manual*. The *Printing Guide* also has information on these commands. When printing from applications (such as MS-Paint), use that application Print command.

In addition, you can print the contents of a screen by pressing the Shift/Prt Sc keys. Figure 4-1 shows the location of the Shift/Prt Sc keys.

**Figure 4-1: Shift/Prt Sc Keys**



MR-1527-GE

The MS-DOS GRAPHICS command in your AUTOEXEC.BAT file sets up your workstation to print MS-Windows screens. To print screens from MS-DOS, using Shift/Prt Sc, you must edit this file and delete the /R switch from the GRAPHICS command line. For information on the GRAPHICS command, see the *MS-DOS Reference Manual*.

From MS-DOS you can also press Ctrl/P to use line echo. Line echo sends to the printer whatever is displayed on the workstation screen. See the *Printing Guide* for more information on using Ctrl/P.

## Using Your Workstation as a Terminal

A terminal emulator is an application that lets you use your workstation as a terminal connected to a DIGITAL host computer. The terminal emulator makes your workstation keyboard and screen behave like a DIGITAL terminal. You can establish terminal sessions using the serial communications port or the optional integral modem.

## *Working with Your VAXmate Standalone Workstation*

The VAXmate standalone software provides two methods for connecting your workstation to a host computer:

- MS-Windows VT220 terminal emulator application
- SETHOST (VT240 terminal emulator)

During the software installation process described in Chapter 3, the files for these programs (VT220.EXE, SETHOST.EXE, and SETHOST.PIF) were copied to the hard disk.

To learn about the VT220 terminal emulator, see the *MS-Windows User's Guide*. To learn about using SETHOST, see Part II of this guide.

## Chapter 5

---

# Problem Solving

This chapter contains useful guidelines and tables to help you prevent and correct problems. It includes:

- Correcting startup problems
- Preventing hard disk problems
- Interpreting installation error messages

### Correcting Startup Problems

Your workstation can have a problem if one of the following occurs at power-up:

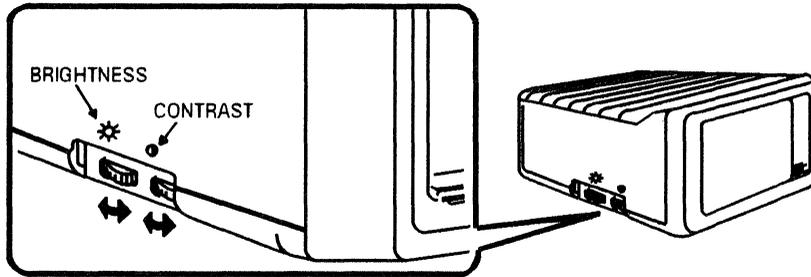
- Screen remains blank
- Screen displays a blinking cursor
- Screen displays a number

**Did the VAXmate workstation display the number 83? The powerup test detected the presence of an unformatted hard disk. Ignore the number and go to Chapter 2, Preparing the Hard Disk.**

## Screen Remains Blank

If the screen remains blank, try adjusting the workstation brightness and contrast controls. Figure 5-1 shows the brightness-contrast adjustment.

**Figure 5-1: Brightness-Contrast Adjustment**



MR-1434-GE

If the screen is still blank, try the following:

1. Unlock the diskette drive and turn the workstation off.
2. Press down on the VAXmate workstation to make sure the workstation is firmly connected to the expansion box.
3. Make sure the power cable is firmly connected. Wait 15 seconds.
4. Lock the diskette drive and turn the workstation on.  
The powerup test runs again.
5. If the screen still remains blank, unlock the drive and remove the diskette. Then, turn off the workstation and call your authorized service representative.

## Screen Displays a Blinking Cursor

The screen can display a blinking cursor for any of the following reasons:

- Diskette drive is not locked.
- Diskette is not inserted incorrectly.
- Wrong diskette is inserted.

Try to correct the problem as follows:

1. Unlock the diskette drive and remove the diskette.
2. Turn the workstation off and wait 15 seconds.
3. Insert the correct diskette for your system in the diskette drive and lock the drive.

### NOTE

Make sure the write-protect notch is facing down and that you lock the drive.

4. Turn the workstation on. The powerup test runs again.

### NOTE

During installation, if the screen displays the system startup screen, go to Chapter 3 and proceed with keyboard selection.

5. If the screen still displays the blinking cursor, unlock the diskette drive and remove the diskette. Then, turn the workstation off and call your authorized service representative for help.

## Screen Displays a Number

During installation, or at other times, the powerup test screen can display a number indicating that there is a problem.

If the system displays a number, try to correct the problem as follows:

1. Remove any diskette and turn off the workstation.
2. Make sure all cables are firmly connected.
3. Make sure the expansion box is firmly connected to the workstation.

## Problem Solving

4. Wait 15 seconds and then insert your diskette (notch facing down).
5. Lock the diskette drive and turn on the workstation.

The powerup test runs again.

### NOTE

During installation, if the startup screen appears, go to Chapter 3 and proceed with keyboard selection.

6. If the fault does not clear, use Table 5-1 for further instructions.

**Table 5-1: Powerup Test Numbers and Solutions**

Number	Probable Cause	Solution
0-40	VAXmate CPU or I/O problem	Call your authorized service representative.
40-46	Unformatted diskette, drive not locked, or diskette inserted wrong.	Use a formatted diskette, insert with notch facing down, and lock the drive. Then, restart the system.
50-52	Memory option board is loose.	Reseat the memory option board, then restart the system.
60	Keyboard cable is loose, or something is pressing on the keyboard during self-test.	Turn off the system. Firmly connect the keyboard cable, then restart the system.
70-82,84-86	Problem might be with hard disk drive or hard disk controller.	Turn off, then turn on the system. If the problem persists, call your authorized service representative.
83,87	Hard disk drive is present but not formatted.	Turn off, then turn on the system. Run the extended powerup test and then the FDISK and FORMAT utilities. See Chapter 2, for information on FDISK and FORMAT.
90-99 and 9A	Modem option board or telephone cables are loose.	Reseat the modem option board. Firmly connect telephone cables, then restart the system.

## Preventing Hard Disk Problems

To help prevent problems or lose of data with your hard disk drive, use the following guidelines:

- Back up all files on the hard disk on a regular basis using the BACKUP command. To learn about the BACKUP command, see the *MS-DOS Reference Manual*.
- Keep a backup copy on diskette of data on the hard disk to recover information in case of accidental loss or hardware failure.
- Back up all files on the hard disk if you want to change partition size.
- Do not switch the system off while you are editing or entering text on the hard disk.
- Help prevent static electricity by using antistatic mats around and under the workstation. Static electricity can add characters or garble the text on your hard disk.
- Keep magnets and demagnetizing equipment away from the disk.

## Installation Error Messages

This section contains a list of error messages you can receive while running FDISK, FORMAT or the installation program. The message is printed first, followed by an explanation and solution. In some cases you must see the *MS-DOS Reference Manual* for more information or call your authorized service representative for assistance.

**Table 5-2: Installation Error Messages**

Message	Explanation	Solution
Bad partition table.	Existing hard disk partition information has become invalid.	Reformat the hard disk and set up a new partition with FDISK. Caution! Data will be lost.
Bad sectors in DOS system area.	Media has a damaged area.	Run FDISK to repartition the disk to avoid the damaged sector. Use less than the percentage requested, for example try 98%, instead of 100%. If you still get the message, call your authorized service representative.
Cannot format an ASSIGNED or SUBSTed drive.	The ASSIGN and SUBST commands mask device letters, so you cannot run the FORMAT utility.	Do not try to run FORMAT. Use the MS-DOS ASSIGN or SUBST commands to unassign or unsubstute the drive. For more information on these commands, see the <i>MS-DOS Reference Manual</i> .
Disk type unknown.	The installed disk type does not match the software setting.	Using FDISK, recheck the setting and set the appropriate value.
Drive not initialized.	The disk is not formatted.	Run the FORMAT utility.
Drive not ready.	The diskette drive door is open or unlocked. The cable is detached or expansion box is detached from the workstation.	Make sure the drive door and drive is locked. Check the cable. Make sure the workstation is properly attached to the expansion box.

**Table 5-2 (Cont.): Installation Error Messages**

<b>Message</b>	<b>Explanation</b>	<b>Solution</b>
Error reading partition table. Error writing partition table.	Partition table is damaged.	Run FDISK and FORMAT again to create a new partition table. Caution! Data will be lost.
Error writing directory.	Fatal error. The root directory could not be created successfully.	Run FORMAT again. Caution! Data will be lost. If you receive the message again, call your authorized service representative.
Error writing FAT.	Fatal FORMAT error. Disk is unusable.	You must run FORMAT again. If you receive the message again, call your authorized service representative.
FORMAT failure.	Fatal error. Disk or diskette is unusable and should be replaced.	If it is a diskette, try running FORMAT again using a new one. If it is the hard disk, run FORMAT again. In both cases if you receive the message again, call your authorized service representative.
Incorrect MS-DOS version.	FORMAT cannot run because of incorrect DOS version.	Supply correct MS-DOS version.
Insufficient memory for system transfer.	This message occurs during installation or running FORMAT, if you do not have enough memory to copy files over.	Restart the workstation. Close any MS-Windows applications that are using up memory.
Invalid characters in volume label.	You tried to enter a volume letter with invalid MS-DOS characters.	See the <i>MS-DOS Reference Manual</i> for correct characters.
Invalid media or track bad. Disk unusable.	Disk or diskette parameters you supplied do not match media or device.	Change to appropriate media or check parameters that you typed in.

**Table 5-2 (Cont.): Installation Error Messages**

<b>Message</b>	<b>Explanation</b>	<b>Solution</b>
Invalid volume ID.	Volume ID is incorrect. FORMAT cannot continue unless it is supplied.	Supply the existing volume ID to FORMAT. Use the LABEL or DIR commands to display the existing label, then type it in.
No room for system on destination disk.	The available space on the startup partition is too limited for MS-DOS system files.	Reformatting is required. Reorganize the disk or change startup partitions.
No drive letter was specified.	FORMAT is missing a drive letter as a parameter.	Reenter FORMAT with a drive letter.
Not ready error reading drive.	The diskette drive may be unlocked, or the diskette is not properly inserted.	Remove the diskette, reinsert it, and lock the drive.
Parameters of hard disk cannot be determined.	The ROM is returning invalid values for the installed disk.	Run FDISK and set the drive type to coordinate drive characters between hardware and software utilities.
Unable to write boot.	FORMAT could not transfer the bootstrap disk block. Media is unusable.	Run FORMAT again. If you receive the message again, call your authorized service representative.

## Appendix A

---

# Copying a Diskette

This appendix gives instructions on how to copy your VM S/A Install diskette. You can use the same procedure to copy any of your VAXmate standalone kit diskettes.

To make a copy of the VM S/A Install V2.0 diskette:

1. Insert the PCSA VM MS-DOS V3.3 diskette into the diskette drive and lock the drive door.
2. Turn on or restart the workstation. To restart the workstation, hold down the Ctrl and Alt keys and press the Del key:

Ctrl/Alt/Del

3. At the system prompt, type:

```
A:\>DISKCOPY A: A:
```

4. When prompted for the source diskette, insert the PCSA VM S/A Install V2.0 diskette into the diskette drive and press the Return key.
5. When prompted for the target diskette, remove the PCSA VM S/A Install V2.0 diskette from the diskette drive and insert the blank diskette into the drive. Press the Return key.

