Guide to VAX/VMS Software Installation

September 1984

This manual provides detailed information to support the software installation booklets. It also includes information to support the installation of system upgrades, maintenance updates, and optional software products.

Revision/Update Information:	This is a new manual.
Software Version:	VAX/VMS Version 4.0

September 1984

The postpaid READER'S COMMENTS form on the last page of this document requests the user's critical evaluation to assist in preparing future documentation.

The following are trademarks of Digital Equipment Corporation:

DEC	DIBOL	UNIBUS
DEC/CMS	EduSystem	VAX
DEC/MMS	IAS	VAXcluster
DECnet	MASSBUS	VMS
DECsystem–10	PDP	VT
DECSYSTEM–20	PDT	
DECUS	RSTS	
DECwriter	RSX	digital

ZK–2327

This document was prepared using an in-house documentation production system. All page composition and make-up was performed by TEX, the typesetting system developed by Donald E. Knuth at Stanford University. TEX is a registered trademark of the American Mathematical Society.

Contents

Contents

iv

Contents

Contents

vi

Contents

Contents

Contents

Contents

INDEX

FIGURES

Contents

Preface

This manual contains information that supports the various
software installation booklets used to install VAX/VMS
operating systems. It also includes instructions to support
the installation of system upgrades, maintenance updates, and
optional software products.

All references to VAX–11/780 apply to the VAX–11/782 and
VAX–11/785 except as noted. For information on installing
MicroVAX, see the *MicroVAX User's Guide*.

Intended Audience

This manual is intended for all users who are qualified to install
software on VAX processors.

Structure of This Document

This manual contains information that supports the various
software installation booklets used to install VAX/VMS
operating systems. It also includes instructions to support
the installation of system upgrades, maintenance updates, and
optional software products.

All references to VAX–11/780 apply to the VAX–11/782 and
VAX–11/785 except as noted. For information on installing
MicroVAX, see the *MicroVAX User's Guide*.

This manual includes seven chapters and four appendixes.

- Chapter 1 introduces software installation on VAX
  computers and includes an overview of the installation
  process.

- Chapter 2 describes the CPU console controls and indicators
  for the various VAX processors that relate to software
  installation.

- Chapter 3 describes the controls and indicators for the various storage devices used to install the operating system. These include console storage devices, disk drives and magnetic tape drives. This chapter describes how to load media into the various drives.

- Chapter 4 expands on the installation procedures.

- Chapter 5 describes the VMSINSTAL command procedure.

- Chapter 6 explains how to upgrade your operating system to Version 4.0.

- Chapter 7 describes the User Environment Test Package and includes some instructions on operator-level troubleshooting.

- Appendix A lists the files that make up a Version 4 operating system.

- Appendix B provides source kit licensees with instructions on how to invoke the source kit command procedure and a pointer to supporting documentation.

- Appendix C outlines the VMSUPDATE command procedure that was previously used to install updates and products onto the operating system.

- Appendix D describes the contents of the various software distribution kits.

Associated Documents

The following documents include material that may be helpful in doing installations of the operating system on VAX processors, and are referenced throughout this manual:

- *Guide to VAX/VMS System Management and Daily Operations*

- *VAX/VMS DCL Dictionary*

- *VAX/VMS Utilities Reference Volume*

Conventions Used in This Document

Convention	Meaning
[RET]	A symbol with a one- to three-character abbreviation indicates that you press a key on the terminal, for example, [RET] .
[CTRL/x]	The phrase CTRL/x indicates that you must press the key labeled CTRL while you simultaneously press another key, for example, CTRL/C, CTRL/Y, CTRL/O.
$ SHOW TIME 05-JUN-1985 11:55:22	Command examples show all output lines or prompting characters that the system prints or displays in black letters. All user-entered commands are shown in red letters.
$ TYPE MYFILE.DAT . . .	Vertical series of periods, or ellipsis, mean either that not all the data that the system would display in response to the particular command is shown or that not all the data a user would enter is shown.
file-spec,...	Horizontal ellipsis indicates that additional parameters, values, or information can be entered.
[logical-name]	Square brackets indicate that the enclosed item is optional. (Square brackets are not, however, optional in the syntax of a directory name in a file specification or in the syntax of a substring specification in an assignment statement.)

Convention	Meaning
quotation marks apostrophes	The term quotation marks is used to refer to double quotation marks ("). The term apostrophe (') is used to refer to a single quotation mark.

1 Overview of Software Installation

You install software on a VAX processor by restoring (or converting) the software from a transportable form on magnetic tape or disk to a form that can be bootstrapped from a disk and run on a processor.

There are three kinds of VAX/VMS software installation:

- Installing a newly purchased operating system

- Upgrading your system to the most recent major version

- Installing maintenance updates and optional software products

If you are installing a new system, follow the instructions in the software installation booklet appropriate to your configuration. Currently, booklets are provided for 16 installation scenarios:

- Installing VAX/VMS on a dual-RL02 system

- Installing VAX/VMS on a VAX–11/725

- Installing VAX/VMS on a VAX–11/730 configured for R80 /RL02 operations

- Installing VAX/VMS on a UDA-based VAX–11/730 from an RA60 distribution disk

- Installing VAX/VMS on a UDA-based VAX–11/730 from magnetic tape

- Installing VAX/VMS on a VAX–11/750 from an RL02 distribution kit

- Installing VAX/VMS on a VAX–11/750 from an RK07 or RA60 distribution disk

- Installing VAX/VMS on a VAX–11/750 configured for dual-RK07 operation

- Installing VAX/VMS on a VAX–11/750 from magnetic tape

- Installing VAX/VMS on a VAX–11/780 from an RK07 or RA60 distribution disk

- Installing VAX/VMS on a VAX–11/780 configured for dual-RK07 operation

- Installing VAX/VMS on a VAX–11/780 from magnetic tape

- Installing VAX/VMS on a VAX–11/750 from an HSC-based disk

- Installing VAX/VMS on a VAX–11/750 from an HSC-based tape

- Installing VAX/VMS on a VAX–11/780 from an HSC-based disk

- Installing VAX/VMS on a VAX–11/780 from an HSC-based tape

Installation procedures for each of these scenarios are described in 16 booklets, which are shipped in your distribution kit. Each booklet defines the installation environment and includes key messages generated by the program.

In most cases, variable information (that is, dates and times and so forth) is included in displays for illustration purposes only. You should substitute values appropriate to your installation. Note too, that time intervals specified for various events may vary from installation to installation.

Other variables reflect the many hardware configurations that may be encountered in an installation. These include device name variables (load device and system device) and boot name variables. See Chapter 4 for help with handling these variables.

1.1 Summary of a New Installation

A newly purchased operating system can be installed by anyone authorized to do it.

The following is a summary of how a newly purchased operating system is installed:

1 Boot stand-alone BACKUP from the distribution kit into the processor.

2 Using stand-alone BACKUP, restore the *required* save set to get a partial, but bootable, system.

3 Boot the partially restored system.

4 Restore the rest of the operating system.

5 Boot the restored system.

6 Run the User Environment Test Package (UETP).

1.2 Summary of a System Upgrade

Upgrades are installed on systems to bring them up to the latest major revision from the most recent maintenance version. For example, if you want to bring a Version 3.5 system up to Version 4.0, you would install the Version 4.0 upgrade kit. To upgrade a system, it must be in an upgradeable condition; that is, the system files must be substantially the same as when the system was installed.

A system upgrade is performed in nine steps:

1 Back up your current system disk to the disk that will be used for the upgrade.

2 Boot the new system volume and log in to the system manager's account.

3 If you want to keep any files that may be affected by the upgrade, move them to a user directory.

4 Be sure you have enough space on the new system volume to do the upgrade.

5 Be sure the CPU is set up for automatic restart.

6 Prevent users from logging into the system, stop all queues, and, if applicable, disconnect the system from DECnet.

7 Invoke the VMSINSTAL command procedure and follow instructions that are displayed through the five upgrade phases.

8 If any of the system reboots fail, change system parameters as directed.

9 When the upgrade is completed, boot the new system and make any required changes in system procdures before resuming normal operations.

1.3 Summary of Maintenance Update and Optional Product Installations

When you install a maintenance update, you must refer to the accompanying release notes for directions. Similarly, when you install an optional software product you should refer to the installation documentation that is included with the product.

You use the VMSINSTAL command procedure to install all maintenance updates and some optional software products. A few of the optional software products are installed using the VMSUPDATE command procedure.

Both procedures require very little interactive input. You need only prepare the system for the installation, and then physically manipulate the media as directed by prompting messages during the installation.

Below is a summary of how to install maintenance updates and optional products:

1 Log in under the SYSTEM account.

2 Back up your system disk.

3 Keep users off the system.

4 Set your default to point to the disk receiving the software.

5 Invoke the installation command procedure.

6 Respond to prompts from the command procedure.

7 Shut down the system and then reboot with the new product(s) installed.

2 Console Subsystems

During the installation, you use the console subsystem to communicate with the central processing unit (CPU). Each VAX console subsystem includes the following:

- Indicators and controls on the processor control panel

- A console terminal

- One or two block storage devices (depending on the type of processor you have)

- The optional remote diagnostic port (not usually used to install software)

- A console command language

Because the processor control panels vary from processor to processor, particularly the VAX–11/750, this chapter discusses each VAX system separately.

The console terminal, however, is the same for all VAX processors so a common description is given.

The block storage devices vary somewhat, but the console command language is the same for all VAX processors.

VAX console subsystems run in two mutually exclusive modes: program mode and console mode. When power is applied, the subsystem is in one of these modes. If the CPU is running, the console subsystem is in program mode, but if the CPU is halted, the console subsystem is in console mode.

In program mode, the console terminal is used like other terminals on your system. During an installation, you go to program mode to use the stand-alone Backup Utility (BACKUP) to enter DCL commands and to receive appropriate installation messages and prompts.

In console mode, you have access to the console command language and are given a special console prompt ($>>>$). If you have a VAX–11/725, VAX–11/730, or VAX–11/750, you put the system into console mode by pressing CTRL/P. If you have a VAX–11/780, however, you must first press CTRL/P and then enter the HALT console command. For most installations, the console command you will use most often is BOOT. In some cases, you may also use the HALT command and the DEPOSIT command.

You use the BOOT command to load software into the CPU; the stand-alone Backup Utility is loaded first and then the restored components of the operating system.

When HSC disks are used to install a new system, the DEPOSIT command is used to prepare the general registers for booting the HSC disks.

2.1 Processor Control Panels

This section describes the processor control panels for the various VAX processors.

2.1.1 VAX–11/780 Processor Control Panel

The VAX–11/780 processor control panel consists of four indicator lights, an AUTO RESTART switch, a BOOT switch, and a 5-position keylock switch. The indicator lights reflect the current state of the system, and the switches dictate how the system will react to initial bootstrapping, shut downs, and restarts. Figure 2–1 shows the VAX–11/780 processor control panel.

Figure 2–1 The VAX–11/780 Processor Control Panel
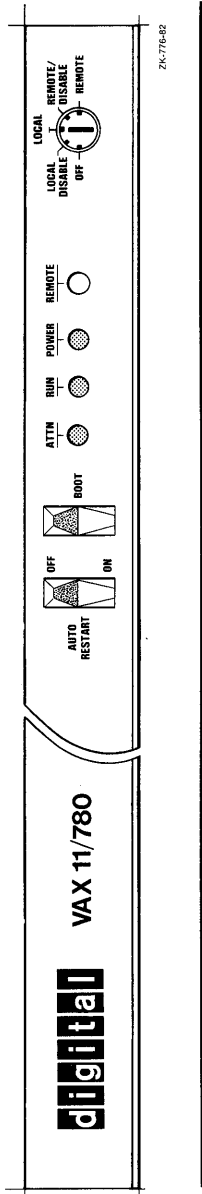
2.1.1.1 Indicator Lights

There are four indicator lights on the processor control panel. The first three indicate the operational state of the CPU. The fourth indicates the state of remote diagnostic procedures (not usually applicable to software installation).

ATTN   When lit, indicates that you have the attention of the console program.

RUN   When lit, indicates that the processor is running.

POWER   When lit, indicates that power is present in the processor and that the keylock switch is not in the OFF position.

REMOTE When lit, indicates that remote diagnostic procedures are being performed on the system.