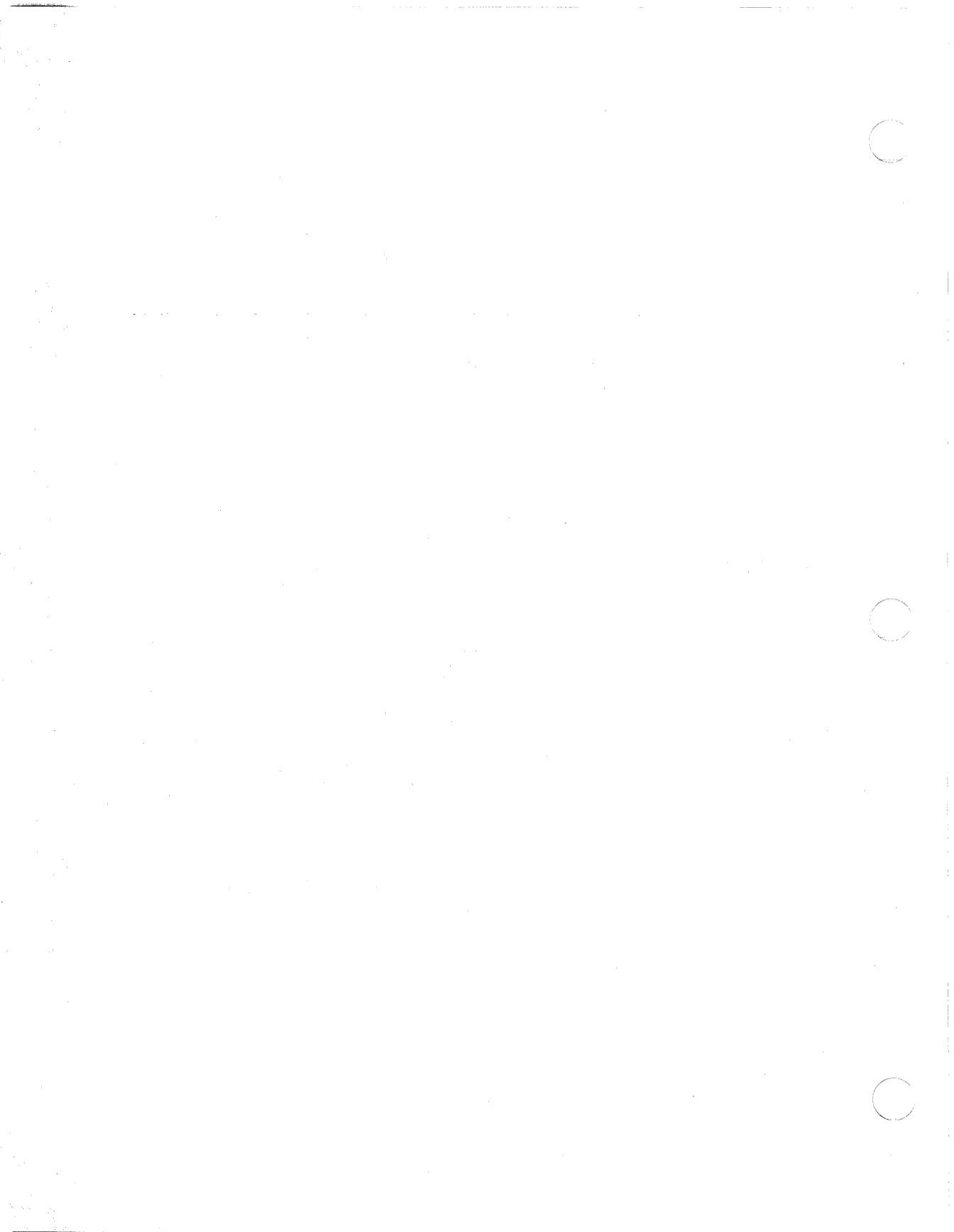
When the copy operation is finished, set aside the original PCSA VM S/A Install V2.0 diskette. Use the copy of the PCSA VM S/A Install V2.0 diskette in the installation procedure described in Chapter 3.



SETHOST: Terminal Emulation

---

Insert tabbed  
divider here.



## Chapter 6

---

# Overview of SETHOST

This part of the *VAXmate Standalone Guide* describes the SETHOST<sup>1</sup> terminal emulator.

This chapter covers:

- What is SETHOST?
- Accessing Remote Hosts
- Help for SETHOST
- Using SETHOST Function Keys
- Character Sets
- Starting SETHOST
- Exiting SETHOST

---

<sup>1</sup> SETHOST was formerly known as the VT240 terminal emulator when used on a VAXmate workstation.

## **What is SETHOST?**

When your personal computer emulates a terminal, you can perform many standard terminal functions of a host computer and gain access to that host's resources as if your personal computer was a terminal directly connected to the host. SETHOST lets your workstation behave like VT52, VT100, or VT200 7- and 8-bit control terminals that support both DEC and ISO Latin-1 character sets. You can connect to a host computer through the asynchronous communications port or through the integral mode option, if installed, to a terminal server.

SETHOST offers you two video modes: Fast Text Only and Text and Graphics. The Text and Graphics mode is used for ReGIS graphics applications, such as DECslide and DECgraph.

The SETHOST emulator allows you to:

- Run host applications (including ReGIS applications)
- Receive or send information to or from a host using files
- Save and recall Set-Up selection settings using Set-Up configuration files

## **Accessing Remote Hosts**

You can gain access to a remote host system by using the asynchronous communication port.

The asynchronous communication port (which can be either COM1 or COM2) provides an asynchronous network connection. You can use this port to connect your terminal line to a terminal server. SETHOST defaults to the COM1 port.

Each time you log onto a host, the connection you make is called a session. The following are the types of sessions you can choose from:

- COM1 port
- COM2 port
- Integral Modem Port

You can define the type of connection you want to access by using the Communications Set-Up screen. For more information on Set-Up screens, see Chapter 7.

## Help for SETHOST

SETHOST provides an on-line help utility when you type SETHOST HELP at your MS-DOS prompt. To view the help text, type the following command:

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

### NOTE

The information on LAT and CTERM do not apply for the standalone workstation. The standalone software does not support these network connections.

## Using SETHOST Function Keys

The top row function keys have fixed meanings for the SETHOST emulator. The labels in parentheses refer to the labels on the standard United States legend strip. Table 6-1 lists these function keys and their descriptions. (This table also includes special keyboard keys that you can use to make selections from each Set-Up screen.)

**Table 6-1: SETHOST Function Keys**

Function Key	Description/Function
<F1> (Hold)	Freezes the screen. None of the characters you type are displayed on the screen until you press <F1> again. Pressing the HOLD key also lights the Hold indicator.  Pressing <Shift/F1> starts an autotyping operation. That is, SETHOST sends the characters in a file as if you typed them on your keyboard.  Pressing <CTRL/F1> starts or stops logging the SETHOST session to a disk file.
<F2> (Print Screen)	Sends the screen text to the printer.  Pressing <CTRL/F2> toggles Auto Print mode. That is, the first time you press it, Auto Print is turned on and SETHOST begins printing the session on your local printer; pressing <CTRL/F2> again turns off Auto Print mode.
<F3> (Set-Up)	Enters (and exits) Set-Up.  Pressing <CTRL/F3> closes session logging.

**Table 6-1 (Cont.): SETHOST Function Keys**

<b>Function Key</b>	<b>Description/Function</b>
<F4> (Data/Talk)	Use this key only if a modem is installed. For more information, see your modem user's guide.
<F5>	Transmits a break (if you enabled break in Set-Up). Pressing <Shift/F5> initiates a communications line disconnect. Pressing <CTRL/F5> sends an answerback message as defined in Set-Up.
<CTRL/F9>	Pressing <CTRL/F9> lets you use MS-DOS. When you type EXIT while using MS-DOS, you are returned to your SETHOST session.
<F10> (Exit)	Pressing this key while holding down the CTRL key (<CTRL/F10>) exits you from the SETHOST emulator. This action also returns you to the MS-Windows screen you were using, if you started SETHOST from MS-Windows.
<F11> (ESC)	Generates an escape character when in VT52 and VT100 modes. In VT200 modes, <F11> sends an escape sequence which indicates to a VAX application that you have pressed the F11 key.
<F12> (BS)	Generates a backspace character when in VT52 and VT100 modes. In VT200 modes, <F12> sends an escape sequence which indicates to a VAX application that you have pressed the F12 key.
<F13> (LF)	Generates a line feed character when in VT52 and VT100 modes. In VT200 modes, <F13> sends an escape sequence which indicates to a VAX application that you have pressed the F12 key.
COMPOSE	Initiates a COMPOSE sequence.
Next	Displays the next Set-Up screen.
Prev	Displays the previous Set-Up screen.
<Right Arrow>	Allows you to make a selection from either side of the Set-Up screen (depending on which column you are using and which item is highlighted).
<Left Arrow>	
<End>	
Shift/Left Arrow	Flips the screen to the left.

**Table 6-1 (Cont.): SETHOST Function Keys**

Function Key	Description/Function
Shift/Right Arrow	Flips the screen to the right.
<Up Arrow> or <Down Arrow>	Pressing the Up arrow or Down arrow keys moves you either up or down the list of selections in either column.

## Character Sets

To configure SETHOST for use with international keyboards, follow these steps:

1. Follow the instructions that came with your VAXmate for selecting an international keyboard and character set. For example, to use the DEC Multinational character set, you need to use the DECKEYB file MCSUS.KEY.

To use a 7-bit National Replacement Character Set, use KEYB to select the appropriate .KEY file. SETHOST will automatically use that character set. For 8-bit character sets, you have a choice between DEC Multinational and ISO Latin-1 character sets.

2. Tell SETHOST which character set you wish to use. (Follow Steps 3 through 7.)
3. Press <F3> to get into the Set-Up menu.
4. Use <Next> or <Previous> to get to the General menu.
5. Use <Up Arrow> or <Down Arrow> to move to the character set selection option.
6. Select the character set you want to use.
7. Save this as the new default by using the Save Set-Up Parameters option on the Action menu.

For more information on character sets, see the *International Features Guide*.

## Starting SETHOST

Before you can use the Set-Up screens to create configuration files, you must first start a SETHOST session. To start a session, type SETHOST at the system prompt. For example:

```
C:>SETHOST <RET>
```

The system displays the following information:

```
SETHOST Version 2.0  
Copyright (c) 1987, 1988 by Digital Equipment Corporation  
Press Shift Prev or Shift F9 for Previous Session  
Press Shift Next or Shift F10 for Next Session  
Press Ctrl F9 to hotkey to MS-DOS, Press Ctrl F10 to exit  
Press Set-Up (F3) to enter Set-Up
```

Once you see this information on the screen, you can press F3 to enter Set-Up and use the Set-Up screens.

### NOTE

You can also start SETHOST from a window, if you are using MS-Windows. However, SETHOST is only a Class C windows application. This means that once you start SETHOST, it takes over your screen and you can no longer use MS-Windows. To issue SETHOST commands, you must use the keyboard keys.

## **Sample Formats for Starting a Session**

The following examples illustrate different formats of the SETHOST command that you can use to start a SETHOST session.

1. SETHOST

This format uses all of the SETHOST defaults, since nothing else is specified.

2. SETHOST COM1:

This format starts a SETHOST session using the COM1 port.

3. SETHOST COM2:

This format starts a SETHOST session using the COM2 port.

## **Exiting SETHOST**

To exit from SETHOST, press <CTRL/F10>. SETHOST exits and returns you to the prompt or window that you were using previously.



## Chapter 7

---

# Using SETHOST Set-Up

The SETHOST terminal emulator offers a Set-Up utility featuring many parameters or selections that you can change or set depending on your needs.

This chapter covers:

- Set-Up configuration files
- Introduction to Set-Up screens
- Saving and recalling Set-Up files
- Changing settings
- Using SETHOST keyboard keys
- Exiting Set-Up
- Actions screen
- Communications screen
- Display screen
- General screen
- Keyboard screen
- Printer screen
- Tabs screen
- Telephone screen

## Set-Up Configuration Files

The SETHOST emulator allows you to save Set-Up parameters in a Set-Up configuration file. SETHOST reads this file whenever you start the program. By default, the file is named SETHOST.DAT and is stored in the SYSTEM directory. You can also have several different configuration files, each of which can specify different settings. To use a different configuration file, you must include the /S= switch and the file name on the command line when you first start SETHOST.

When you start the SETHOST emulator, it looks for the default Set-Up configuration file (SETHOST.DAT) in the SYSTEM directory. If you specify a different configuration file name, SETHOST looks for that file in your current directory. If it does not find the file in the current directory, SETHOST continues to look in the other directories specified in the path list of directories. Once SETHOST locates the file, it uses the values in the file to configure the emulator with the initial values of all the Set-Up selections.

You can also direct SETHOST to run a different Set-Up configuration file when you start it. If SETHOST does not find any configuration files, then it configures the emulator using the factory default settings. (The default settings are to use the COM1 port at 9600 baud.)

In order to use Set-Up screens to create configuration files, you must first start SETHOST. Once you start the session, you can press <F3> to enter Set-Up and use the Set-Up screens.

## Using Non-Default Configuration Files

You can specify your own configuration file to use when you start the SETHOST emulator, instead of using the SETHOST.DAT file. Once you have created your configuration file, you can use a switch on the SETHOST command line that will force SETHOST to use your file instead of the default.

The command line syntax looks like this:

```
SETHOST/S=startup.ext
```

The *startup* variable is the name of the file you have created. The *ext* is the file type. You can assign any three-character string as the file type.

As an example, if you have a configuration file named MYSETUP.FIL, you would enter the name with the SETHOST command as follows:

```
C:\>SETHOST/S=MYSETUP.FIL 
```

This command would cause SETHOST to start using the settings in MYSETUP.FIL.

## Using Scripts

The SETHOST utility provides a script processor that lets you use scripts for executing commands. A script is a text file that contains the commands you want to use to perform a function, such as automatically logging on to a host system. The scripts automatically execute each command to simulate 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 a Set-Up file, but the script can include commands and a Set-Up file only includes parameters to use for configuring your terminal emulator session.

The command line syntax for using a script file looks like this:

```
SETHOST/SCRIPT=login.scr
```

The *login* variable is the name of the script file you have created. The *scr* variable is the file type.

For more detailed information on creating scripts and using script commands, see Chapter 8.

## Introduction to Set-Up Screens

The SETHOST emulator provides eight different Set-Up screens that offer various selections for changing the following information:

- Terminal characteristics
- Screen displays
- Communications environment
- Keyboard operations
- Printer characteristics
- Tab settings
- Telephone answering characteristics
- General

## Using SETHOST Set-Up

Each Set-Up screen is divided into three sections:

1. A top section indicating the available Set-Up screens.
2. A middle section reflecting information about the Set-Up screen you are currently using.
3. A bottom section describing the Set-Up keys you can use to change or select information.

Table 7-1 describes each of the Set-Up screens.

**Table 7-1: SETHOST Set-Up Screens**

<b>Set-Up Screen</b>	<b>Description</b>
Actions	Contains selections relating to resetting the terminal, clearing the screen, clearing communications, saving and recalling Set-Up settings, setting default parameters, and showing status.
Communications	Contains selections for defining the communications environment, including the communications port and disconnect delays, speed, parity, stop bits, XOFF information, and transmit/receive information.
Display	Contains selections for defining screen display, such as columns, auto wrap, background, and cursor style.
General	Contains commonly used general operating selections, such as local echo, character sets, and terminal type (VT200, VT100, VT52).
Keyboard	Contains selections for defining keyboard operating characteristics, such as keypad, cursor keys, margin bell, answerback, and keyclick.
Printer	Contains selections for defining printer operations, such as print size, type of printer, and print terminator.
Tabs	Contains selections for defining the tab settings.

**Table 7-1 (Cont.): SETHOST Set-Up Screens**

Set-Up Screen	Description
Telephone	Lets you determine how telephone answering is done. You can also enter the telephone numbers you want to use with the modem.

From each Set-Up screen you can do the following:

- View the current selections and settings
- Change the current settings to suit your needs
- Use the Next Screen or Previous Screen keys to access any of the other Set-Up screens

All settings can be saved or recalled from Set-Up files.

## **Saving and Recalling Set-Up Files**

After you select the desired Set-Up settings, you can save and recall them from the default file, SETHOST.DAT, or a file you specify.

When you start the SETHOST emulator, it uses the values contained in SETHOST.DAT or the specified file to initialize settings.

The SETHOST emulator also lets you create files in which you save Set-Up settings. After making the desired Set-Up changes, you select the Save Set-Up Parameters selection from the Actions screen. You can then create Set-Up files with settings tailored to your own preferences or for use with different applications.

You can select any of these files while running the SETHOST emulator by choosing the Recall Set-Up Parameters selection from the Actions screen.

## Saving Selection Settings

To save all your SETHOST Set-Up selections, follow these steps:

1. Select the Actions screen.
2. Select the Save Set-Up Parameters option.
3. Press the Select key to make a selection from the right column.  
Either the default file name SETHOST.DAT or the name of the most recently created save file is displayed in the right column.
4. Follow one of these steps to save your selections:
  - Select the SETHOST.DAT file.
  - Select the currently displayed file.
  - Type in a new file name.

### NOTE

If you select a currently displayed file, the new values you set replace the existing values in that file.

- Press the Return key to save your selections in the new or selected file.

## Recalling Selection Settings

To recall SETHOST selection settings under Set-Up, follow these steps:

1. Select the Actions screen.
2. Select the Recall Set-Up Parameters option.
3. Press the Select key to make a selection from the right column.  
Either the default file name SETHOST.DAT or the name of the most recently created save file is displayed in the right column.
4. Follow one of these steps to recall your selections:
  - Select the SETHOST.DAT file.
  - Select the currently displayed file.
  - Type in a file name to contain Set-Up values.
5. Press the Return key to recall your saved Set-Up selections from the selected file.

## Changing Settings

To choose a Set-Up screen and change a setting, follow these steps:

1. Select the desired Set-Up screen by pressing the Next or Prev key.
2. After the desired screen is displayed, press the up or down arrow key to view different selections.

The current settings are displayed in the left column. The setting selections are displayed in the right column.

### NOTE

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

3. Press the Select key to move to the right column and select a setting.

The selection in the left column is now bolded to indicate which selection you are setting.

4. Use the up or down arrow key to highlight the desired setting.
5. Press the Select key to enter this setting.

The new setting is now highlighted in the left column.

### NOTE

Some settings take effect immediately; others take effect when you exit Set-Up.

## **Using SETHOST Keyboard Keys**

You can make selections while using SETHOST with the keyboard keys. (If you started SETHOST from an MS-Windows environment, you can use only the keyboard for making selections. You can no longer use the mouse.)

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

## **Exiting Set-Up**

To exit Set-Up, press the F3 (Set-Up) key again.

The following sections describe each Set-Up screen and the options that are available for each screen.

## Actions Screen

After you enter Set-Up, the first screen that SETHOST displays is the Actions screen. This screen provides options that allow you to clear your screen, adjust or reset the terminal emulator, clear communications information, save or recall the Set-Up settings, and display the status of your terminal.

Table 7-2 defines the Actions Set-Up selections.

**Table 7-2: Actions Set-Up**

Selections	Function
Reset Terminal	Resets many operating settings to power up default state. Exits ReGIS. Does not affect communications, NRC (National Replacement Characters) and multinational modes, or user-defined keys.
Clear Display	Clears the screen and sends the cursor to home position when you leave Set-up.
Clear Communications	Clears communication lines immediately and stops any print operation in progress. It also ends printer controller mode.
Recall Set-Up Parameters	Replaces all existing Set-Up settings with values you saved in a default file called SETHOST.DAT, or in a file you specify. When you select this option, the default file name appears in the right column. If you want to change the file name, press the Select key. You can now edit the current file name or enter a new file name for your Set-Up values. Press the Return key when you are done.
Save Set-Up Parameters	Saves the Set-Up settings in a default file called SETHOST.DAT, or in a file you specify. You can also change the current file name by pressing the Select key and entering a new file name. Press the Return key when you are done.
Set Default Set-Up Parameters	Replaces all current Set-Up settings with factory default settings.

**Table 7-2 (Cont.): Actions Set-Up**

<b>Selections</b>	<b>Function</b>
Show Status	Displays information on the state of such items as modems, file sending or receiving, and communications ports.
To MS-DOS	Lets you temporarily leave SETHOST while you use the MS-DOS operating system. When you finish using MS-DOS commands, you can return to your session by typing EXIT at the MS-DOS prompt.  If there is not enough memory to run COMMAND.COM, a message is displayed. If there is enough memory to run COMMAND.COM but not enough memory to use other MS-DOS commands, you are returned to the emulator Set-Up.
Exit SETHOST	Allows you to exit the emulator and return to the prompt you were using previously. If you were using MS-Windows, you return to the window you were using previously.

## Displaying Status Information

Status information is available only from the SETHOST Actions screen. It is a reporting area only. When you select this option, SETHOST displays the following information:

### **Receiving or sending a file**

Idle or In Progress

### **Printing**

Normal Print Mode, Auto Print Mode, Printer Controller Mode

### **Modem usage**

Valid only if the optional integral modem is installed, and you selected the integral modem.

DSR or No DSR

### **Telephone modes**

Valid only if the optional integral modem is installed, and you selected the integral modem.

Manual or Automatic Answer

### **Insert/Replace mode**

[no screen display]

### **Keyboard**

Keyboard file currently in use

### **Character Set**

Character sets currently in use

### **Communications**

Com Port 1, Com Port 2, or Network Terminal Service

## Receiving Characters From the Host Into a File

Show Status in the Actions screen indicates whether session logging is active or idle.

To receive or log characters from the host into a file using session logging, follow these steps:

1. Press <CTRL/F1>. *SETHOST* displays this prompt on the bottom line:

**Receive filename:**

2. Type in a file name for storing characters that will 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.
3. Press <CTRL/F1> to stop characters from logging into the file.
  4. Press <CTRL/F3> to close the Receive file.

## Autotyping Characters to the Host

Show Status in the Actions screen indicates whether autotyping is active or idle.

When a file is autotyped to VMS, you must first enable HOSTSYNC under VMS. For example:

```
$ SET TERMINAL/HOSTSYNC
```

This prevents data overruns on a VMS host.

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

1. Press <Shift/F1>. SETHOST displays this prompt on the bottom line:

**Send filename:**

2. Type in the name of the file that will be autotyped to the host.

File autotyping stops when the end of the file is reached.

You can also press <Shift/F1> again to stop autotyping the file.

## Communications Screen

The Communications screen contains selections for defining the communications environment (including the communications port and disconnect delays), speed, parity, stop bits, XOFF information, and transmit/receive information.

Table 7-3 defines the Communications Set-Up selections and their available options.