2.1.1.2 Switches

The AUTO RESTART switch controls the processor's power-up sequence. This switch has two positions:

ON   Restarts the system automatically, even when unattended.

OFF   Halts the system and displays the console prompt ($>>>$) at the console terminal.

The BOOT switch bootstraps the system using the default bootstrap command

The keylock switch lets you control system power from the front of the CPU cabinet, and lets you establish the operating mode as described below. Power is applied to the system when the keylock switch is in any position but OFF. During installation, you set the switch to LOCAL.

OFF   No power to the CPU (except battery backup to the time-of-year clock) or to memory.

LOCAL DISABLE   Local console terminal commands are ignored.

LOCAL   Processor responds to local console commands and the remote line is disabled. This position is used for installing the operating system software.

REMOTE DISABLE Processor cannot respond to commands from remote line terminals.

REMOTE   Processor responds to commands from
                  a remote line terminal but local console
                  commands are ignored.

2.1.2   VAX–11/750 Processor Control Panel

The VAX–11/750 processor control panel includes seven
indicator lights, four switches, and a TU58 cartridge drive.
The indicator lights reflect the current status of the system,
and the switches dictate how the system will react to initial
bootstrapping, shutdowns, and restarts. Figure 2–2 shows a
diagram of a VAX–11/750 processor control panel.

Figure 2-2 The VAX-11/750 Processor Control Panel
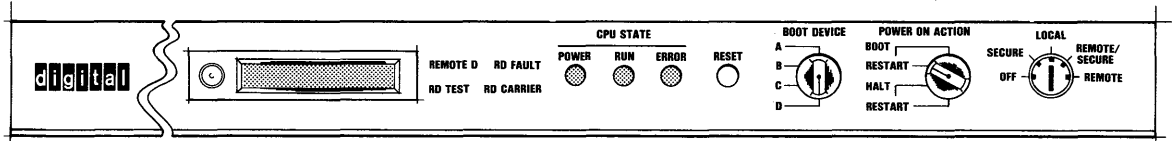
ZK-995-82

2.1.2.1   Indicator Lights

There are seven indicator lights on the processor front panel: the first three indicate the state of the processor, and the remaining four indicate the state of the remote diagnostic procedures.

The CPU status indicators are described in the following list:

POWER   When lit, indicates that the power is present in the processor and that the keylock switch is not in the OFF position.

RUN   When lit, indicates that the processor is running.

ERROR   When dimly lit, indicates the processor is running normally. When brightly lit, indicates that the processor has experienced an unrecoverable error. If this occurs, push the white INITIALIZE button to reset the machine. (The ERROR light will remain dimly lit.) If the light is not lit during normal operation, the bulb is probably defective.

The remote diagnostic indicator lights appear only on processors that have installed a remote diagnostic ROM, and are visible only when lit.

REMOTE D   When lit, indicates that the keylock switch is in the REMOTE position.

RD FAULT   When lit for more than 10 seconds, indicates that the remote diagnostic ROM has detected an error. When a processor configured with a remote diagnostic ROM is powered up, this light will come on for about 10 seconds to indicate that a self test is in progress.

RD TEST   When lit, indicates that the remote host connection has been made with successful protocol exchange.

RD CARRIER   When lit, indicates that a carrier signal is present on a dial-up line.

2.1.2.2   Switches

The POWER ON ACTION switch controls the processor's power-up sequence. The processor interprets and acts on the position of the POWER ON ACTION switch in the following situations:

• After an orderly shut down

• After a brief power failure

2–7

- After the microcode detects a halt condition (for example, a HALT instruction executed in kernel mode)

Each position of the switch provides the processor with instructions on how to power up the system for different situations.

BOOT	Bootstraps the system from the device selected by the BOOT DEVICE switch.
RESTART (BOOT)	Restarts the system. A bootstrap is executed from the device selected by the BOOT DEVICE switch if the restart fails.
HALT	Halts the system and displays the console prompt (> > >) at the console terminal.
RESTART (HALT)	Restarts the system. The system halts if the restart fails.

During normal operations, the POWER ON ACTION switch should be set to RESTART (BOOT) for the most comprehensive system recovery procedure.

The BOOT DEVICE switch is used to select the system's bootstrap device. The processor interprets and acts on the position of the BOOT DEVICE switch in the following situations:

- When you issue the console BOOT command without specifying a device name

- When the system attempts an automatic bootstrap

In either situation, the processor examines the BOOT DEVICE switch and assumes that the selected bootstrap device is the first unit on the first controller, that is, unit number 0 on controller A.

The BOOT DEVICE switch has four positions: A, B, C, and D. Position A always causes the system to bootstrap from the TU58 cartridge drive. Positions B, C, and D correspond to the remaining disk controllers. Consult your DIGITAL field service representative to determine how the positions correspond to the available disk controllers. Typically, you should set the BOOT DEVICE switch to the position corresponding to your system disk.

The RESET button (in earlier models labeled INITIALIZE) provides for an automatic system shutdown followed by a power-up sequence without disturbing the contents of memory. The system determines the power-up action by examining the position of the POWER ON ACTION switch.

The 5-position keylock switch gives you ready access to control over system power, and lets you establish the various operating modes. During installation, you set the switch to LOCAL. Power is applied to the system when the keylock switch is in any position except OFF.

OFF	No power to the CPU (except battery backup to the time-of-year clock) or to memory.
SECURE	Console commands are ignored, the remote diagnostic line is disabled, and the RESET switch is disabled.
LOCAL	Processor acts on console commands, but the remote diagnostic line is disabled.
REMOTE/SECURE	Processor responds to commands from the remote diagnostic line, but local console terminal commands are ignored and the RESET switch is disabled. The remote diagnostic module cannot be used for fault isolation in this setting.
REMOTE	Processor responds to commands from the remote diagnostic line but local console terminal commands are ignored; however, the remote line can be used to restore control to the local console terminal.

2.1.3   VAX–11/730 and VAX–11/725 Processor Control Panels

The processor control panel is nearly identical on the VAX–11 /725 and the VAX–11/730 processors. The only difference is that there is no BATT (battery) indicator for the VAX–11/725 because the system does not include a battery backup feature.

The panel includes several indicator lights, an AUTO RESTART /BOOT switch, and a 6-position keylock switch. The indicator lights reflect the current state of the system and the switches dictate how the system will react to initial bootstrapping, shutdowns, and restarts.

Figure 2–3 presents a diagram of the processor control panel.

Figure 2-3   VAX-11/730 and VAX-11/725 Processor Control Panel
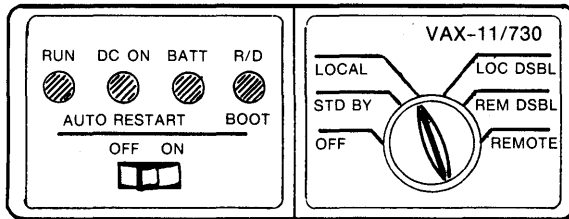
Battery Backup Not Included on VAX-11/725

ZK-1002-82

2.1.3.1   **Indicator Lights**

The four processor control panel indicator lights and their respective functions are described below.

RUN   When lit, this light indicates that the processor is running in program mode.

DC ON   When lit, this light indicates that DC power is present in the processor, and that the keylock switch is not in the OFF position.

BATT   When lit steadily, light indicates that the backup battery is at greater than 90% of full power. (This indicator is not included on the VAX-11/725 processor control panel.)

When flashing slowly (less than once every second), this light indicates that the battery is at less than 90% of full power and is charging.

When flashing quickly, this light indicates that the battery is at less than 90% of full power and is discharging.

When the light is out, the battery is dead or not present.

R/D   When lit, this light indicates that remote diagnostic procedures are being performed on the system.

2.1.3.2   **Switches**
This section describes the AUTO RESTART/BOOT switch and
the keylock switch.

The AUTO RESTART/BOOT switch has two functions: it
instructs the processor what to do when halt conditions occur,
and it bootstraps the system. The switch has three positions:
ON, OFF, and BOOT. The first two positions, ON and OFF,
control the action of the processor in the following situations:

- During initial power-up

- After a momentary power failure

- When the microprogram detects an error halt condition (for
  example, a HALT instruction executed in kernel mode)

If the switch is in the ON position when these conditions occur,
the system restarts automatically. If the switch is in the OFF
position, a console prompt ($>>>$) appears at the console
terminal and no action is taken by the processor.

The BOOT position is spring loaded. When the switch is
momentarily pushed to the BOOT position, it bootstraps the
system, using the default bootstrap command procedure.

During installation, the AUTO RESTART switch must be set to
the OFF position. During normal operation, the switch should
be left in the ON position. This position permits the most
comprehensive system recovery procedure.

The keylock switch lets you control system power from the
front of the CPU cabinet and establish various operating modes.
Power is applied to the system when the keylock switch is in
any position except OFF. During installation, you set the switch
to LOCAL. Each of the positions is described below:

OFF	Power is off.
STD BY	Power is applied only to the WCS module and main memory.
LOCAL	Processor can respond to console commands from the local console, but the remote diagnostics line is disabled.

LOC DSBL   Processor cannot respond to console commands, but the console terminal can be used as a standard terminal. The remote line is disabled. The BOOT position of the AUTO RESTART/BOOT switch is ignored.

REM DSBL   Processor cannot respond to console commands from remote line terminals, but the remote terminal is usable as a standard terminal. The local console is disabled.

REMOTE   Processor responds to console commands from a remote line terminal, but the local console terminal is disabled.

2.2 Console Terminal

The console terminal is a hard-copy terminal that you use as a control station during the installation. You apply power to the console terminal by pushing the console power switch to the ON position.

You use the terminal to input all commands and to read all output messages and prompts during the installation. Be sure you have an adequate supply of paper in the terminal printer before you begin the installation.

You put the system into console mode by pressing CTRL/P on the console terminal keyboard to halt the CPU. The system indicates that it is in console mode by printing the console program prompt (> > >) on the terminal printer. When a bootstrap function is completed, the console program returns the system to program mode automatically.

2.3 Block Storage Devices

The block storage device is the console subsystem component that provides storage for the console program. Block storage devices may also be used to install updates and optional products on a system. If you install or upgrade your system using a magnetic tape distribution kit, stand-alone BACKUP is distributed on block storage media. When you bootstrap your system, the block storage device provides a command procedure that bootstraps the operating system from the system device to the CPU.

Table 2–1 lists the block storage devices for the various VAX processors.

Table 2–1   Block Storage Devices

VAX Processor	Block Storage Device
VAX–11/780	RX01
VAX–11/782	RX01
VAX–11/785	RX01
VAX–11/750	TU58
VAX–11/730	TU58[1]
VAX–11/725	TU58[1]

[1]These systems feature two TU58 drives.

2.3.1   Block Storage on the VAX–11/780

The VAX–11/780 block storage device is an RX01 floppy diskette drive assigned the logical device name CSA1. To insert a floppy diskette into CSA1, proceed as follows while referring to the photographs for each instruction.
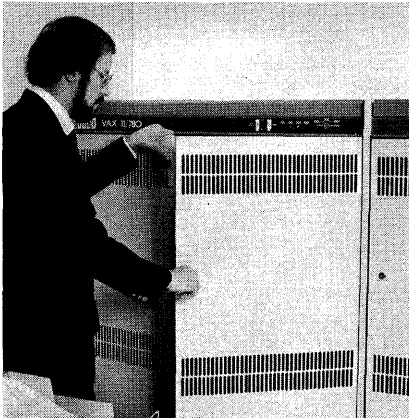
1

2

1 Unlatch and open the cabinet doors of the CPU cabinet.

2 Swing out the floppy drive assembly until it is at a right angle to the cabinet.

The drive assembly is a rectangular, unpainted steel box in the lower right-hand corner of the central processor cabinet. There is a black handle on the right of the drive assembly. Pull the handle to swing out the drive assembly until it is perpendicular to the cabinet. If you do not swing it out far enough, or if you swing it out too far, you will not be able to insert the floppy diskette.