**Table 7-3: Communications Set-Up**

<b>Current Settings</b>	<b>Description and Available Selections</b>
Com Port 1: Data Leads Only	<p>Provides a selection of communication port settings (Com Port 1 or Com Port 2) and network communications. The following selections are available for the communications ports:</p> <ul style="list-style-type: none"><li>Com Port 1: Data Leads Only (default)</li><li>Com Port 1: Full Modem Control</li><li>Com Port 2: Data Leads Only</li><li>Com Port 2: Full Modem Control</li><li>Com Port 2: Integral Modem Port</li><li>Network Communications Port</li></ul> <p><b>Note:</b> This selection is not applicable in the VAXmate standalone environment.</p>
Disconnect Delay	<p>If you are using modem control, this option selects the amount of time before the workstation disconnects from the communications line if the carrier detect is lost. The following selections are available:</p> <ul style="list-style-type: none"><li>Disconnect, 2-second delay (default)</li><li>Disconnect, 60 ms delay</li></ul> <p>Most countries except the United Kingdom use the 2-second delay; the United Kingdom uses the 60-millisecond delay.</p>
XOFF	<p>Selects the XOFF point or disables the automatic XON/XOFF flow control. For most applications you should set XOFF at 64 or 256. If you set XOFF greater than 256 and have a buffer overflow problem, set XOFF to the next lower value.</p>

**Table 7-3 (Cont.): Communications Set-Up**

<b>Current Settings</b>	<b>Description and Available Selections</b>		
	The XOFF point can have one of the following values: 64 (default) 256 512 1024 No XOFF		
	If you select No XOFF, the characters are received continuously from the host and some of those characters may be lost.		
<b>Speed</b>	Selects the rate at which characters are received or transmitted. The default is 9600 baud. The speed can have one of the following values:		
	50	300	2400
	75	600	3600
	110	1200	4800
	134.5	1800	9600 (default)
	150	2000	19200
<b>Data Bits and Parity</b>	Selects the character format used for communication with a host computer. The following selections are available: 8 bits - No Parity (default) 8 bits - Even Parity 8 bits - Even Parity - No Check		

**Table 7-3 (Cont.): Communications Set-Up**

<b>Current Settings</b>	<b>Description and Available Selections</b>
	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
<b>Stop Bits</b>	Selects the number of stop bits (1 or 2) used for communicating with the host. The default is 1 stop bit.
<b>Transmit = Receive</b>	Characters can be received and transmitted at the same rate, or at differing transmit and receive rates. These settings are usually equal. As a default, the transmit rate is set equal to the receive rate. You can also set the transmit rate to 1200.

## Display Screen

The Display screen contains selections for defining screen display characteristics, such as number of columns, auto wrap, background, and cursor style.

Table 7-4 defines the Display Set-Up selections and their available options.

**Table 7-4: Display Set-Up**

Current Setting	Description and Available Selections
Columns	<p>Selects an 80- or 132-column screen for text. A change to this setting takes effect on exiting Set-Up. The default setting is 80-column screen.</p> <p>If you select 132 columns, you should note that fast text can display 80 columns at a time. This means the screen must be panned to display the rest of the data.</p> <p>In Graphics Mode the font size changes so that all 132 columns can be seen.</p>
Auto Wrap	<p>Selects whether text automatically wraps on the screen. The following selections are available:</p> <p>Auto Wrap On - Causes characters that reach the right margin to automatically be displayed in the first character position of the next line.</p> <p>Auto Wrap Off - Causes characters that go beyond the right margin to overwrite the last character position of the current line. This is the default.</p>
Background	<p>Selects either a dark background or a light background as the screen display type. The default is Dark Background.</p>
Cursor Style	<p>Selects either a block cursor or an underline cursor as the style of cursor. The default is Block Cursor.</p>
Cursor Type	<p>Selects either a visible cursor or an invisible cursor. The default is Visible Cursor.</p>
Monochrome	<p>Selects the type of monitor you are using.</p> <p>The choices are either Monochrome or Color. If you select Color, you can determine the degree of intensity for character display. Black would be the darkest and intense white the lightest.</p>

**Table 7-4 (Cont.): Display Set-Up**

<b>Current Setting</b>	<b>Description and Available Selections</b>
M0 Background Color	Lets you determine the setting for background color.
M1 Reverse Video	Lets you determine the setting for reverse video. The same color choices are available for this setting as for M0.
M2 Bold Text	Lets you determine the setting for bold text. The same color choices are available as for M0. (If the setting is Monochrome, this is the shade of the text.)
M3 Foreground Color	Lets you determine the setting for foreground color. The same color choices are available as for M0. (If the setting is Monochrome, this is the shade of the bold text.)
Video Mode	Determines how characters are displayed on your screen. Two selections are available: Fast Text Only Video Mode Text and Graphics Video Mode Fast Text Only selects an 80 X 25 text mode. This is the default. Text and Graphics selects an 800 X 250 ReGIS graphics mode. If you choose this selection, the following options are available: Graphics Cursor Macro Report Print Background Print Image

**Table 7-4 (Cont.): Display Set-Up**

<b>Current Setting</b>	<b>Description and Available Selections</b>
	For the Graphics Cursor, you can choose either Visible or Invisible.
	For Macro Report, you can choose to send the contents of a macro report in response to the ReGIS Report Macrograph command R(M...), or you can choose not to send the contents of the report.
	For Print Background, you can choose to print or not print the background color. This option has no effect on monochrome print mode.
	For Print Image, you can choose one of the following:
	A compressed print image of 6 X 3 inches
	An expanded print image of 12 X 8 inches
	A rotated image (an image printed in 8 X 12 inches and rotated 90 degrees)

## **Using Video Modes**

The Fast Text Only mode operates faster than the ReGIS Text and Graphics mode. However, each mode displays the following information differently:

- ReGIS graphics
- 132 columns
- Line attributes

When operating SETHOST in Text and Graphics mode, 132-column displays appear exactly as they would on a VT200-type terminal.

When operating SETHOST in Fast Text Only mode, all 132 columns cannot be viewed at the same time. Only the first 80 of the 132 columns appear on your screen.

To view the remaining columns, use one of these key combinations:

**<Shift/Right/Arrow>**

Displays columns 53-132

**<Shift/Left Arrow>**

Displays columns 1-80

## Displaying Line Attributes

When using SETHOST in Text and Graphics mode, all line attributes appear as they would on a VT200-type terminal.

When using SETHOST in Fast Text Only mode, some line attributes are displayed differently for double width and double height/double width.

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
```

Since double height takes up two lines, the text on the first line is followed by a blank line on the second line.

## Displaying Character Attributes

All of the character attributes for a VT200-type terminal are supported, with this exception:

When operating in Fast Text Only mode, characters intended to be underscored will appear without the underscoring.

## General Screen

The General screen contains commonly used general operating selections, such as local echo, character sets, and terminal type (VT200, VT100, or VT52).

Table 7-5 defines the General Set-Up selections and their available options.

**Table 7-5: General Set-Up**

<b>Current Setting</b>	<b>Description and Available Selections</b>
Line Mode	<p>Lets you select the mode of operation. You can select either On-line or Local.</p> <p>On-line allows your workstation to communicate with a host computer. This is the default.</p> <p>Local stops communication between your workstation and the host. Data entered at the keyboard goes directly to the workstation screen only.</p>
Local Echo	<p>Enables or disables the local echo setting. You can select either Local Echo Off or Local Echo On.</p> <p>Local Echo Off directs characters from the keyboard to a host computer only. The host in turn may or may not send the characters back to the screen. This is the default.</p> <p>Local Echo On directs characters from the keyboard to the screen as well as to the host. Use this option if the host does not send characters back to the screen.</p>
New Line	<p>Selects whether the Return key generates a carriage return only or a combination carriage return and a line feed. You can select one of the following:</p> <p>No New Line generates a carriage return only. Received carriage returns do not cause a new line operation.</p> <p>New Line generates a carriage return and a line feed. Received carriage returns cause a new line operation.</p> <p><b>Note:</b> When SETHOST is in numeric keypad mode, this setting affects the ENTER key in the same way it does the Return key.</p>

**Table 7-5 (Cont.): General Set-Up**

Current Setting	Description and Available Selections
Character Mode	<p>Determines the category of character set mappings to use for your terminal. You can choose either the National Replacement Character (NRC) mode or the Multinational mode.</p> <p>If you choose Multinational mode, you then choose one of the two available multinational character sets. NRC mode allows you to choose one of the National Replacement Character Sets.</p>
Character Set	<p>Lets you choose the specific character set.</p> <p><b>Note:</b> This setting is not applicable for the VAXmate standalone workstation. Use the FONT and DECKEYB commands. For more information on these commands, see the <i>International Features Guide</i> and the <i>MS-DOS Reference Guide</i>.</p>
Terminal Emulation	<p>Lets you identify the type of terminal you want to emulate. The following selections are available:</p> <p>VT200 with 8-bit controls</p> <p>VT200 with 7-bit controls (default)</p> <p>VT100</p> <p>VT52</p>
User-defined Keys	<p>Lets you determine whether a host can change user-defined key (UDK) definitions. The following selections are available:</p> <p>User-defined Keys Locked</p> <p>User-defined Keys Unlocked</p> <p>When you select Locked, the UDKs cannot be changed by the host.</p> <p>When you select Unlocked, the UDKs can be changed by the host. The default is Unlocked.</p>
User Features	<p>Lets you determine whether a host can change certain user features. The following selections are available:</p> <p>User Features Locked</p> <p>User Features Unlocked</p>

**Table 7-5 (Cont.): General Set-Up**

<b>Current Setting</b>	<b>Description and Available Selections</b>
	<p>If you select <b>Locked</b>, certain features cannot be changed. If you select <b>Unlocked</b>, certain user features can be changed. The default is <b>Unlocked</b>.</p> <p>The user that can be locked or unlocked include the following:</p> <ul style="list-style-type: none"><li>Auto Repeat</li><li>Light/Dark Screen</li><li>Tab Stops</li><li>Keyboard Lock</li></ul> <p><b>Note:</b> Some software applications expect to control these user features. If this applies to your particular software application, select <b>User Features Unlocked</b>.</p>

## Keyboard Screen

The Keyboard screen contains selections for defining keyboard operating characteristics, such as keypad keys, cursor keys, margin bell, answerback, and keyclick.

Table 7-6 defines the Keyboard Set-Up selections and their available options.

**Table 7-6: Keyboard Set-Up**

Current Setting	Description and Available Selections
Keypad Mode	<p>Sets the terminal keypad for using either numbers or control codes. Two selections are available:</p> <p>Numeric Keypad</p> <p>Application Keypad</p> <p>The numeric keypad sends numbers from the numeric keypad. This is the default.</p> <p>The application keypad option sends escape sequences from the numeric keypad.</p>
Cursor Keys	<p>Selects the kind of control codes sent. It is disabled if the terminal is in VT52 mode. Two selections are available:</p> <p>Normal Cursor Keys</p> <p>Application Cursor Keys</p> <p>The normal cursor key option sends ANSI cursor control sequences for cursor keys (such as up, down, left, and right). This is the default.</p> <p>The application cursor key option sends application program control functions for the cursor keys.</p>
Margin Bell	<p>Determines whether the terminal sounds a bell tone when the text cursor approaches the right margin. You can select either Margin Bell or No Margin Bell. The default is Margin Bell.</p>
Warning Bell	<p>Determines whether the terminal generates a bell tone such as for operator errors, mail messages, or system messages. You can select either Warning Bell or No Warning Bell. The default is Warning Bell.</p>
Break	<p>Enables or disables the BREAK key function. You can select either Break or No Break. The default is Break.</p>

**Table 7-6 (Cont.): Keyboard Set-Up**

<b>Current Setting</b>	<b>Description and Available Selections</b>
Auto Answerback	Selects whether the answerback message is automatically sent to a host computer after a communication line connection is established. You can choose either No Auto Answerback or Auto Answerback. The default is Auto Answerback.
Answerback	Selects whether your answerback message entry is displayed on the Set-Up screen. You can choose one of the following options: Answerback Concealed Answerback Not Concealed If you select Answerback Concealed, your answerback message is not displayed on the screen, so it will not be revealed. You can only reset this option to Not Concealed by entering a new answerback message. If you select Answerback Not Concealed, your answerback message is displayed on the screen as you enter it. This is the default.
Answerback String	Allows an answerback message entry. The emulator sends an answerback message when it receives an ENQ (inquiry control character), or if you or the host send a CTRL/BREAK. In the case of ENQ, the message you enter is sent to a host without affecting screen data or requiring further operator action. When you make this selection, a cursor displays in the right column. You can enter any keyboard character to represent your answerback string. The limit is 30 characters.
Keyboard Lock	Determines the function of the Lock key. Two options are available: Caps Lock Shift Lock Caps lock sends uppercase characters for the alphabetic keys only. If you select Shift lock, the alphabetic keys send uppercase characters, while the numeric and symbol keys send the top characters.

**Table 7-6 (Cont.): Keyboard Set-Up**

Current Setting	Description and Available Selections
Auto Repeat	<p data-bbox="522 296 1134 374">Pressing the Lock key turns on the lock indicator on the keyboard. To clear the lock function, simply press the Lock key again.</p> <p data-bbox="522 395 1134 473">Determines whether keystrokes automatically repeat when you press and hold down a key. The following options are available:</p> <ul data-bbox="522 487 753 621" style="list-style-type: none"> <li data-bbox="522 487 700 515">No Auto Repeat</li> <li data-bbox="522 526 723 553">Slow Auto Repeat</li> <li data-bbox="522 564 753 591">Normal Auto Repeat</li> <li data-bbox="522 602 713 630">Fast Auto Repeat</li> </ul> <p data-bbox="522 640 1134 687">No Auto Repeat sends only one character when you press and hold down a key.</p> <p data-bbox="522 713 1134 791">The other selections send the character repeatedly if you press and hold down a key. Sending stops when the key is released. The default is No Auto Repeat.</p> <p data-bbox="522 812 1134 946"><b>Note:</b> If the terminal is in VT100 mode, running EDT under VMS, and you are holding down a key, autorepeating stops. It resumes only when you release the key, then press it again. To prevent this from happening, after the VMS prompt type the following:</p> <pre data-bbox="522 960 897 987">\$ SET TERMINAL/DEVICE=VT200</pre> <pre data-bbox="522 1001 827 1029">\$ SET TERMINAL/INQUIRE</pre>
Keyclick	<p data-bbox="522 1048 1052 1095">Determines the volume of your keyclicks. The following options are available:</p> <ul data-bbox="522 1112 706 1251" style="list-style-type: none"> <li data-bbox="522 1112 655 1140">No Keyclick</li> <li data-bbox="522 1150 669 1178">Soft Keyclick</li> <li data-bbox="522 1189 706 1216">Normal Keyclick</li> <li data-bbox="522 1227 677 1255">Loud Keyclick</li> </ul>

## Printer Screen

The Printer screen contains selections for defining printer operations, such as print size, type of printer, and print terminator.

Table 7-7 defines the Printer Set-Up selections and their available options.

**Table 7-7: Printer Set-Up**

<b>Current Setting</b>	<b>Description and Available Selections</b>
Print Area	Determines how much of the screen is printed during a text print operation. Two options are available: Print Full Page Print Scroll Region Print Full Page prints the entire screen. This is the default. Print Scroll Region prints only the area of the screen that is defined to be a scrolling region.
Printer Type	Lets you select the printer you want to use. You can choose either a DEC printer or any other non-DEC printer.
Print File Name	Allows you to send the screen text to a file. When you select this setting, the cursor flashes in the right column, indicating that you should enter a file name. You can enter either the default name (PRN), or any other file name you want to use.  Once you exit Set-Up, you press the Print Screen key to send the text to a file.
Print Mode	Selects the operating mode for the printer. The following selections are available: Normal Print Mode Auto Print Mode Printer Controller Mode

**Table 7-7 (Cont.): Printer Set-Up**

<b>Current Setting</b>	<b>Description and Available Selections</b>
Print Terminator	The Normal Print Mode sends information to the printer only when you invoke print functions from the keyboard. The Auto Print Mode prints the current text line when the terminal receives a line feed, form feed, or vertical tab code from a host. The Printer Controller Mode causes the printer port to treat the device that is connected to it as a terminal, while SETHOST monitors traffic. The printer and host computer transfer data without displaying the data on the screen.  Lets you determine whether a form feed terminator is sent at the end of a print screen operation. You can choose either Form Feed or None (for no terminator).

## Tabs Screen

The Tabs screen contains selections for defining the tab settings.

Table 7-8 defines the Tab Set-Up selections and their available options.

**Table 7-8: Tab Set-Up**

<b>Selection</b>	<b>Function</b>
Clear All Tabs	Clears all tab settings. When you select this option, the current tab settings are displayed at the bottom of your screen. The tab settings are indicated by capital Ts. When you press the Select key, all the current settings are removed.
Set 8 Column Tabs	Sets tabs every eight columns, starting with column 9. When you press the Select key, tabs are set every eight columns and are indicated with capital Ts.
Set Tab Interval	Lets you determine the interval between tabs. When you select this option, the right column displays an interval prompt with the default value of 8. You can change the default by entering a new value.  To have a new value take effect, press the Select key. The tabs are automatically reset at the value you specified.
Manually Set Tabs	Allows you to change individual tab stops, rather than have the stops at uniform intervals only. When you select this option, instructions appear in the right column. You can set or clear any tab stop by pressing the T key. When you complete your tab settings, press the Select key. The tab intervals are updated on the screen.

## **Setting Tabs Manually**

To set tabs manually for the SETHOST emulator, follow these steps:

1. Select the Manually Set Tabs option using the up or down arrow key.
2. Press the Select key.  
A blinking cursor now appears in the tabs ruler at the bottom of your Tabs Set-Up screen.
3. Use the right or left arrow key to position the cursor on the desired tab stop setting.
4. Press the Enter key to set the tab.

### **NOTE**

If you want to erase a tab stop, position the cursor on the tab setting using the left or right arrow key, then press the Enter key.

5. Press the Select key to exit manual tab setting.

## Telephone Screen

The Telephone screen lets you determine how telephone answering is accomplished. You can also enter telephone numbers that you want to use with a modem.

Table 7-9 defines the Telephone Set-Up selections and their available options. You save these settings by using the Save Parameters option (from the Actions screen) before exiting Set-Up.

**Table 7-9: Telephone Set-Up**

Current Setting	Description and Available Selections
Modem Answer Selection	<p>Lets you determine how the modem will be answered. You can choose either manual telephone answering or automatic telephone answering. Automatic answering is done by the emulator.</p> <p>Manual Answer leaves the modem in Talk mode after disconnecting from the host. This allows you to answer the incoming call. Manual Answer is required for normal telephone operation. It is the default.</p> <p>Auto Answer leaves the modem in data mode after disconnecting from the host. This allows the terminal to answer the incoming call automatically.</p>
A-J =	<p>Indicates that ten telephone numbers (A-J) can be stored for automatic dialing.</p> <p>Each time you select a letter from the left column, that letter appears in the right column while the cursor flashes to prompt you to enter a phone number.</p> <p>To change or enter a phone number, select the letter that represents the number you want to change.</p> <p>Press the Select key to move to the right column and type over the old number or enter a new number. Press the Select key or the Return key when you are done. The number now appears in the left column, next to the corresponding letter. Repeat this process until you change or enter all the desired phone numbers.</p>

## **Using Telephone Selections**

Before you can use the Telephone Set-Up selections, you must first do the following:

- Install the integral modem option (This is not available for all countries.)
- Select and save the following information from the Communications screen:
  - The integral modem selection
  - The appropriate transmit and receive speeds

For more information on telephone settings, see your modem user's guide.



## Chapter 8

---

# SETHOST Script Processor

A script is a text file with commands that allow SETHOST to perform many operations automatically. For example, a script can hold an interactive session with a host computer.

This chapter covers:

- General concepts
- Starting a script
- Session log files
- Script language
- Commands
- A sample COM log-in script

### General Concepts

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 a small amount of 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.

## Starting a Script

You specify the initial script when you start SETHOST:

```
SETHOST/SCRIPT=file-name
```

The *file-name* is the name of the script file you want to start the session. The default extension for script files is SCR; so, you need not specify the extension if the script file name ends with .SCR. For example, to use the script file LOGIN.SCR, you can type:

```
SETHOST/SCRIPT=LOGIN<RET>
```

## Session Log Files

When you start up SETHOST with a script session, the script processor always opens a session log file in which it writes a record of the script session. You can use this file to help track down problems in scripts.

The session log file has the same file name as the initial script file, with the extension LOG. So, if your initial script in the session is FIRST.SCR, the session log file for the session is FIRST.LOG.

There is a second type of log file, one that is under your control. The OPEN and CLOSE commands in the script language let you store session information in log files. When this chapter discusses log files, it refers to those under your control.

## Script Language

A script is a file containing commands. The script file is a text file that 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:

1. Communication Commands

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

The communication commands are:

BAUD RATE  
DATA BITS  
DISCONNECT  
XON/XOFF  
NO XON/XOFF  
NTS (not applicable in the standalone environment)  
PARITY  
PORT  
STOP BITS  
RX BAUD  
TX BAUD

## 2. Control Commands

The control commands define the sequence of operation in a script. They are:

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

## 3. Data Commands

The data commands manipulate and monitor data going to and coming from the communication line. The data commands are:

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

**4. Display/Print Commands**

The display/print commands control your screen and printer. They are:

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

**5. File Commands**

The file commands perform operations on files. They are:

CHAIN  
CLOSE  
LOAD  
OPEN  
SCRIPT  
SEND FROM

**6. Other Commands**

The rest of the script commands perform miscellaneous functions:

COMMENT  
KEYBOARD OFF  
KEYBOARD ON  
LABEL  
PAUSE  
RETRY  
SET  
SYSTEM  
TIMEOUT  
TIMER  
TIMER OFF

## Command Line

Each line in the script file can contain one command or a label. Or, it can be empty. Lines in a script file cannot be longer than 80 characters.

Empty lines have no effect on script execution, but they can 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.

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

## Strings

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

```
DISPLAY: string
```

When *string* appears in a format, it represents a string of characters that begins immediately after the colon. Furthermore, unless the description states otherwise, you can represent special characters, such as control characters, in the command. In the following example, the string is 8 characters long. The first character is a space and the last character is a carriage return character.

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

## Special Characters

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

1. 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.

2. You can use the symbols for special characters listed in Table 8-1.

Table 8-1 shows the non-printable characters, control characters, function keys, and keypad keys that you can use in script commands.

**Table 8-1: 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>
	-.	-.
	-.	-.
	-.	-.
Keypad Keys	F20	<F20>
	PF1	<PF1>
	PF2	<PF2>
	PF3	<PF3>
	PF4	<PF4>

**Table 8-1 (Cont.): Special Characters Used in Script Commands**

<b>Category</b>	<b>Character/Key</b>	<b>Script Representation</b>
	Keypad 0	<KP0>
	Keypad 1	<KP1>
	-	-
	-	-
	-	-
	-	-
	Keypad 9	<KP9>
	-	<KPMINUS>
	,	<KPCOMMA>
	-	<KPPERIOD>
	Enter	<KPENTER>
	Find	<Find>
	Insert Here	<INSERTHERE>
	Next	<NEXT>
	Prev	<PREV>
	Remove	<REMOVE>
	Select	<SELECT>
Non-Printable Characters	Carriage Return	<CR>, <RET>, or <RETURN>
	Delete	<DEL>
	Escape	<ESC>
	Form Feed	<FF>
	Line Feed	<LF>
	Tab	<TAB>

## Variables

Variables are named data items whose values can be set and tested 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, the SET command can delete variables when they are no longer required, so they can be replaced by 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 "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 74 characters. However, the maximum practical size of a variable is determined by the format of a command line. For more information, see the CASE, READ, and SET commands in this chapter.

## Error Processing

All script commands return a status after executing. The variables `Error_Number` and `Error_Message` contain the results. All values other than zero (0) are errors.

- 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 - MS-DOS command failed
- 13 - Print Error
- 14 - Label not found
- 15 - Logfile already open
- 16 - No logfile open
- 17 - File Transfer Error
- 18 - Variable not found
- 19 - Comm Port not available
- 20 - Nested ON ERROR routines not allowed

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

### **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. So, 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. The `TIMER OFF` command turns off the timer completely so that a timeout error cannot occur, for example.

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. That means that a script can handle errors in exactly the way required by the current situation.

## SETHOST Script Processor

For example, look at these script segments:

```
...
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, control returns to the statement following the one that caused the handler to be invoked. This allows execution of a script to continue if the ON ERROR segment was able to get around the error. The error handler can, however, transfer control elsewhere, or it can end the script and even 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.