3

4
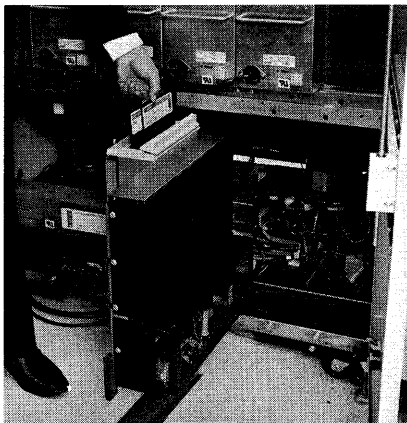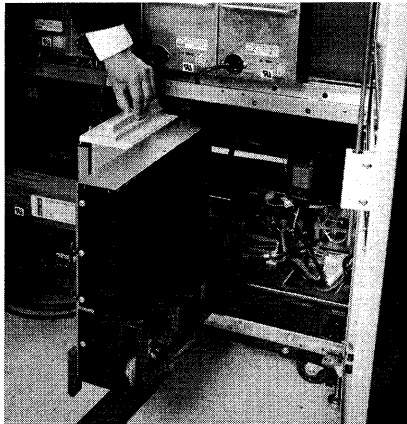
3 Press the black push button to unlock the slot cover. The cover will spring open.

4 Insert the console floppy in the drive. Its label should be at the top and should face the right-hand cabinet door. The oval slot on the diskette should be at the bottom.
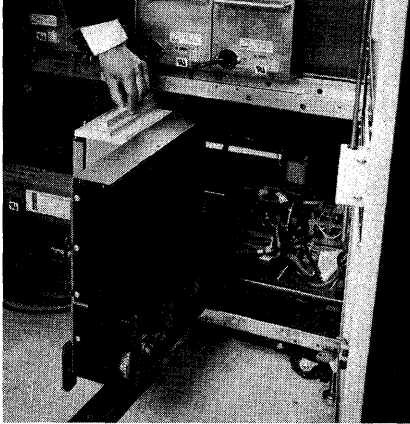
5
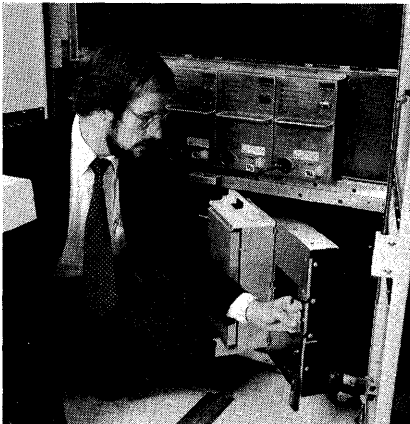
6

5 Close the diskette slot cover.

6 Swing the drive assembly back into the cabinet.
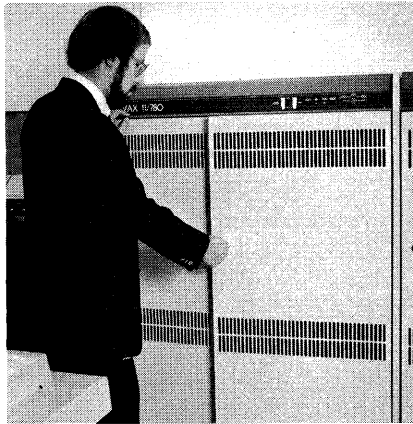
7 Close and latch the door on the cabinet.

To remove a floppy from the drive, follow the same procedure to gain access to the diskette, then lift it firmly but gently out of the drive, taking care not to touch any exposed platter surfaces.

2.3.1.1   Handling TU58 Cartridges
The block storage devices for all VAX processors except the VAX–11/780 family are TU58 tape cassette. To insert a cassette into a TU58 drive, proceed as follows while referring to the photograph for each instruction.

1
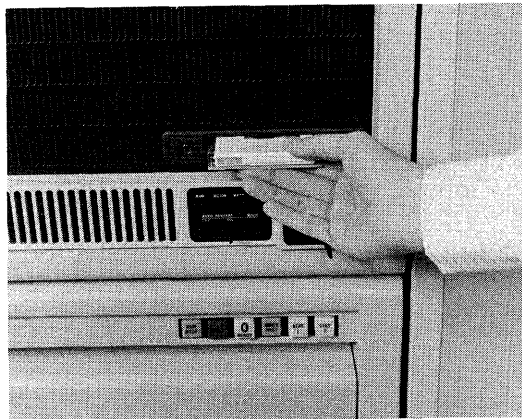
2

1 Locate the slotted opening at the front of the drive.

2 If a cassette is already in the drive, remove the cassette by gripping it firmly with your fingertips and pulling it gently towards you.
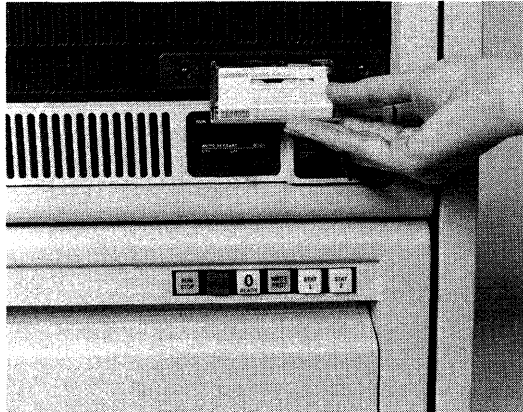
3
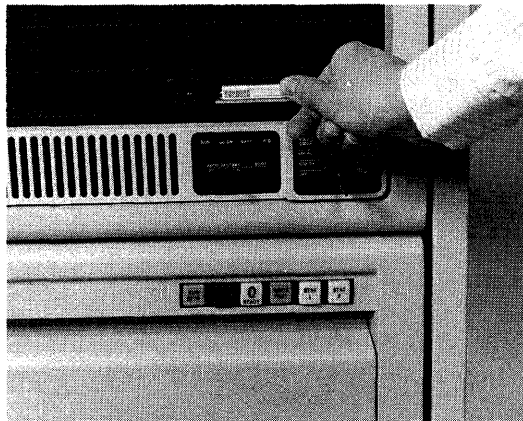
4

3 Insert a TU58 with the label facing up and the RECORD tab
facing forward.

4 Gently push the cassette into the slot until it seats. When
fully inserted, the cassette protrudes from the drive about
one-half inch.

2.3.1.2   **Block Storage on the VAX–11/750**

The VAX–11/750 uses a single TU58 cassette drive as its block storage device. The primary purpose of the TU58 is to give you the option of using a special stand-alone bootstrap facility called BOOT58 if you cannot boot from the system disk.

The recommended method for daily operations is to boot from the system disk itself by positioning the BOOT DEVICE switch to the appropriate setting and then entering the BOOT command without specifying a boot command procedure. If you are not sure of the correct setting, see your DIGITAL field service representative.

If you do not have a position on the BOOT DEVICE switch for the system device, or if you want to boot from a system root other than SYS0, invoke the BOOT58 program by setting the BOOT DEVICE switch to position A and then entering the BOOT command. When you enter the BOOT command, the BOOT58 program boots, and lets you select the appropriate boot command procedure. (See the *Guide to VAX/VMS System Management and Daily Operations* for more information on bootstrapping VAX/VMS on a VAX–11/750.)

You can also invoke BOOT58 facility by entering the following console command:

```
>>>B/800 DDA0
```

In addition to providing bootstrap functions, the TU58 drive is used for installing maintenance updates and optional software products, and for booting stand-alone BACKUP on magnetic tape system installations.

2.3.1.3   **Block Storage on the VAX–11/730**

There are currently two VAX–11/730 system controller configurations: the Integrated Disk Controller (IDC) configuration, and the UNIBUS Disk Adapter (UDA) configuration. Although both configurations use a dual-TU58 block storage subsystem, the physical locations of the TU58 drives are different in each configuration.

The two TU58 cartridge drives are assigned the device names CSA1 and CSA2, respectively.

CSA1 is used to install updates and optional products and to boot stand-alone BACKUP from magnetic tape distribution kits. CSA2 is used to boot the console program from the console TU58.

Figure 2–4 shows the locations of the TU58 drives on an IDC-based VAX–11/730. CSA1 is located at the front of the CPU cabinet above the processor control panel. The other TU58 drive, CSA2, is accessible only by sliding the CPU out from the cabinet, as shown.

Caution: **If you have to slide out the CPU, be sure to extend the safety foot located under the cabinet first.**

Figure 2–4   TU58 Locations on IDC-Configured VAX–11/730

CSA2

Safety Foot (extended)

SIDE VIEW

Internal Cassette Drive (CSA2)

Front Cassette Drive (CSA1)

TOP VIEW
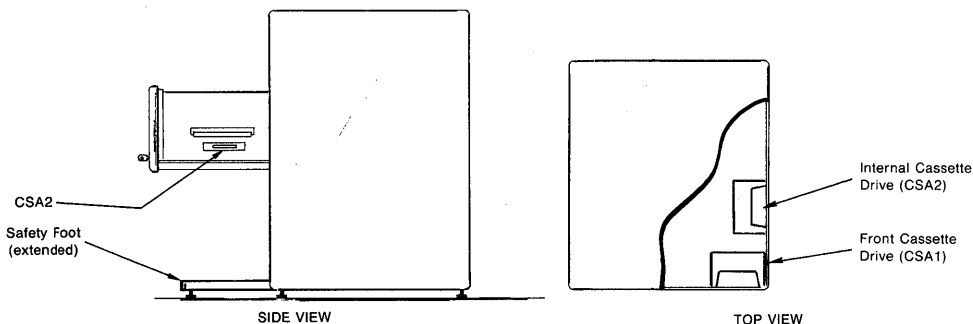
ZK-1000-82

There is a LED indicator light on either side of the CSA1 drive. When lit, the left-hand LED indicates that the front drive (CSA1) is active. When the right-hand LED is lit, the internal drive (CSA2) is active.

On the UDA-configured VAX–11/730 processors, both TU58 drives are located at the front of the CPU, and both are readily accessible. CSA1 is located on the left side of the front panel and is labeled **0**. CSA2 is located on the right side of the front panel and is labeled **1**. Figure 2–5 shows the locations of the TU58 drives on UDA-configured VAX–11/730 systems.

Figure 2–5 TU58 Locations on UDA-Configured VAX–11/730
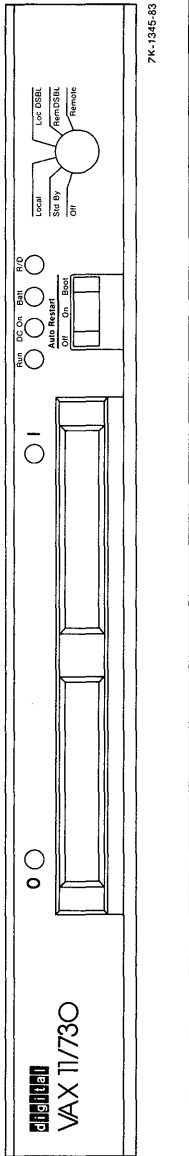
2.3.1.4   **Block Storage on the VAX–11/725**
On the VAX–11/725 system, CSA1 is located at the front of
the CPU cabinet, and CSA2 is located at the right side of the
cabinet. The difference between the VAX–11/725 and the
IDC-configured VAX–11/730 is that CSA2 is readily accessible
through an opening in the side of the cabinet.

Activity indicators, labeled **0** for CSA1 and **1** for CSA2, are
located at each side of CSA1. Figure 2–6 shows the location of
the TU58 drives on the VAX–11/725.