## Commands

### BAUD RATE

The BAUD RATE command sets the rate at which characters are both received and transmitted.

#### Format

BAUD RATE: *speed*

#### Remarks

The value of *speed* must be one of the following:

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

The RX BAUD and TX BAUD commands are exactly the same as this command.

For more information, see the PORT command in this chapter.

#### Example

```
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

*SETHOST Script Processor*  
**BREAK**

**BREAK**

The **BREAK** command 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**.

**Remarks**

**SHORT** specifies a length of 0.24 seconds.

**LONG** specifies a length of 3.5 seconds.

For more information, see the **PORT** command in this chapter.

**Example**

Break:35  
**BREAK:** **LONG**  
**BREAK:**Short

The first two examples are equivalent.

**Results**

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

## CASE

The CASE command selects an action depending on the contents of a variable.

### Format

```
CASE: variable  
  
"string" command  
"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 itself and the name of the controlling variable. CASE END marks the end of the CASE command, and it is required.

If there is a DEFAULT command, it must appear immediately before CASE END.

The *variable* must exist before the CASE command executes. That is, the *variable* must have a value that was assigned to it by a SET or READ command. The *variable* can also 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 *command* paired with the first matching "string" (if any) is executed.

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 *command* associated with the DEFAULT command is executed.

If there is no DEFAULT command and there is no match between any "string" and the *variable*, control transfers to the command following CASE END.

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

## SETHOST Script Processor

### CASE

#### Example

1. This example waits for the user to enter a day of the week. The input is stored 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 the user types "MO" or "mo", control transfers 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, the script SENDFILE.SCR is called. 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" could 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", control transfers 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**

The **CASE END** command terminates the multiple-line **CASE** command.

### **Format**

**CASE END:**

### **Example**

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

### **Results**

0 - Command Successfully Completed

SETHOST Script Processor  
CHAIN

## CHAIN

The CHAIN command 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.

For more information, see the SCRIPT and EXIT SCRIPT commands in this chapter.

The default extension of the script file is SCR.

### Example

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

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

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.

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

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
```

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

*SETHOST Script Processor*  
**CLEAR LINE**

**CLEAR LINE**

The CLEAR LINE command clears all data from the communication line buffer.

**Format**

CLEAR LINE:

**Remarks**

The CLEAR LINE command discards all characters that have been 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**

The CLOSE command 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.

For more information, see the OPEN command in this chapter.

### **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 is then closed and the script continues.

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

### **Results**

```
0 - Command Successfully Completed  
16 - No logfile open
```

*SETHOST Script Processor*  
**COMMENT**

**COMMENT**

The COMMENT command causes the remainder of the command line to be ignored.

**Format**

COMMENT: *comment-text*

**Remarks**

You can use COMMENT commands to make your scripts more understandable to others.

**Example**

```
...  
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

The DATA BITS command 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 specifies the number of data bits per character in communication line text.

For more information, see the PORT command in this chapter.

### Example

```
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

*SETHOST Script Processor*  
**DEBUG**

**DEBUG**

The **DEBUG** command displays each line of the script before it is executed.

**Format**

**DEBUG:**

**Remarks**

The **DEBUG** command displays each line of the script on the screen before it is executed.

Use the **NO DEBUG** command to stop the display.

**Example**

You generally use the **DEBUG** command only in scripts that 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.

In this example, for instance, if **NO ECHO** 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

The DIAL command 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 DIAL.

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

For more information, see the PORT command in this chapter.

### Example

The first example shows a usual combination of communication parameters.

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

### Results

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

*SETHOST Script Processor*  
**DISCONNECT**

**DISCONNECT**

The DISCONNECT command specifies the length of time to wait before disconnecting after losing the carrier.

**Format**

DISCONNECT: 2

or

DISCONNECT: 60

**Remarks**

When the modem carrier signal stops, it usually means that further communication is not possible without making a new connection. The DISCONNECT command allows you to specify how long to wait before hanging up automatically.

DISCONNECT:2 specifies a 2-second wait.

DISCONNECT:60 specifies a 60-millisecond wait.

For more information, see the PORT command in this chapter.

**Example**

```
PORT: Data-1
PARITY: None
DATA BITS: 8
STOP BITS: 1
BAUD RATE: 2400
DISCONNECT: 2
DIAL:E
```

**Results**

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

## DISPLAY

The DISPLAY command writes a string of characters to the user's screen.

### Format

DISPLAY: *string*

### Remarks

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

The DISPLAY command does not affect the contents of the log file.

The *string* can contain control characters to perform screen formatting.

This command 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
```

Note the positioning of the output. The asterisk (\*) shows the position of the cursor on the screen after the DISPLAY commands execute:

```
AbcdEfghi Jklmno
Pqrst*
```

### Results

0 - Command Successfully Completed

*SETHOST Script Processor*  
**DTR CLEAR**

**DTR CLEAR**

The DTR CLEAR command turns off the DTR signal between the computer and the modem.

**Format**

DTR CLEAR:

**Remarks**

The DTR CLEAR command turns off the DTR signal between the computer and the modem. Most modems hang up when the DTR signal is turned off.

The DTR CLEAR command is invalid when you use NETWORK, LAT, or CTERM in a PORT command.

For more information, see the PORT command in this chapter.

**Example**

This segment of a script 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

The DTR SET command turns on the DTR signal between the computer and the modem.

### Format

DTR SET:

### Remarks

The DTR SET command turns on the DTR signal between the computer and the modem.

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

For more information, see the PORT command in this chapter.

### Example

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 "HELLO". It then sets the timer to allow timeouts, and continues.

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

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

*SETHOST Script Processor*  
**ECHO**

**ECHO**

The ECHO command causes all characters received through the communication line to be sent to the screen.

**Format**

ECHO:

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

The ECHO command causes all characters received over the communication line to be sent to the screen.

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

**Example**

This script segment finishes logging into a host. The user sees nothing, because the default is NO ECHO. 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**

The END ON ERROR command 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 to be executed when the script processor detects an error.

ON ERROR and END ON ERROR must occur in pairs.

See the ON ERROR command.

### **Results**

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

*SETHOST Script Processor*  
**EXIT EMULATOR**

**EXIT EMULATOR**

The EXIT EMULATOR command ends processing of both the current script and SETHOST.

**Format**

EXIT EMULATOR:

**Remarks**

EXIT EMULATOR 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

The EXIT EMULATOR ON command ends processing of both the current script and SETHOST if a specific string of characters is received from the communication line.

### Format

EXIT EMULATOR ON: *string*

### Remarks

The EXIT EMULATOR ON command waits until *string* is received over the communication line or the command times out.

If the script processor detects the *string*, it ends the script and session in the same way as the EXIT EMULATOR command. See EXIT EMULATOR.

If the EXIT EMULATOR ON command times out, the timeout error condition occurs.

For more information, see the RETRY and ON ERROR commands and the section on Error Processing in this chapter.

### Example

In this example, the script has tried to connect to a remote node. If the string "not reachable" is detected, 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

*SETHOST Script Processor*  
**EXIT SCRIPT**

**EXIT SCRIPT**

The **EXIT SCRIPT** command immediately ends processing for the current script.

**Format**

**EXIT SCRIPT:**

**Remarks**

The **EXIT SCRIPT** command ends processing of the current script.

If a **SCRIPT** command was executed 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**.

For more information, see the **SCRIPT** command in this chapter.

**Example**

These examples show how scripts can call one another and pass information in script variables:

SETHOST Script Processor  
EXIT SCRIPT

```
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:
```

*SETHOST Script Processor*  
**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

The EXIT SCRIPT ON command ends processing of the current script if a specific string of characters is received.

### 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 the *string*, it ends processing of the current script in the same way as the EXIT SCRIPT command. See EXIT SCRIPT.

If the EXIT SCRIPT ON command times out, the timeout error condition occurs.

For more information, see the RETRY and ON ERROR commands and the section on Error Processing in this chapter.

### Example

```
$restart:
COMMENT: Set up empty error handler so script will fall through
COMMENT: 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, it means that something is wrong because
COMMENT: the report is taking too long. We will try it again.
COMMENT: Set up an error handler to quit SETHOST 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

*SETHOST Script Processor*  
**FPRINT**

**FPRINT**

The FPRINT command 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. Error handling is not shown.

```
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

The GOTO command 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.

For more information, see the LABEL command in this chapter.

### 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:
```

*SETHOST Script Processor*  
**GOTO**

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

The HANG UP command is the same as the DTR CLEAR command. It turns off the DTR signal between the computer and the modem.

### Format

HANG UP:

### Remarks

The HANG UP command is the same as the DTR CLEAR command. It turns off the DTR signal between the computer and the modem.

For more information, see the PORT command in this chapter.

### 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

*SETHOST Script Processor*  
**KEYBOARD OFF**

**KEYBOARD OFF**

The **KEYBOARD OFF** command causes **SETHOST** to start ignoring **<CTRL/C>** and **<CTRL/F10>** keystrokes 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 a **READ** or **TYPE UNTIL** command executes, the keyboard is temporarily turned on during execution of the command.

If the user enters **<CTRL/C>** or **<CTRL/F10>** while **KEYBOARD OFF** is in effect, the keystroke may be recognized later if a **KEYBOARD ON** command executes, because keystrokes are stored temporarily in a "typeahead buffer" and may still be there when keyboard input is again accepted.

**Example**

This script segment sends the contents of a file after disabling the keyboard. When the file has been sent, the user is again able to end the session from the keyboard.

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

**Results**

0 - Command Successfully Completed

## KEYBOARD ON

The KEYBOARD ON command 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 a READ or TYPE UNTIL command executes, the keyboard is temporarily turned on during execution of the command.

### Example

This script segment sends the contents of a file after disabling the keyboard. When the file has been sent, the user is again able to end the session from the keyboard.

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

### Results

0 - Command Successfully Completed

*SETHOST Script Processor*  
**LABEL**

**LABEL**

A *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

The LOAD command 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, the LOAD command ends with an error.

If *file-name* appears but the file is not found, the LOAD command ends with an error.

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

### 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

*SETHOST Script Processor*  
**NO DEBUG**

**NO DEBUG**

The NO DEBUG command stops displaying script lines before execution.

**Format**

NO DEBUG:

**Remarks**

The NO DEBUG command reverses the effect of a DEBUG command. NO DEBUG is the default when SETHOST begins.

**Example**

For more information, see the DEBUG command in this chapter.

**Results**

0 - Command Successfully Completed

## **NO ECHO**

The NO ECHO command turns off the display of characters received over 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 displayed by DEBUG and DISPLAY.

### **Example**

For more information, see the ECHO command in this chapter.

### **Results**

0 - Command Successfully Completed

*SETHOST Script Processor*  
**NO SKIP ON**

**NO SKIP ON**

The NO SKIP ON command executes the next line in the script if a specific string of characters is received.

**Format**

NO SKIP ON: *string*

**Remarks**

The NO SKIP ON command waits until *string* is received over the communication line or the command times out.

If the script processor detects the *string*, it transfers control to the next line.

If the NO SKIP ON command times out, control transfers to the line following the next line.

A timeout error cannot occur, because timeout is a normal result of the NO SKIP ON command.

**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>
```

For more examples, see the SKIP ON command in this chapter.

**Results**

0 - Command Successfully Completed

## **NO XON/XOFF**

The NO XON/XOFF command 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/XOFF command if the script processor should ignore XON and XOFF characters.

For more information, see the XON/XOFF command in this chapter.

### **Example**

NO XON/XOFF:

### **Results**

- 0 - Command Successfully Completed
- 5 - Incompatible Settings

## NTS

The NTS command specifies a Network Terminal Service.

### NOTE

This command is not applicable in the VAXmate standalone environment.

### Format

NTS: *service*

### Remarks

The NTS command specifies a Network Terminal Service that the script session will connect to.

The *service* can be a LAT service name or a DECnet node name.

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

When the NTS command executes, it ends with an error if the PORT command in effect does not specify NETWORK, LAT, or CTERM.

For more information, see the PORT command in this chapter.

The NTS command is optional if the SETHOST Set-Up parameters specify LAT, CTERM, or NET and a valid service.

If the PORT command specifies LAT, the NTS *service* must be a LAT service. If the PORT command specifies CTERM, the NTS *service* must be a valid DECnet node name.

### Example

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

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

### Results

- 0 - Command Successfully Completed
- 4 - Invalid Parameter
- 5 - Incompatible Settings
- 7 - Service not available

## ON ERROR

The ON ERROR command 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 to be executed when the script processor detects an error.

ON ERROR and END ON ERROR must occur in pairs.

The ON ERROR command marks the next line as the first line to be executed when an error occurs in the current script.

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

Until an ON ERROR command executes in the current script, no error handling is defined.

The lines between ON ERROR and END ON ERROR are ignored by the script processor 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.

For more information, see the section on Error Processing in this chapter.

*SETHOST Script Processor*  
**ON ERROR**

For more information on timeout error handling, see the **RETRY** command in this chapter.

**Example**

For an example, see the section on Error Processing in this chapter.

**Results**

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

## OPEN

The OPEN command reads 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 file is not found, the OPEN command creates it.

If the file is found, the OPEN command opens the file and appends the new text at the end of the existing file.

After the OPEN command executes, all characters received from the communication line that are sent to the screen are also stored 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 is always created automatically by the script processor. 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 text received from the communication line is appended 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
```

*SETHOST Script Processor*  
*OPEN*

**Results**

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

## PARITY

The PARITY command 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. For more information, see the PORT command in this chapter.

### Example

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

```
PARITY: odd
```

```
PARITY: none
```

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

**SETHOST Script Processor**  
**PAUSE**

**PAUSE**

The PAUSE command suspends processing for a specified time.

**Format**

PAUSE: *length-of-pause*

**Remarks**

The PAUSE command causes processing of the script to be suspended 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 zeroes, as in the examples that follow.

**Example**

This first set of examples explains 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

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
```

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

**SETHOST Script Processor**  
**PORT**

**PORT**

The PORT command 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  
NETWORK

**NOTE**

CTERM, LAT, and NETWORK are not available as methods in the VAXmate standalone environment.

**Remarks**

The PORT command tells the script processor how to communicate with the host computer. If there is no PORT command, the SETHOST Set-Up defaults are used.

DATA-1 means Port 1, data leads only.

MODEM-1 means Port 1, full modem control.

DATA-2 means Port 2, data leads only.

MODEM-2 means Port 2, full modem control.

INTEGRAL-2 means Port 2, integral modem.

CTERM specifies the DECnet CTERM protocol.

LAT specifies the LAT protocol.

NETWORK specifies use of the LAT protocol, but if that fails, then CTERM.

For more information, see the NTS command in this chapter.

**Example**

PORT: NETWORK

PORT: Data-1

**Results**

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

*SETHOST Script Processor*  
**PRINT SCREEN**

**PRINT SCREEN**

The PRINT SCREEN command prints the current screen on the local printer.

**Format**

PRINT SCREEN: [*file-name*]

**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 sends the contents of the screen to the local printer. The second uses a file named SCRNL.

PRINT SCREEN:

PRINT SCREEN: scrnl

**Results**

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

## **PRINTER OFF**

The **PRINTER OFF** command turns off echo printing.

### **Format**

**PRINTER OFF:**

### **Remarks**

Reverses the action of the **PRINTER ON** command.

**PRINTER OFF** is the default when the script session starts.

For more information, see the **PRINTER ON** command in this chapter.

### **Example**

**PRINTER OFF:**

### **Results**

0 - Command Successfully Completed

*SETHOST Script Processor*  
**PRINTER ON**

**PRINTER ON**

The PRINTER ON command turns on echo printing.

**Format**

PRINTER ON: [*file-name*]

**Remarks**

After the PRINTER ON command executes, all text received from the communication line is sent 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 output is appended to the end of the file. Otherwise, PRINTER ON creates a new file with the name specified in *file-name*.

**Example**

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

PRINTER ON:

PRINTER ON: scrnl

**Results**

- 0 - Command Successfully Completed
- 4 - Invalid Parameter

## PURGE TYPE

The PURGE TYPE command clears the contents of the typeahead buffer.

### Format

PURGE TYPE:

### Remarks

Keyboard input is stored temporarily in the typeahead buffer until the script processor is ready to use it. The PURGE TYPE command clears the contents of the typeahead buffer. Use this command when you want the script to ignore previous keystrokes from the user.

### Example

This is a segment of a script that 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

The READ command 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 the Return key. The characters typed before the Return key is pressed are stored in the variable named by *variable-name*.

The user can type any number of characters before pressing the Return key. However, the READ command stores no more than the first 74 characters, and the excess characters are ignored by the script processor.

### NOTE

ECHO must be on when the READ command executes for the user to be able to see the input on the screen.

### 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

**RETRY**

The RETRY command 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 zero when it starts the first script, just as if the first command in the script were RETRY: 0.

A script's retry counter is also set to zero when the script is called with a SCRIPT or CHAIN command.

The RETRY command stores the number *counter*, which must not be negative, in the script's retry counter.

For more information, see the section on Error Processing in this chapter.

**Example**

The first example sets the script's retry counter to 3, which means that unless another RETRY command executes, each time (up to 3) that a timeout error occurs, the retry counter is decremented and control transfers 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.

The second example sets the script's retry counter to 0, which means that reruns do not occur when timeout errors are detected.

RETRY: 3

...

RETRY: 0

**Results**

- 0 - Command Successfully Completed
- 4 - Invalid Parameter

## **RX BAUD**

The RX BAUD command sets the rate at which characters are both received and transmitted.

### **Format**

RX BAUD: *speed*

### **Remarks**

The RX BAUD command is exactly the same as the BAUD RATE command. For more information, see the BAUD RATE command in this chapter.

## SCRIPT

The SCRIPT command 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 control returns to the line following it as the result of executing an EXIT SCRIPT command in a called script. The script processor also saves other 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. For more information, see the section on Error Processing in this chapter.
- 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 can be called. A script can call itself with a SCRIPT command, or it can be called by a script it calls 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.

For example coding, 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

For more examples of the SCRIPT command, see the section on Error Processing in this chapter.

### 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.

## SETHOST Script Processor

### SEND

#### SEND

The SEND command sends a string of characters through the communication line.

#### Format

SEND: *string*

#### Remarks

The *string* is examined for special character translation and is then sent 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

The SEND FROM command 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 contents of the file are treated as a stream of bytes.

### 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

## SET

The SET command 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 returned by the called script. If the value is not either "ERROR" or "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
```

SETHOST Script Processor  
SKIP

**SKIP**

The SKIP command 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**

These 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 "ok" is not detected and a timeout occurs, both 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.

But 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

**SETHOST Script Processor**  
**SKIP ON**

**SKIP ON**

The SKIP ON command 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 *string* is received over the communication line or 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.

For more information, see the RETRY and ON ERROR commands and the section on Error Processing in this chapter.

**Example**

For an example of the SKIP ON command, see the PAUSE command in this chapter.

**Results**

- 0 - Command Successfully Completed
- 2 - Timeout

## **STOP BITS**

The STOP BITS command sets the number of stop bits to 1 or 2.

### **Format**

STOP BITS: 1

or

STOP BITS: 2

### **Remarks**

For more information, see the PORT command in this chapter.

### **Example**

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

*SETHOST Script Processor*  
**SYSTEM**

**SYSTEM**

The SYSTEM command executes an MS-DOS system command.

**Format**

SYSTEM: [*command*]

**Remarks**

If there is no *command*, the user is left at the MS-DOS prompt. The user can return to the script by typing the MS-DOS EXIT command.

If there is a *command*, the script processor passes it to MS-DOS for execution.

Any output from the MS-DOS command will appear on the screen.

**Example**

For examples of the SYSTEM command, see the DIAL command in this chapter.

**Results**

0 - Command Successfully Completed  
12 - MS-DOS command failed

## TIMEOUT

The TIMEOUT command forces an immediate timeout error condition.

### Format

TIMEOUT:

### Remarks

The timeout error condition occurs.

If control is not transferred elsewhere as a result of an ON ERROR segment or RETRY command, then the script processor executes an implicit EXIT SCRIPT command, and the script ends.

For more information, see the RETRY and ON ERROR commands and the section on Error Processing in this chapter.

### 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 block, if one is active. If there is no active ON ERROR block, 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 zero value.

**SETHOST Script Processor**  
**TIMER**

**TIMER**

The **TIMER** command 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.

For more information, see the **TIMER OFF** command in this chapter.

**Example**

For an example of the **TIMER** command, see the **DTR SET** command in this chapter.

**Results**

- 0 - Command Successfully Completed
- 4 - Invalid Parameter

## **TIMER OFF**

The **TIMER OFF** command 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**

For an example of the **TIMER OFF** command, see the **DTR SET** command in this chapter.

### **Results**

0 - Command Successfully Completed

*SETHOST Script Processor*  
**TX BAUD**

**TX BAUD**

The TX BAUD command sets the rate at which characters are both received and transmitted.

**Format**

TX BAUD: *speed*

**Remarks**

The TX BAUD command is exactly the same as the BAUD RATE command. For more information, see the BAUD RATE command in this chapter.

## TYPE UNTIL

The TYPE UNTIL command 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 generated characters are sent 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

For an example of the TYPE UNTIL command, see the OPEN command in this chapter.

### Results

0 - Command Successfully Completed

**SETHOST Script Processor**  
**WAIT FOR**

**WAIT FOR**

The WAIT FOR command suspends processing until a specific string of characters is received from the communication line.

**Format**

WAIT FOR: *string*

**Remarks**

The WAIT FOR command waits until *string* is received over the communication line or 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**

For an example of the WAIT FOR command, see the SEND command in this chapter.

**Results**

- 0 - Command Successfully Completed
- 2 - Timeout

## XON/XOFF

The XON/XOFF command causes the script processor to use the XON/XOFF protocol.

### Format

XON/XOFF: *buffer-limit*

### Remarks

The XON/XOFF command turns on use of the XON/XOFF 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.

The value of *buffer-limit* must be one of the following:

64

256

512

1024

### Example

XON/XOFF: 256

### Results

0 - Command Successfully Completed  
4 - Invalid Parameter

## A Sample COM Log-In Script

This section contains a sample script that you can use with SETHOST. It shows steps for creating a script file and includes comments about how the commands operate within the script.

This sample creates a script file (COMLOGIN.SCR) that will log you into a VMS host using the COM1 port. The 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.
2. Use the ECHO command so that you can see what is happening as the script executes each command.