Figure 2-6 TU58 Locations on VAX-11/725
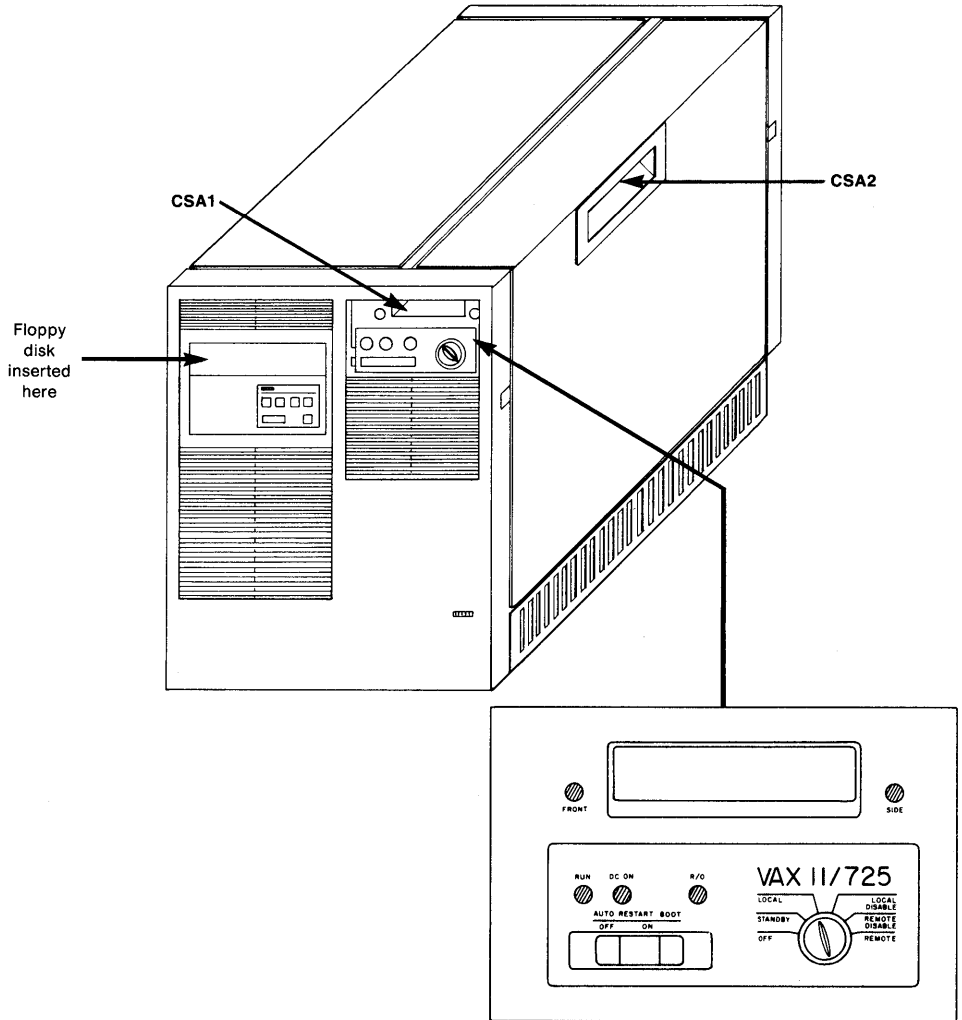
ZK-1272-83

3 Controls and Indicators for Peripheral Devices

This chapter covers operating information related to the peripheral devices you use when installing software on a VAX processor. It describes how to use the operating controls and indicators for the various drives, and how to handle the various types of storage media.

The peripheral devices are magnetic tape drives and disk drives that may be configured in various system hardware configurations.

- RL02 disk drive

- RK07 disk drive

- RP05, RP06 disk drives

- RP07 disk drive

- RM03, RM05 disk drives

- R80, RM80 disk drives

- RA60, RA80, RA81 disk drives

- RC25 disk drive

- TU78 tape drive

- TE16, TU45, TU77 tape drives

- TS11 tape drive

Each device is described as either a load (source) device or a system (target) device. During an installation, the load device is the storage system that contains the distribution media, and the system device is the storage system that contains the new system volume.

3.1 RL02 Disk Drive

The RL02 disk drive is used with VAX–11/730 systems as either a load device, a system device or, in a dual-RL02 configuration, as both. With VAX–11/750 systems, the RL02 is used solely as a load device.

In VAX–11/730 system configurations, the RL02 connects to the UNIBUS by means of an integrated disk controller (IDC). In non-IDC configurations, it uses the RL211 controller.

The RL02 provides four front panel controls and indicators (see Figure 3–1.)

• LOAD switch and indicator

• Logical Address Receptacle and READY indicator

• FAULT indicator

• WRITE PROT (write protect) switch and indicator

Figure 3–1   RL02 Front Panel Controls and Indicators
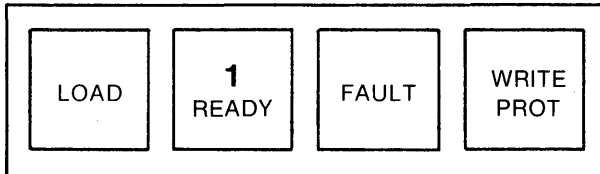
ZK-1950-84

The **LOAD switch and indicator** combines an alternate-action switch with a lamp that indicates when it is safe to load (or unload) the disk from the drive. When the drive is at rest, the LOAD indicator is lit, and you may either load or unload the disk as appropriate. When you insert a disk into the drive and press the LOAD switch, the indicator immediately turns off and the drive begins to accelerate to operating speed. When the drive reaches operating speed, the READY indicator turns on.

Before you can remove the disk from the drive, you must bring the drive to rest. If the LOAD indicator is not lit, the drive is at operating speed, so you must press the LOAD switch to bring it to rest. When the drive comes to rest, the LOAD lamp turns on, and you may remove the disk.

The **logical address receptacle** lets you select a unit address for each drive by simply inserting the appropriate logical address plug. With each drive, you receive four plugs (numbered 0 through 3) from which you may select the desired unit address.

Each plug is a mechanical device that actuates a set of address switches corresponding to the unit address numeral inscribed on the plug face. For example, if the plug is inscribed with the numeral **0**, it will actuate a set of switches that assign the unit address 0 to the drive.

Each logical address plug incorporates a **READY indicator** that turns on when the drive is at operating speed. The indicator remains on until you press the LOAD switch to begin applying braking action to the drive spindle.

The **FAULT indicator** turns on when a hardware fault is detected. For example, if a loss of write current is detected, the FAULT lamp will turn on and will remain on until the faulty condition is corrected. If a fault condition persists, notify maintenance personnel.

The **WRITE PROT switch** (write protect) is an alternate-action switch that protects you from inadvertently overwriting the distribution disk. When the switch is activated, the indicator turns on and the disk is write protected. Conversely, if the switch is deactivated, the indicator turns off and the processor is allowed to write to the disk.

During installation, you are usually instructed to write-protect the distribution disk.

3.2 RK07 Disk Drive

The RK07 is a medium-capacity disk that is used to distribute software for VAX–11/750 and VAX–11/780 systems. The RK07 connects to the UNIBUS by means of the controller, and it provides the controller with static, dual-port access to the drive.

The following controls and indicators are located on the RK07 front panel:

- RUN/STOP switch and indicator

- Logical Address Plug and READY indicator

- FAULT indicator

- WRITE PROT (write protect) switch and indicator

- Port Selection controls (the A selector switch and indicator and the B selector switch and indicator)

The RK07 front panel is shown in Figure 3–2.

Figure 3–2   RK07 Front Panel Controls and Indicators

| RUN STOP | 1 READY | FAULT | WRITE PROT | A | B |
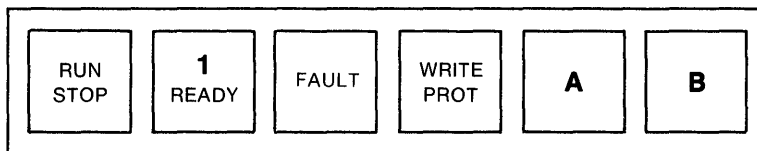
ZK-1952-84

The **RUN/STOP switch** is an alternate-action switch that you use to either bring the disk drive up to operating speed (RUN active) or bring the drive to rest (STOP active). When the indicator is off, the drive is at rest. To bring the drive up to speed, press the switch once. The indicator turns on and the drive begins to accelerate to operating speed. When operating speed is reached, the READY indicator lights.

To bring the drive to rest when it is at operating speed, press the switch once. The READY indicator will turn off immediately and the drive will begin to decelerate. The RUN /STOP indicator turns off when the drive stops.

The **logical address receptacle** lets you select a unit address by simply inserting the appropriate logical address plug. With each drive, you receive eight plugs (numbered 0 through 7) from which you may select the desired unit address.

Each plug is a mechanical device that actuates a set of address switches corresponding to the unit address numeral inscribed on the plug face. For example, if the plug is inscribed with the numeral **0**, it will actuate a set of switches that assign the unit address 0 to the drive.

Each logical address plug incorporates a **READY indicator** that turns on when the drive is at operating speed and remains on until you press the RUN/STOP switch to begin applying braking action to the drive spindle.

The **FAULT indicator** turns on when a hardware fault is detected. For example, if a loss of write current is detected, the FAULT indicator turns on and remains on until the faulty condition is corrected. See your DIGITAL field service representative for instructions on correcting fault conditions.

The **WRITE PROT switch** (write protect) is an alternate-action switch that protects you from inadvertantly overwriting the distribution disk. When the switch is activated, the indicator turns on and the disk is write protected. Conversely, if the switch is deactivated, the indicator turns off, and the processor is allowed to write to the disk.

During installation, you are usually instructed to write protect the distribution disk.

The controller may access the RK07 disk drive through I/O port A, I/O port B, or both ports, depending on the setting of the I/O port selector switches on the drive front panel. You use the **A selector switch** to enable access through port A and the **B selector switch** to enable acess through port B. The **A** indicator lights when port A is being used, and the **B** indicator lights when port B is being used.

3.3 RP05 and RP06 Disk Drives

The RP05 and RP06 are MASSBUS devices that connect to
the processor by means of the MASSBUS controller. They are
large-capacity storage systems; the RP05 has a capacity of 100
megabytes, and the RP06 has a capacity of 200 megabytes. Port
selection gives the controller static, dual-port access to each of
these drives.

The front panel controls and indicators for the RP05 and RP06
include the following:

* One logical address plug

* Four switches

* Seven indicators

Figure 3–3 shows the locations of the controls and indicators.

Figure 3–3   RP05 and RP06 Front Panel Controls and Indicators


```
                                                    WRITE
                                                   PROTECT
                                                      |    CONTROL
                                              START   |      A

                                              STOP        CONTROL
                                                             B

    START          STANDBY          WRITE PROTECT
  CONTROL A B
    READY                           DOOR LOCKED            2
    UNSAFE

                                 NOTE:
                                    LOGICAL ADDRESS PLUGS ARE REMOVED BY PULLING STRAIGHT
         LAMP TEST SWITCH            OUT  WHILE DISENGAGING (677-51  DEC DRIVES ONLY) A SPRING
                                     CLIP AT THE REAR OF THE PLUG.

                                                                      ZK-1959-84
```
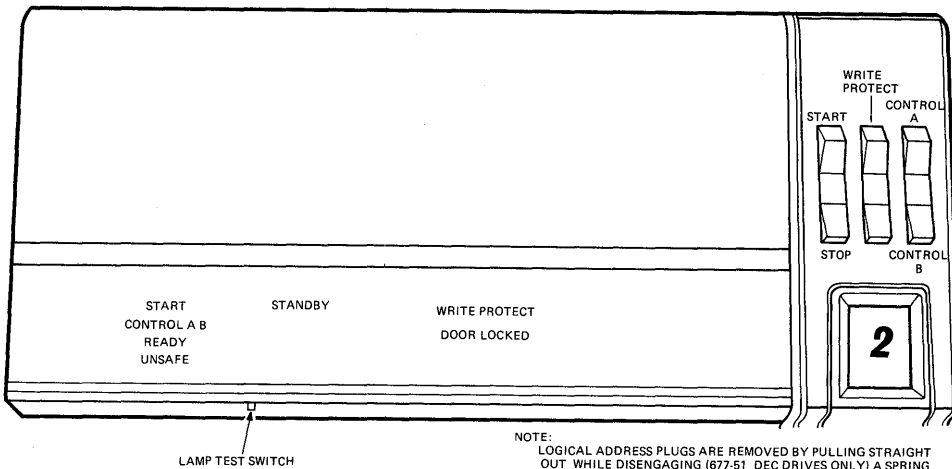
Three of the switches are rocker switches and the other is
a pushbutton switch. The indicators provide short status
messages and are visible only when active.

The **logical address receptacle** lets you select a unit address for each drive by simply inserting the appropriate logical address plug. With each drive, you receive eight plugs (numbered 0 through 7) from which you may select the desired unit address, and a service plug with the letter **S** inscribed on its face.

Each address plug is a mechanical device that actuates a set of address switches corresponding to the unit address numeral inscribed on the plug face. For example, if the plug is inscribed with the numeral **0**, it will actuate a set of switches that assign the unit address 0 to the drive. If you have multiple drives, be sure that you do not use identically numbered plugs on drives assigned to the same controller.

The **START/STOP switch** is a 3-position toggle switch: START, STOP, and an unlabeled center position.

If all interlocks are satisfied and the access door is closed (DOOR LOCKED indicator lit), you begin a power-up sequence by momentarily pressing the switch to START. The START indicator turns on and the STANDBY light turns off (assuming it was lit when you presses the switch). As soon as you release the switch, it returns to the center position, but the START indicator remains on until you stop the drive. When the drive reaches operating speed, the heads extend and read/write operations may begin.

To stop the drive, press the switch to the STOP setting (it will remain there). The heads retract and the drive begins to decelerate. When the drive stops, the DOOR LOCKED indicator turns off and you may safely remove the disk pack.

The **WRITE PROTECT switch** is a 2-position rocker switch that protects you from inadvertantly overwriting the data on the disk when doing read-only operations. When you set the switch to the WRITE PROTECT position, the disk is protected and the WRITE PROTECT indicator turns on. When you press it to the unlabeled "off" position, protection is removed and the WRITE PROTECT indicator turns off.

The **CONTROL A/CONTROL B switch** is a 3-position rocker switch that determines whether the controller accesses the drive through port A, port B, or through both ports. When you set the switch to CONTROL A, the controller accesses the drive through port A, and the CONTROL A indicator turns on. When you set the switch to CONTROL B, the controller accesses the drive through port B, and the CONTROL B indicator turns on.

If you position the switch to the unlabeled center position, the controller may access the drive through either port and both the CONTROL A and CONTROL B indicators will turn on.

The **lamp test switch** is an unlabeled pushbutton switch used to verify the status of the front panel indicators. Replace any indicator that fails to turn on when you press the switch.

The **START indicator** turns on when you press the START /STOP switch to the START position, and it remains on until you stop the drive.

The **CONTROL A/B indicators** indicate which port(s) is being used by the controller.

The **READY indicator** indicates the drive is on line.

The **UNSAFE indicator** indicates there is a drive malfunction that requires attention.

The **STANDBY indicator** indicates the drive is in STANDBY mode ready to be powered up. As soon as the drive is powered up, the STANDBY indicator turns off and the START indicator turns on.

The **WRITE PROTECT indicator** indicates the disk is write protected.

The **DOOR LOCKED indicator** indicates the drive access door is locked and the disk is not accessible.

3.4 RP07 Disk Drive

The RP07 disk drive implements fixed-head technology incorporating a sealed-head-disk assembly to provide fast access and 516 megabytes of storage. It connects to the MASSBUS by means of the MASSBUS controller and provides the controller with static dual access to the drive.

The front panel controls and indicators for the RP07 include four toggle switches and six indicators:

*   START/STOP switch

*   ON-LINE switch (not labeled) and labeled indicator

*   WRITE PROTECT switch (not labeled) and labeled indicator

3-8

- ACCESS A, A/B, ACCESS B switch, and associated indicators

- UNSAFE indicator

Figure 3–4 shows the locations of the controls and indicators.

Figure 3–4   RP07 Front Panel Controls and Indicators
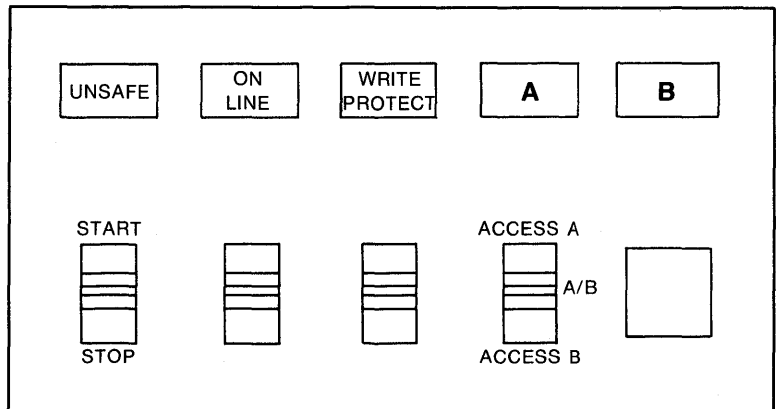
ZK-1954-84

The **START/STOP switch** is a 2-position toggle switch used to bring the head-disk assembly up to speed by setting the switch to the START position. When you set the switch to the STOP position, a braking action is applied to the head-disk assembly.

The **ON-LINE switch** (unlabeled) is a 2-position toggle switch used to put the drive on line, that is, to make the drive available to the processor. In the up position, the drive is on line, and the ON-LINE indicator turns on. Conversely, the ON-LINE indicator turns off and the drive goes off line when the switch is in the down position.

The **WRITE PROTECT switch** (unlabeled) is a 2-position toggle switch used to prevent inadvertant overwriting of the disk. When the switch is in the up position, the WRITE PROTECT indicator turns on and the media is protected. When the switch is set to the down position, the indicator turns off and the processor may write to the media.

3–9

The **ACCESS A, A/B, ACCESS B switch** is a 3-position toggle switch the controller uses to access the drive. In the ACCESS A position, the A indicator turns on and the controller accesses the drive through port A. In the ACCESS B position, the B indicator turns on and the controller accesses the drive through port B. In the A/B position, both indicators turn on and the controller can access the drive through both ports.

The **UNSAFE indicator** flashes when the drive logic senses a potential problem such as an inordinate rise in temperature or insufficient air flow. If a problem needs immediate attention, for example if a start-spindle error occurs, the indicator lights steadily.

3.5   RM03 and RM05 Disk Drives

The RM03 and RM05 disk drives are MASSBUS devices that connect to the processor by means of the MASSBUS controller. The RM03 has a 67-megabyte storage capacity. The RM05 has a storage capacity of 256 megabytes. Optional port selection gives the controller static, dual-port access to each of these drives.

The front panel controls and indicators for the RM03 and RM05 include:

- START switch and associated indicator

- Unlabeled Drive Number receptacle

- READY indicator

- FAULT CLEAR switch and associated indicator

- WRITE PROTECT switch and associated indicator

You may also choose a pair of port selection switches that mount on the drive cabinet door.

Figure 3–5 shows the locations of the controls and indicators.

Figure 3–5   RM03/RM05 Front Panel Controls and Indicators
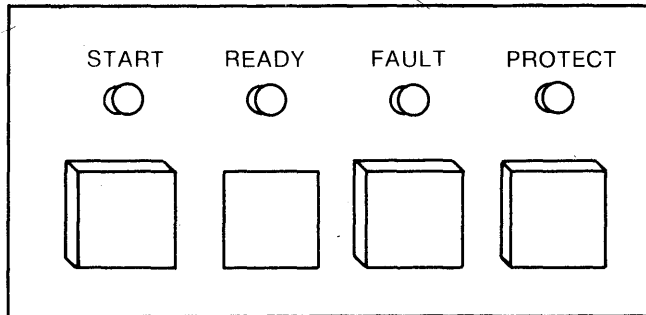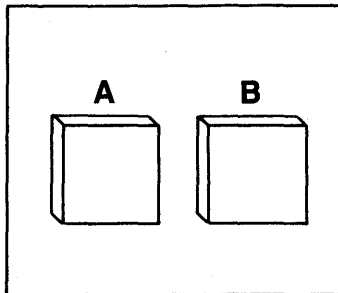
ZK-1953-84

The **START switch** is an alternate-action switch used to power up the drive when the disk pack is installed, the access door is locked, and the power supply circuits are on. If these conditions are met when you activate the switch, the START indicator turns on and the drive spindle begins to accelerate to operating speed.

When you actuate the START switch, the **READY indicator** switch flashes once a second until the drive spindle reaches operating speed. When the drive is up to speed and the heads are loaded, this indicator remains on steadily if no fault conditions are detected.

Each drive comes with eight logical address plugs numbered 0 through 7. You select the drive's unit address by inserting the appropriately numbered plug into the **drive select receptacle**. The plugs are mechanical devices that actuate a set of switches to select the unit address that corresponds to the number inscribed on the plug face.

The **FAULT indicator** turns on when one or more of the following fault conditions are detected:

- Write fault

- Head select fault

- Read/write fault

- Read/write while off cylinder fault

- Voltage fault

If the fault indicator lights due to a momentary fault, you can turn it off by pressing the **FAULT CLEAR switch** on the front panel. If the fault indicator stays on when you press the FAULT CLEAR switch, notify maintenance personnel.

The **write protect switch** is an alternate-action switch used to protect against unintentional overwriting of the disk pack. When you activate the switch, the PROTECT indicator turns on and the drive write circuits are disabled. Conversely, when you deactivate the switch, the PROTECT indicator turns off and the controller may write to the disk pack.

If you have port selection, you use the **A/B switches** to manually give the controller access to the drive through either port A, port B, or both ports. Each switch has an associated activity indicator located directly above it.

3.6 R80 and RM80 Disk Drives

The R80 and RM80 disk drives are random-access, disk-storge systems that implement Winchester fixed-media technology.

The RM80 is a MASSBUS device that connects to either a VAX–11/780 or VAX–11/750 by means of the MASSBUS controller. It has a storage capacity of 124 megabytes.

The R80 is used with VAX–11/730 systems as a system device and has a storage capacity of 121 megabytes. The R80 connects to the UNIBUS by means of an integrated disk controller.

Both the RM80 and the R80 have identical front panel controls and indicators:

- RUN/STOP switch and indicator

- FAULT switch and indicator

- Logical address receptacle and READY indicator

- WRITE PROT switch and indicator

- STAT 1 and STAT 2 indicators

The RM80 optionally provides two port selection switches.

Figure 3–6 gives the locations of the controls and indicators.

When the drive is at rest, the **RUN/STOP switch** is used to start the drive spindle accelerating to operating speed. The indicator turns on when you press the switch and remains on until you stop the drive.

When the drive is at operating speed, the RUN/STOP switch is used to bring the spindle to a gradual stop. When you press the switch, the drive spindle begins to decelerate, but the RUN /STOP indicator remains on until the spindle stops.

The **FAULT indicator** turns on when a fault is detected. When you press the lit switch, five front panel indicators flash an error code that identifies the faulty component.

Figure 3–6   R80 and RM80 Front Panel Controls and Indicators

RUN STOP	FAULT	2 READY	WRITE PROT	STAT 1	STAT 2

FRONT PANEL CONTROLS AND INDICATORS

A	B

DRIVE CABINET SWITCHES
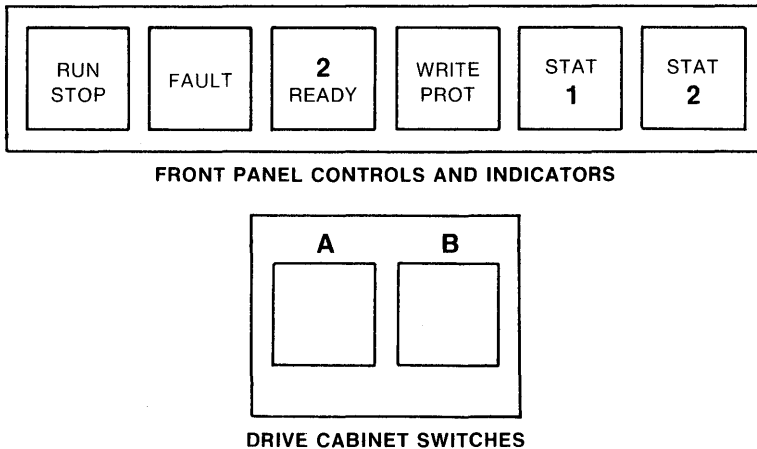
ZK-1951-84

The **logical address receptacle** lets you select a unit address for each drive by simply inserting the appropriate logical address plug into the receptacle. With each drive, you receive eight plugs (numbered 0 through 7) from which you may select the desired unit address.

Each plug is a mechanical device that actuates a set of address switches corresponding to the unit address numeral inscribed on the plug face. For example, if the plug is inscribed with the numeral **0**, it will actuate a set of switches that assign the unit address 0 to the drive.

Each logical address plug incorporates a **READY indicator** that turns on when the drive is at operating speed and remains on until you press the RUN/STOP switch to stop the drive.

The **STAT 1 and STAT 2 indicators**, together with the FAULT, READY, and WRITE PROTECT indicators, flash an error code when you press the FAULT switch.

The **WRITE PROT (write protect) switch** is an alternate-action switch that protects you from inadvertantly overwriting the disk. When the switch is activated, the indicator turns on and the disk is write protected. Conversely, if the switch is deactivated, the indicator turns off and the processor is allowed to write to the disk.