```
PORT:Data-1
```

```
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 4 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, you need to send a carriage return character to get the username prompt. Include the WAIT FOR command here.

```
SEND:<CR>
```

```
WAIT FOR:Username:
```

5. Once the username prompt appears, you can cancel any remaining RETRY 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 4 times).

RETRY:4

WAIT FOR:Password:

8. The password prompt appears, so you can cancel the RETRY 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. An exception may be if 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:



# Index

---

## A

---

Actions Set-Up screen, 7-9  
Add a partition  
    choosing size, 2-7  
AUTOEXEC.BAT.bat file  
    modifying, 4-2  
AUTOEXEC.BAT Command Lines,  
    4-3t

## B

---

BAUD RATE command, 8-11  
Blank screen  
    corrective procedure, 5-2  
Blinking cursor  
    corrective procedure, 5-3  
BREAK command, 8-12  
Brightness control  
    adjusting, 5-2

## C

---

CASE command, 8-13  
CASE END command, 8-15  
Caution  
    FDISK, 2-4  
    FORMAT, 2-4  
CHAIN command, 8-16  
Character attributes  
    displaying, 7-21  
CLEAR LINE command, 8-18

CLOSE command, 8-19  
COMMENT command, 8-20  
Communications Set-Up screen,  
    7-14  
CONFIG.SYS Command Lines, 4-4t  
CONFIG.SYS file  
    modifying, 4-4  
Contrast control  
    adjusting, 5-2  
Conventions  
    Ctrl/Alt/Del keys, xi  
Copying diskettes, A-1  
Country specific set-up, 4-2  
Ctrl/Alt/Del Keys, 2-9f  
Ctrl/P key sequence, 4-7  
Customizing start-up, 4-2

## D

---

DATA BITS command, 8-21  
Date  
    format, 2-5  
DEBUG command, 8-22  
DIAL command, 8-23  
Directory structure  
    after software installation, 3-4  
DISCONNECT command, 8-24  
DISKCOPY command, A-1  
Diskette  
    inserting, 2-3f  
    parts of, 2-2f

Diskette drive  
  inserting diskettes, 2-3  
  locking, 2-3  
Diskettes  
  copying, A-1  
  handling, 2-2  
  precautions, 2-2  
DISPLAY command, 8-25  
Display Set-Up screen, 7-17  
Drive type  
  RD31, 2-7  
  RD32, 2-7  
  setting, 2-6  
DTR CLEAR command, 8-26  
DTR SET command, 8-27

## E

---

ECHO command, 8-28  
END ON ERROR command, 8-29  
EXIT EMULATOR command, 8-30  
EXIT EMULATOR ON command,  
  8-31  
EXIT SCRIPT command, 8-32  
EXIT SCRIPT ON command, 8-35  
Extended self-test, 2-1

## F

---

Fast Text Only mode  
  displaying character attributes,  
    7-21  
  viewing columns, 7-20  
FDISK  
  error messages, 5-6  
FDISK utility, 2-4  
  caution, 2-4, 2-5  
  command format, 2-5  
  menu selections, 2-5  
  startup partition, 2-8  
FORMAT  
  error messages, 5-6  
FORMAT utility, 2-9  
  caution, 2-4, 2-9  
  command format, 2-9

FORMAT utility (cont'd.)  
  /S function, 2-9  
FPRINT command, 8-36

## G

---

General Set-Up screen, 7-22  
Getting help for  
  SETHOST, 6-3  
GOTO command, 8-37  
GRAPHICS command, 4-7

## H

---

HANG UP command, 8-39  
Hard disk  
  formatting, 2-9  
  setting the startup partition, 2-8  
Hard disk directory structure, 3-4f  
Hard disk drive, 2-1  
  adding a partition, 2-7  
  backing up files, 5-5  
  initializing, 2-4  
  initializing the drive, 2-6  
  partitioning, 2-4  
  preventing problems, 5-5  
  setting the drive type, 2-6  
  setup program menu, 2-5  
Hardware installation, 1-5

## I

---

Information System  
  menu selections, 1-5  
  software installation, 3-3  
Initialize the hard disk, 2-6  
Inserting a Diskette, 2-3f  
Installation Error Messages, 5-6t  
Installation instructions  
  VAXmate standalone software,  
    3-2

## K

---

Keyboard  
  default, 3-2  
KEYBOARD OFF command, 8-40

KEYBOARD ON command, 8-41  
Keyboard selection, 3-2  
Keyboard Set-Up screen, 7-25

## L

---

Label  
    in sethost scripts, 8-42  
Line attributes  
    displaying, 7-21  
LOAD command, 8-43

## M

---

MS-DOS diskette  
    software installation, 3-3  
MS-DOS Executive application, 1-4  
MS-DOS operating system, 1-3  
    features, 1-3  
MS-Windows  
    desktop applications, 1-4  
    software installation, 3-3  
MS-Windows user interface, 1-4

## N

---

NO DEBUG command, 8-44  
NO ECHO command, 8-45  
Non-default configuration files  
    using, 7-2  
NO SKIP ON command, 8-46  
NO XON/XOFF command, 8-47  
NTS command, 8-48

## O

---

ON ERROR command, 8-49  
OPEN command, 8-51  
Operating system, 1-3  
    features, 1-3  
    installation, 3-3

## P

---

PARITY command, 8-53  
Partition  
    startup, 2-8

Partitions  
    adding, 2-7  
Parts of a Diskette, 2-2f  
PATH command, 4-3  
PAUSE command, 8-54  
PORT command, 8-56  
Powerup self-test, 2-4  
Powerup test  
    blank screen, 5-2  
    blinking cursor, 5-3  
    number display, 5-3  
Powerup Test Bar, 2-4f  
Powerup test numbers, 5-4  
Powerup Test Numbers and  
    Solutions, 5-4t  
Print commands, 4-6  
PRINTER OFF command, 8-59  
PRINTER ON command, 8-60  
Printers, 4-5  
Printer setup, 4-5  
Printer Set-Up screen, 7-28  
Print screen, 4-7  
PRINT SCREEN command, 8-58  
Problem solving  
    blank screen, 5-2  
    blinking cursor, 5-3  
    installation messages, 5-6  
    screen displays a number, 5-3  
    startup, 5-1  
PURGE TYPE command, 8-61

## R

---

READ command, 8-62  
Remote hosts  
    accessing, 6-2  
RETRY command, 8-64  
RX BAUD command, 8-65

## S

---

SCRIPT command, 8-66  
Scripts  
    using, 7-3  
    *see also* Chapter 8

- Selection settings
  - changing, 7-7
  - recalling, 7-6
  - saving, 7-6
- SEND command, 8-68
- SEND FROM command, 8-69
- Serial Communications Port, 4-6
- Serial Printer Port, 4-6
- Session logging
  - receiving characters from a host, 7-12
- SET command, 8-70
- Sethost
  - help, 6-3
- SETHOST, 4-7, 6-1
  - exiting, 6-7
  - features, 6-2
  - overview, 6-2
  - starting, 6-6
- SETHOST function keys, 6-3
  - Arrow keys, 6-5
  - COMPOSE, 6-5
  - Ctrl/F10, 6-5
  - Ctrl/F2, 6-5
  - Ctrl/F5, 6-5
  - F1 (Hold), 6-5
  - F10 (Exit), 6-5
  - F11 (ESC), 6-5
  - F12 (BS), 6-5
  - F13 (LF), 6-5
  - F2 (Print Screen), 6-5
  - F3 (Set-Up), 6-5
  - F4 (Data/Talk), 6-5
  - F5 (Break), 6-5
  - Next, 6-5
  - Prev, 6-5
  - Shift/F1, 6-5
  - Shift/F5, 6-5
- SETHOST keyboard keys, 7-8
- Set-Up files
  - recalling, 7-5
  - saving, 7-5
- Set-Up screens
  - Actions, 7-9

- Set-Up screens (cont'd.)
  - Communications, 7-14
  - Display, 7-17
  - General, 7-22
  - Keyboard, 7-25
  - Printer, 7-28
  - Tabs, 7-30
  - Telephone, 7-32
  - using, 7-3
- Shift/Prt Sc Keys, 4-7f
- Shift/Prt Sc key sequence, 4-7
- SKIP command, 8-72
- SKIP ON command, 8-74
- Standalone system
  - software installation, 3-2
  - software installation guidelines, 3-1
- Startup partition, 2-8
- Startup preferences, 4-2
- Startup problems
  - blank screen, 5-2
  - blinking cursor, 5-3
  - correcting, 5-1
  - screen displays a number, 5-3
- Status information
  - displaying, 7-11
- STOP BITS command, 8-75
- SYSTEM command, 8-76
- System start-up, 4-2

## T

---

- Tabs
  - setting manually, 7-31
- Tabs Set-Up screen, 7-30
- Telephone selection
  - using, 7-33
- Telephone Set-Up screen, 7-32
- Terminal emulators, 4-7
- Test bar display, 2-4
- Testing
  - extended self-test, 2-1
- Time
  - format, 2-5
- TIMEOUT command, 8-77

TIMER command, 8-78  
TIMER OFF command, 8-79  
TX BAUD command, 8-80  
TYPE UNTIL command, 8-81

## V

---

VAXmate system  
  list of hardware features, 1-2  
  list of software features, 1-2  
VAXmate workstation  
  hardware installation, 1-5  
Video modes

  using, 7-20  
VT220 terminal emulator, 4-7

## W

---

WAIT FOR command, 8-82  
Write-protect notch  
  location, 2-2, 2-3  
  tab, 2-3

## X

---

XON/XOFF command, 8-83



**READER'S COMMENTS**

Your comments and suggestions help us to improve the quality of our publications.

**For which tasks did you use this manual?** (Circle your responses.)

- (a) Installation                      (c) Maintenance                      (e) Training  
(b) Operation/use                      (d) Programming                      (f) Other (Please specify.) \_\_\_\_\_

**Did the manual meet your needs?** Yes  No  Why? \_\_\_\_\_

**Please rate the manual in the following categories.** (Circle your responses.)

	Excellent	Good	Fair	Poor	Unacceptable
Accuracy (product works as described)	5	4	3	2	1
Clarity (easy to understand)	5	4	3	2	1
Completeness (enough information)	5	4	3	2	1
Organization (structure of subject matter)	5	4	3	2	1
Table of Contents, Index (ability to find topic)	5	4	3	2	1
Illustrations, examples (useful)	5	4	3	2	1
Overall ease of use	5	4	3	2	1
Page Layout (easy to find information)	5	4	3	2	1
Print Quality (easy to read)	5	4	3	2	1

**What things did you like *most* about this manual?** \_\_\_\_\_

**What things did you like *least* about this manual?** \_\_\_\_\_

**Please list and describe any errors you found in the manual.**

Page	Description/Location of Error
_____	_____
_____	_____
_____	_____

**Additional comments or suggestions for improving this manual:** \_\_\_\_\_

Name \_\_\_\_\_ Job Title \_\_\_\_\_  
 Street \_\_\_\_\_ Company \_\_\_\_\_  
 City \_\_\_\_\_ Department \_\_\_\_\_  
 State/Country \_\_\_\_\_ Telephone Number \_\_\_\_\_  
 Postal (ZIP) Code \_\_\_\_\_ Date \_\_\_\_\_

-----  
**Fold Here and Tape**  
-----

**Affix  
Stamp  
Here**



**DIGITAL EQUIPMENT CORPORATION  
CORPORATE USER PUBLICATIONS  
200 FOREST STREET MR01-2/L12  
MARLBOROUGH, MA 01752-9101**

-----  
**Fold Here**  
-----