The RM80 **port selection switches** allow two processors to share a driver, or a single processor to access two drivers. At the driver, port access is determined by the state of the **A** and **B** selectors switches on the driver cabinet. You actuate the **A** switch to enable access through port A and the **B** switch to enable access through port B. The **A** indicator turns on when port A is being used, and the **B** indicator turns on when port B is being used.

3.7   RA60, RA80, and RA81 Disk Drives

The RA60, RA80, and RA81 disk drives attach to the host either by means of a UNIBUS disk adapter (UDA) controller, or by means of a hierarchical storage controller (HSC) and the computer interconnect (CI). The RA60 provides a removable disk; the RA80 and RA81 implement fixed-head media. All three drives provide port selection to make them simultaneously accessible to two controllers.

All three drives include the following front panel controls and indicators:

* RUN/STOP switch and indicator

* FAULT switch and indicator

* Logical address receptacle and READY indicator

* WRITE PROT switch and indicator

* Port selection switches and indicators

Figure 3–7 gives the locations of the controls and indicators.

Figure 3–7 RA60, RA80, and RA81 Front Panel Controls and Indicators

| RUN STOP | FAULT | 4 | READY | WRITE PROT | A | B |
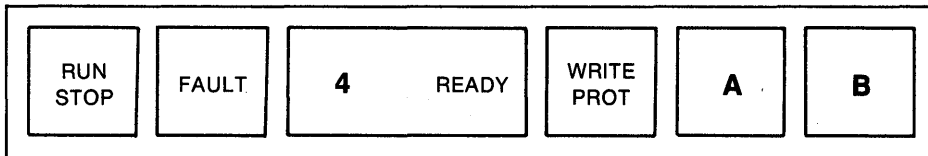
ZK-2035-84

In addition to the functions described below, the indicators flash an error code when you press the FAULT switch.

When the drive is at rest, you use the **RUN/STOP switch** to start the drive accelerating to operating speed. The indicator turns on when you press the switch and remains on until you subsequently stop the drive. You can only gain access to the RA60 drive compartment if the RUN/STOP indicator is turned off.

When the drive is at operating speed, the RUN/STOP switch is used to bring the drive to a gradual stop. The drive spindle immediately begins to decelerate but the RUN/STOP indicator remains on until the spindle stops.

The **FAULT indicator** turns on when a fault is detected. When you press the lit switch, the five indicators will flash an error code that identifies the faulty component.

The **logical address receptacle** lets you select a unit address for each drive by inserting the appropriate logical address plug. With each drive, you receive eight plugs (numbered 0 through 7) from which you may select the desired unit address.

Each plug is a mechanical device that actuates a set of address switches corresponding to the unit address numeral inscribed on the plug face. For example, if the plug is inscribed with the numeral **0**, it will actuate a set of switches that assign the unit address 0 to the drive.

Each logical address plug incorporates a **READY indicator** that turns on when the drive is at operating speed and remains on until you press the RUN/STOP switch to bring the drive to a gradual stop.

The **WRITE PROT switch** (write protect) is an alternate-action switch that protects you from inadvertantly overwriting the disk. When the switch is activated, the indicator lights and the disk is write protected. Conversely, if the switch is deactivated, the indicator turns off and the processor is allowed to write to the disk.

With dual-port capability, the **port selection switches** allow two processors to share a drive, or a single processor to access two drives. When you actuate a port switch, the associated indicator turns on.

3.8 RC25 Disk Drive

The control panel for the RC25 disk drive includes five switches with related indicators:

- The RUN switch and indicator

- The WRITE PROTECT REMOVED switch and indicator

- The WRITE PROTECT FIXED switch and indicator

- The FAULT switch and indicator

- The EJECT switch and indicator

Figure 3–8 shows the locations of the controls and indicators.

The **RUN switch** is used to either bring the drive spindle up to operating speed or to spin it down. Note that the drive spindle *cannot* be brought up to operating speed unless the removable cartridge is in the drive.

When you press the switch to bring the spindle up to speed, the indicator flashes as the spindle is coming up to speed and remains lit when the spindle is at operating speed.

When you use the switch to spin down the spindle, the indicator flashes as the spindle speed diminishes and turns off when the spindle is stopped. Simultaneously, the EJECT indicator turns on to indicate it is safe to remove the cartridge.

Figure 3–8   RC25 Control and Indicators
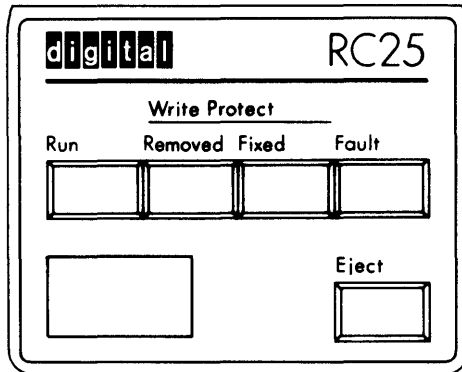
ZK-1001-82

The **WRITE PROTECT switches** are used to write protect the
appropriate head, either the FIXED or the REMOVED. When
you write-protect a head, the corresponding indicator turns on.

The **FAULT indicator** turns on when a fault is detected. When
you press the lit **FAULT switch**, the indicators flash an error
code to identify the faulty component.

The **EJECT switch** is used to open the cartridge receiver door
when you want to insert or extract the removable cartridge.
The **EJECT indicator** turns on when it is safe to open the door.

3.9   Handling Disk Cartridges and Disk Packs

This section provides instructions on how to physically handle
disk media, including the disk cartridges used in the RL02 and
RK07 disk drives, and the large-capacity disks.

3.9.1   Handling RL02 and RK07 Disk Cartridges

This section tells you how to physically load and unload disk cartridges. Refer to Figure 3–9 and Figure 3–10 while using this procedure.

To load a cartridge into a drive:

1 Press the drive lid release bar and raise the lid.

2 Lift the disk cartridge by grasping the top cover handle with one hand.

3 While supporting the base of the protective cover with your other hand, lower the top cover handle.

4 Push the handle slide to the left with your thumb while lifting the handle. You will feel the release as the cartridge disengages from the protective cover.

5 Lift the cartridge away from the protective cover and place it in the drive shroud with the top cover handle recess facing the rear of the drive.

6 Rotate the top cover handle a few degrees clockwise and counterclockwise to be sure the shroud locating studs are properly seated within the cartridge housing.

7 Gently lower the top cover handle to a horizontal position so the cartridge engages the drive spindle.

8 Place the protective cover on top of the cartridge cover.

9 Close the lid.

To unload the cartridge:

1 Bring the drive to rest.

2 Press the lid release bar and raise the lid.

3 Remove the protective cover from the disk top cover.

Figure 3-9   Disk Drive with Lid Released
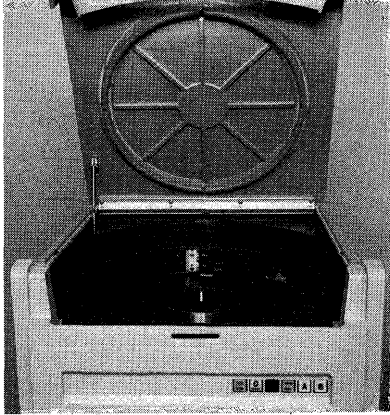
Figure 3-10   Disk Cartridge
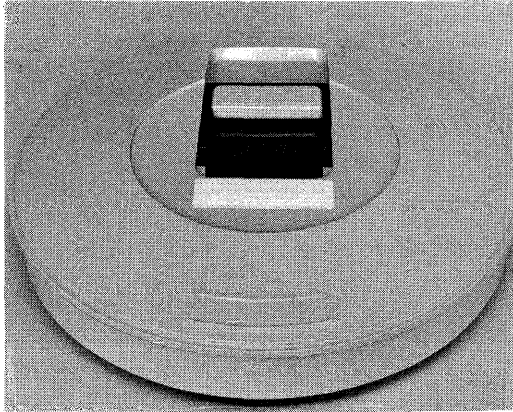
4 While supporting the base of the protective cover with your palm, use the thumb of your other hand to push the cartridge handle slide to the left and raise the handle to disengage the cartridge from the drive spindle.

5 Lift the cartridge out of the shroud and place it in the protective cover.

6 Lower the top cover handle to secure the cartridge to the protective cover.

3.9.2 Handling Disk Packs

Disk packs include three or more platters and are protected by a plastic storage canister when you are handling them (see Figure 3–11).

The plastic storage canister includes a pack handle that is also used to disengage the canister from the disk pack. You disengage the bottom dust cover from the disk pack by squeezing two latching levers located beneath the base. When the plastic canister and the dust cover are removed from the disk pack, they cannot be mechanically coupled.

Before moving the disk pack, be sure it is secured to the plastic canister.

3.9.2.1 Installing the Disk

Use this procedure in conjunction with Figure 3–12 to install the disk in the disk drive.

1 Be sure the disk drive is at rest and that it is safe to open the pack access door (or cover) before attempting to load or unload the disk pack. (See appropriate descriptions on controls and indicators.)

If you have to remove a disk pack before installing a replacement pack, see the next section for instructions on how to remove a disk pack before proceeding to install your disk pack.

2 Open the drive access door (or cover).

3 Hold the plastic canister by the handle in one hand and remove the dust cover by squeezing the latching levers.

Figure 3–11   Disk Pack Construction

PACK HANDLE

STORAGE
CANISTER

HUB OF GUARD DISK

DISK PACK

BOTTOM
DUST
COVER
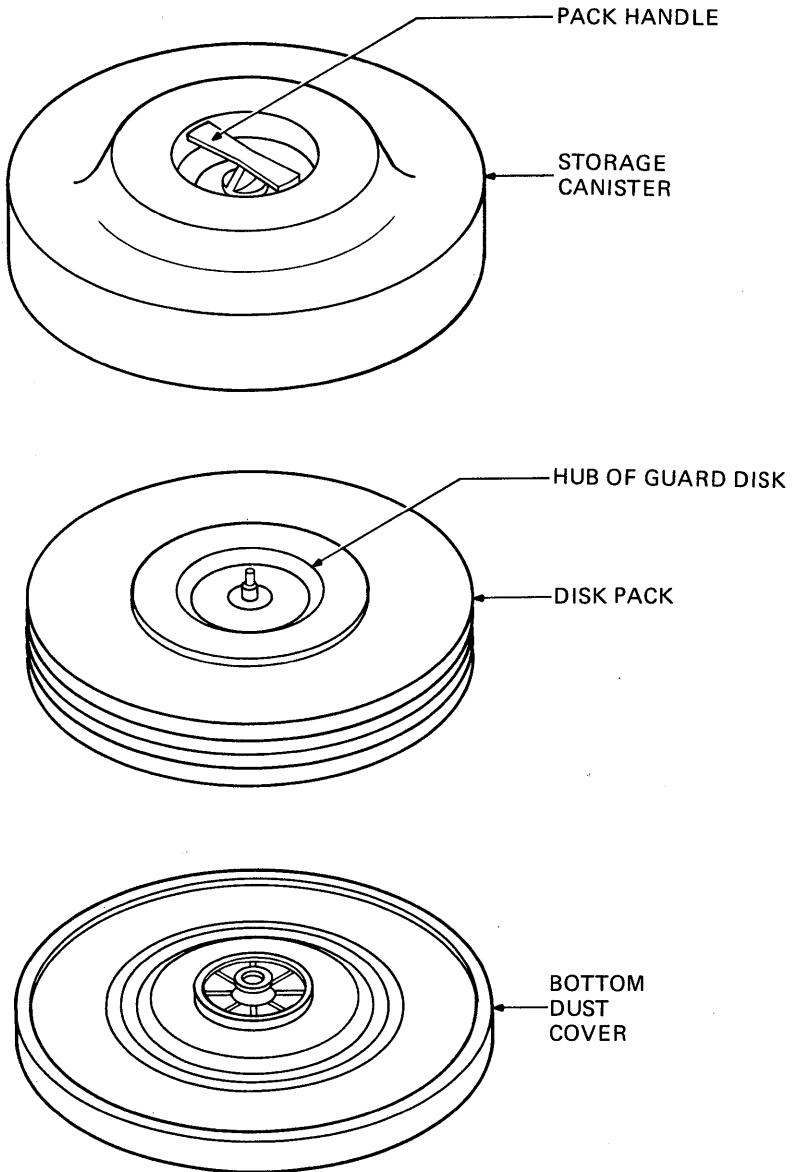
ZK-1960-84

Figure 3–12   Disk Pack Access

PACK ACCESS COVER

HEADS IN LOADED
POSITION

SPINDLE

SHROUD
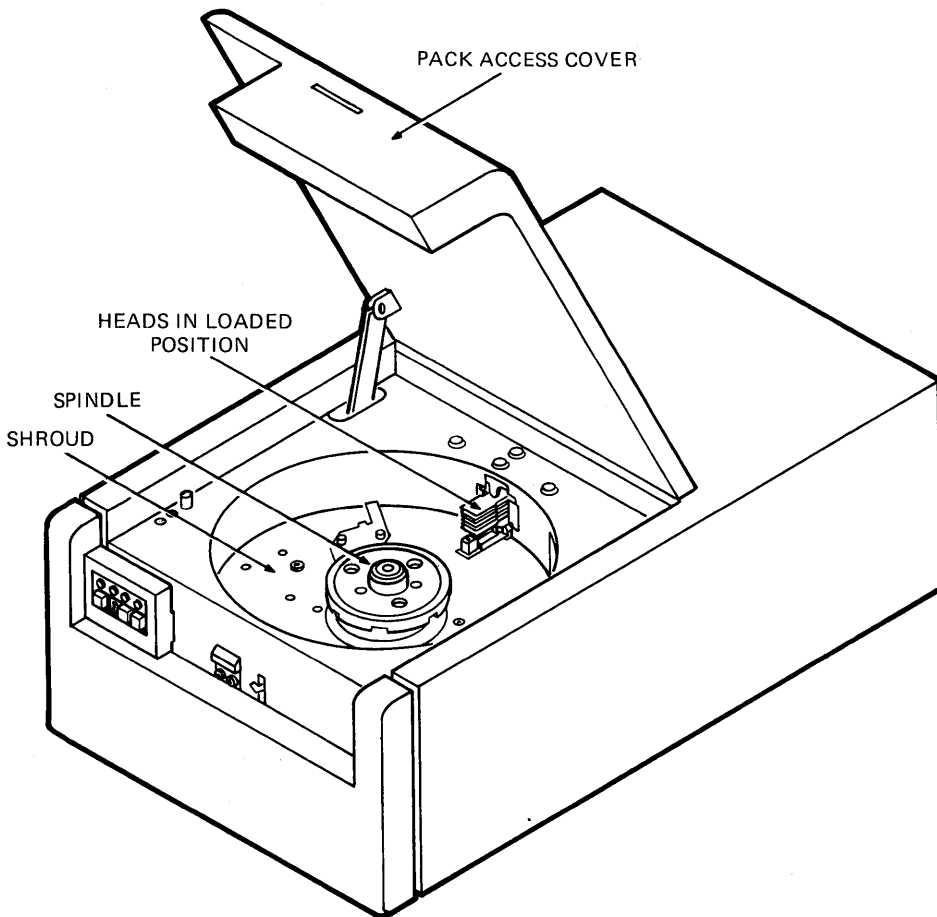
ZK-1961-84

4 Carefully place the disk pack (still attached to the plastic
canister) in the drive so that it rests on the drive spindle.

5 Turn the handle on the plastic canister clockwise until it
stops turning easily. Do **not** force the handle.

6 Lift the plastic canister straight up from the disk pack, taking
care to avoid damaging the edges of the disks.

7 Place the plastic canister on the dust cover and store them in a safe place. Note that the dust cover does not latch to the plastic canister when the disk pack is removed.

8 Close the drive access door (or cover).

3.9.2.2	**Removing the Disk Pack from the Drive**

Use this procedure to remove the disk pack from the drive. If you need help with any of the steps, refer to appropriate descriptions on drive controls and indicators.

1 Stop the drive.

2 When it is safe to remove the disk pack, open the drive access door (or cover).

3 Place the plastic canister over the the disk pack, taking care to avoid damaging the edges of the disks.

4 Turn the handle on plastic canister counterclockwise two full turns. An audible clicking sound indicates the disk is free.

5 Grasp the handle of the plastic canister and carefully lift the disk pack (now attached to the plastic canister) out of the drive.

6 While holding the disk pack in one hand, attach the dust cover to the pack by squeezing the latching levers on the dust cover, seating the disk pack into the dust cover, and then releasing the latching levers.

7 Store the disk pack in a safe place, preferably on a flat shelf. Do not stack disk packs on one another.

3.10 TS11 Tape Drive

The TS11 tape drive is used solely as a load device. It is compatible with all VAX systems but is usually configured with the low- and medium-end members. The TS11 connects to the UNIBUS by means of the I/O bus and a controller.

The TS11 front panel provides two switches and six indicators for controlling and monitoring the drive's status:

- The LOAD/REW/UNLD (LOD in some versions) switch and indicator

- The ON-LINE (ONL in some versions) switch and indicator

- The MICRO OK (UOK in some versions) indicator

- The VOL VALID (VCK in some versions) indicator

- The DENS ERROR (DCK in some versions) indicator

- The WRITE LOCK (WLK in some versions) indicator

- The BOT (beginning of tape) indicator

- The EOT (end of tape) indicator

Figure 3–13 shows the locations of the controls and indicators.

Figure 3–13   TS11 Front Panel Controls and Indicators

LOAD REW UNLD	ON-LINE	MICRO OK	VOL VALID	DENS ERROR	WRITE LOCK	BOT	EOT

ZK-1955-84

The **LOAD/REW/UNLD switch** has three functions depending on the current state of the drive.

- If you have just threaded the tape, you use the switch to load (LOAD position) the tape.

- If you are not at the beginning of the tape, you can use the switch to rewind (REW position) the tape.

- If you are at the beginning of the tape, you can use the switch to unload (UNLD position) the tape.

The **ON-LINE switch** is an alternate-action switch used to bring the tape transport on line or off line as appropriate. The indicator turns on if the transport is on line.

The six indicators are listed below:

The **MICRO OK indicator** turns on only if the microprocessor-controlled formatter is operating without errors.

The **VOL VALID indicator** turns on to alert you to a change in the status of the tape transport; that is, it turns on if the transport has changed from on line to off line, or vice versa, since the last processor command.

The **DENS ERROR indicator** turns on if an invalid identification burst is sensed at BOT. This usually indicates that a phase-encoding error has probably occurred and that the tape data is invalid.

The **WRITE LOCK indicator** turns on if you place a tape reel on the transport that does not have a file protect ring on it.

The **BOT indicator** turns on when the beginning of the tape is sensed.

The **EOT indicator** turns on when the end of the tape is sensed.

3.11   TE16 Tape Drive

The TE16 tape drive is used solely as a load device. It is configured with MASSBUS systems only and connects to the processor by means of the MASSBUS and the MASSBUS adapter.

The TE16 front panel provides four indicators, a transport select receptacle, and three switches.

- The PWR (power) indicator

- The BOT (beginning of tape) indicator

- The WRL (write lock) indicator

- The SELECT indicator
- The TRANSPORT SELECT (unlabeled) receptacle
- The LOAD switch and indicator
- The ON LINE switch and indicator
- The REW/UNLD (rewind/unload) switch

Figure 3–14 shows the locations of the controls and indicators.

Figure 3–14 TE16 Front Panel Controls and Indicators
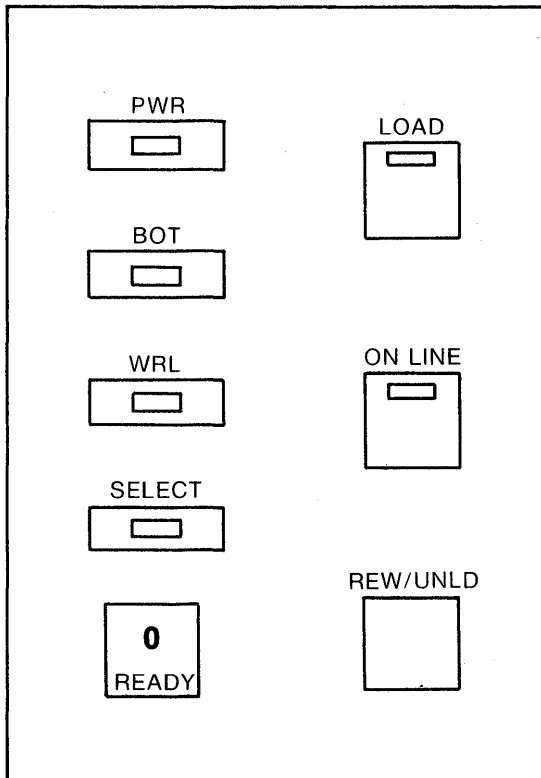
ZK-1956-84

The **PWR indicator** turns on when power is applied to the transport.

The **BOT indicator** turns on when the beginning of the tape is sensed.

The **WRL indicator** turns on if you place a tape reel on the transport that does not have a file protect ring on it.

The **SELECT indicator** indicates which transport the processor has selected.

The **TRANSPORT SELECT receptacle** is used to assign a unit number (from 0 through 7) to a transport by receiving the appropriate unit-number plug. Each TE16 includes eight unit-number plugs, and you may not assign the same unit number to more than one transport on a formatter.

After you have threaded the tape onto the transport, pressing the **LOAD switch** causes the transport to load the tape and turn on the indicator.

The **ON LINE switch** is an alternate action switch used to either put the transport on line or to take it off line. The indicator turns on only when the transport is on line.

The **REW/UNLD switch** is used to either rewind (REW position) the tape; or if the tape is at BOT, unload (UNLD position) it.

3.12 TU77 Tape Drive

The TU77 tape drive is used solely as a load device. It is configured with MASSBUS systems only and connects to the processor by means of the MASSBUS and the MASSBUS adapter.

The TU77 front panel provides a transport select control, four switches, and seven indicators.

- TRANSPORT SELECT (unlabeled) thumbwheel control and indicator

- LOAD REW (load/rewind) switch

- ON LINE switch

- UNLOAD switch
- RESET switch
- POWER indicator
- BOT (beginning of tape) indicator
- ON LINE indicator
- FILE PROTECT indicator
- LOAD FAULT indicator
- Two tape density indicators (800/1600)

Figure 3–15 shows the locations of the controls and indicators.

Figure 3–15 TU77 Front Panel Controls and Indicators
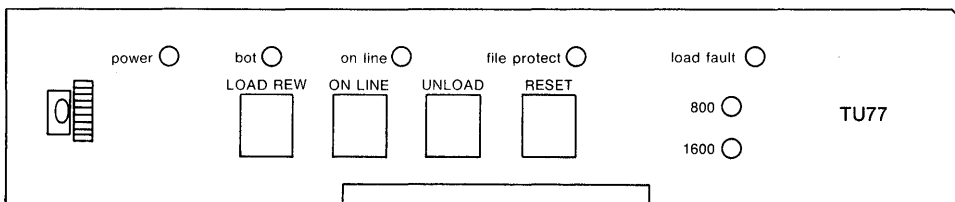
ZK-1957-84

The **transport select control** is a thumbwheel used to assign a unit address to the tape drive. As you turn the thumbwheel, your selection is visible through an adjoining window in the front panel.

After you have threaded the tape onto the transport, putting the **LOAD/REW switch** in the LOAD position causes the transport to load the tape. When the tape is loaded, putting the switch in the REW position rewinds the tape to BOT (beginning of tape).

The **ON LINE switch** is an alternate action switch used to either put the transport on line or take it off line.

The **UNLOAD switch** is operative only when the tape drive is off line. When you press the UNLOAD switch, the transport rewinds and then unloads the tape.

3–30

The **RESET switch** is used to take the tape transport off line and turn off the LOAD FAULT indicator if it is lit.

The **POWER indicator** turns on when power is applied to the transport.

The **BOT indicator** turns on when the beginning of the tape is sensed.

The **ON LINE indicator** turns on when the transport is on line.

The **FILE PROTECT indicator** turns on if the tape reel on the transport does not have a file protect ring on it.

The **SELECT indicator** indicates which transport the processor has selected.

The **LOAD FAULT indicator** indicates the automatic tape loading mechanism tried to load the tape twice and failed.

The **800 indicator** turns on when the transport is set to read or write at 800 bits-per-inch.

The **1600 indicator** turns on when the transport is set to read or write at 1600 bits-per-inch.

3.13   TU78 Tape Drive

The TU78 tape storage system is used solely as a load device. It is configured with MASSBUS systems only and connects to the processor by means of the MASSBUS and the MASSBUS adapter. The TU78 features static dual-port access that gives two processors access to the TU78 or the TU78 access to a processor through two ports.

The TU78 front panel provides a transport select control, four switches, and seven indicators.

- PORT SELECT thumbwheel control and indicator

- LOAD REW (load/rewind) switch

- ON LINE switch

- UNLOAD switch

- RESET switch

- POWER indicator

- BOT (beginning of tape) indicator

- ON LINE indicator

- FILE PROTECT indicator

- LOAD FAULT indicator

- Two tape density indicators (6250/1600)

Figure 3–16 shows the locations of the controls and indicators.

Figure 3–16 TU78 Front Panel Controls and Indicators
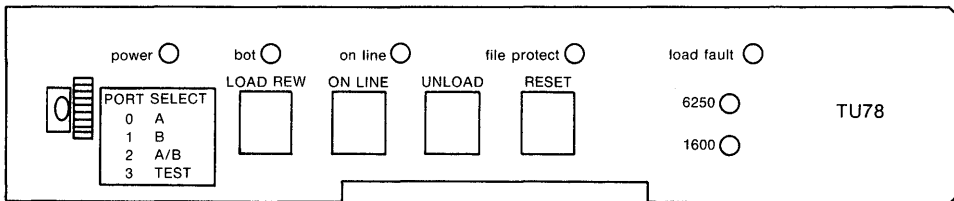
ZK-1958-84

The **port selection control** is a thumbwheel control used to determine port selection for the TU78:

0   MASSBUS port A is selected.

1   MASSBUS port B is selected.

2   Both MASSBUS ports are selected.

3   Neither MASSBUS port is selected.

When the tape is threaded on the transport, putting the **LOAD /REW switch** in the LOAD position causes the transport to load the tape. When the tape is loaded, putting the switch in the REW position rewinds the tape to BOT (beginning of tape).

The **ON LINE switch** is an alternate action switch used to either put the transport on line or to take it off line.

The **UNLOAD switch** is operative when the tape drive is off line. When pressed, the transport rewinds and unloads the tape.

The **RESET switch** takes the tape transport off line and turns off the LOAD FAULT indicator if it is lit.

The **POWER indicator** turns on when power is applied to the transport.

The **BOT indicator** turns on when the beginning of the tape is sensed.

The **ON LINE indicator** turns on when the transport is on line.

The **FILE PROTECT indicator** turns on if the tape reel on the transport has no file protect ring.

The **SELECT indicator** indicates which transport the processor has selected.

The **LOAD FAULT indicator** turns on when the automatic tape loading mechanism fails twice to load the tape.

The **6250 indicator** turns on when the transport is set to read or write at 6250 bits-per-inch.

The **1600 indicator** turns on when the transport is set to read or write at 1600 bits-per-inch.

3.14 TU80 and TU81 Tape Drives

The TU80 and TU81 tape storage systems are used solely as load devices during an installation. In post-installation operations, these systems are used primarily for disk backup.

The TU80 and TU81 tape systems are very similar, as they are both UNIBUS devices. The primary difference is that the TU81, which is the first tape drive supported by DSA software protocol, has a higher storage density and higher data transfer rates than the TU80.

The front panels for both tape drives are virtually the same. Both front panels are partitioned into three sections:

- Operator controls and indicators

- Maintenance controls and indicators

- Special controls and indicators

Figure 3–17 shows the locations of the front panel controls and indicators.

Figure 3–17   TU80 and TU81 Front Panel Controls and Indicators
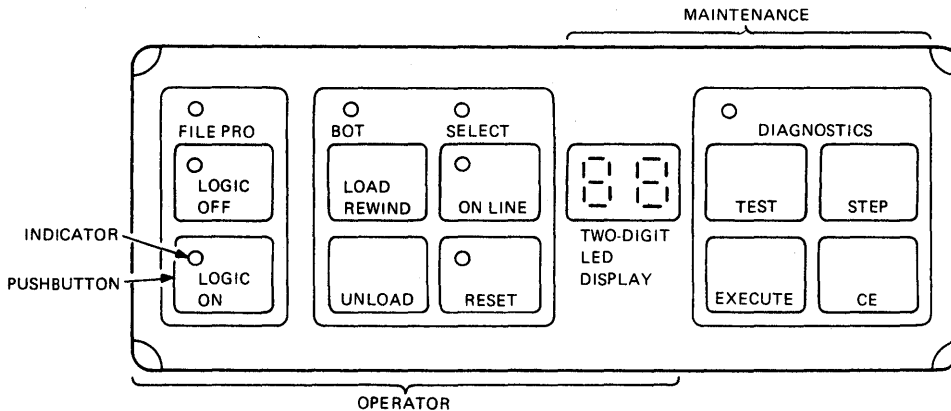
ZK-2075-84

This description is limited to indicators and controls used in routinely operating and monitoring the tape system.

• FILE PRO (file protect) indicator

• LOGIC ON switch and indicator

• LOGIC OFF switch and indicator

• BOT (beginning of tape) indicator

• LOAD/REWIND switch

• UNLOAD switch

• SELECT indicator

• ON LINE switch and indicator

• RESET switch and indicator

The **FILE PRO indicator** turns on when the write enable ring is missing so writing is not allowed.

The **LOGIC OFF switch** is used to take power from the system. The indicator turns on to indicate main power applied but subassembly power removed.

The **LOGIC ON switch** is used to make the tape system ready for operation. When the system is operational, the indicator turns on.

The **BOT indicator** turns on at the beginning of the tape.

The **LOAD/REWIND switch** is used to load the tape when the transport is powered on and the tape is threaded. When the tape is loaded, this switch is used to rewind the tape.

The **UNLOAD switch** is used to unload the tape onto the supply reel when tape is at BOT or when it is threaded but not loaded. If the tape is loaded but is beyond BOT, this switch is used to rewind the tape.

The **ON-LINE switch** is used to make the tape accessible to the host system and to turn on the indicator. Press the RESET switch to take the system off line.

The **SELECT indicator** turns on when the host system is doing group code recording (GCR) operations with the tape drive.

The **RESET switch** is used to take the transport off line, to stop the tape and clear the error state, to turn off the RESET indicator and clear a diagnostic test condition and to stop LOAD or REWIND operations. The indicator turns on if a fault is detected or if the system is in a diagnostic state.

3.15 HSC50 Controller

The HSC50 (hierarchical storage controller) provides one or more host computers with access to a set of mass storage devices by means of the computer interconnect (CI) and the star coupler. It can support a mix of Disk Storage Architecture (DSA) disk devices (RA60, RA80, and RA81) and the TA78 tape systems.

The HSC50 provides the seven controls and indicators on the operators control panel (OCP) pictured in Figure 3–18.

Figure 3–18   Operator Control Panel Controls and/Indicators

ZK-2076-84
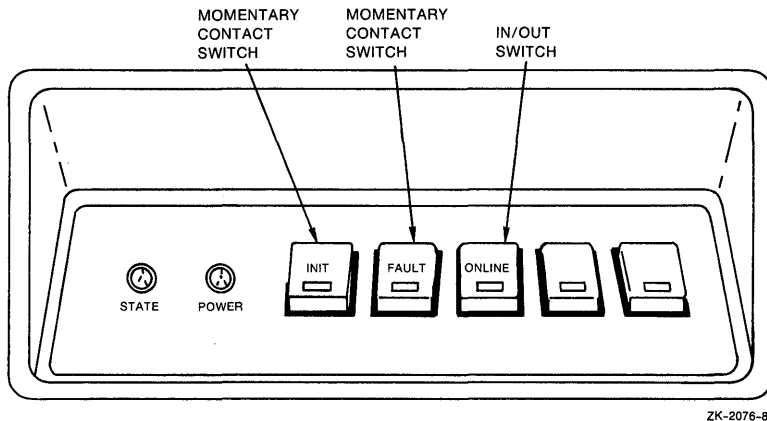
- STATE indicator

- POWER indicator

- INIT (initialize) switch and indicator

- FAULT switch and indicator

- ONLINE switch and indicator

- Two blank indicators

An internal SECURE/ENABLE switch provides security against activating the OCP controls inadvertently. The OCP controls are disabled when the switch is positioned at SECURE.

The **STATE indicator** turns on when the initialization sequence is completed. It remains lit for several minutes during the early stages of the bootstrap diagnostic tests and then flashes at half-second intervals for the balance of the bootstrap process. When the bootstrap process completes, the STATE indicator flashing rate reduces to 1-second intervals and continues to flash at this rate through normal operations.

The **POWER indicator** turns on when power is applied to the HSC. It remains lit if all DC voltages are at the correct level and AC power is present.

3–36

The **INIT switch** is a momentary-contact switch used to
initialize and boot the HSC. When you press the INIT switch,
the indicator turns on and remains on during the initialization
sequence. When the initialization sequence is completed, the
INIT indicator turns off and the STATE indicator turns on.

The **FAULT indicator** turns on when a significant problem is
detected in the HSC.

The **FAULT switch** is used to test the OCP indicator lamps and
to determine the nature of detected faults.

If the FAULT lamp is not lit (no fault present), the switch may
be used to test the condition of the OCP indicators. If any OCP
indicator does not turn on, notify maintenance personnel.

If the FAULT indicator is lit (fault is present) when you press
the switch, the OCP indicators will flash a 5-bit code that
identifies the problem. If more than one fault condition exists,
pressing the FAULT switch causes the appropriate fault codes
to flash.

To determine the effect of a fault on the operational state
of the HSC, press the FAULT switch a second time. If the
HSC is not operating, the fault code continues to flash. If the
HSC is operating, the FAULT indicator turns off and the other
indicators return to their normal state.

ONLINE switch is a 2-position switch that enables the HSC
to make connection with other nodes on the CI. See the *HSC50
User Guide* for additional control settings to put the HSC on
line.

When the switch is in the active position (IN), the HSC is
able to connect with host computers on the CI. The **ONLINE
indicator** turns on when the first connection is made and
remains on until all connections are terminated even if you
deactivate the **ONLINE switch**.

When you deactivate the switch (OUT position), the HSC is not
allowed to make any further connections with nodes on the CI;
however, existing connections are maintained and the HSC is
still on line.

3–37

If the **ONLINE indicator** is not lit (indicating no host computer connections) when the switch is deactivated, the HSC immediately goes off line.

The **two blank indicators** are used to provide the two least significant bits in the OCP fault codes.

4 Support Information

The software installation booklets provide a step-by-step approach to installing a new VAX/VMS system under 16 scenarios. This chapter supports the booklets with additional details.

4.1 Logging In to the System

When you boot the system, it announces itself with the following message:[1]

```
   VAX/VMS Version 4.0 16-SEP-1984 12:00
%OPCOM, 20-DEC-1984 16:36:29.58, logfile initialized by operator OPA0
        log file is SYS$MANAGER:OPERATOR.LOG
   Login quotas - Interactive limit=64, Current Interactive value=0
SYSTEM  job terminated at 20-DEC-1984 16:36;31.41
```

To log in to the system as the system manager, you must perform the following steps:

1 Press the RETURN key on the console terminal.

2 When the system prompts you for your username, type the word SYSTEM as shown:

```
USERNAME: SYSTEM
```

3 When the system prompts for it, enter your password. Note that the password will not echo, that is, you will not see the password displayed on the screen as you enter it.

When you enter the proper password, the system prints the following message on the console terminal:

```
Welcome to VAX/VMS Version 4.0.
```

---

[1] All dates and times used in message examples are for illustration only.

4–1

When you log in, make your entries quickly and accurately. The security features of the system may disable login if you make too many mistakes or if you take too much time. If you do make a mistake, press the RETURN key to start the login procedure again.

When the console terminal prints the dollar-sign prompt ($), you may start to use the system.

4.2 Booting During Installation

At various points in the installation procedure, you must boot software into memory from a storage device, either a console storage device (TU58 or floppy diskette) or from a disk. For example, you begin the installation by booting stand-alone BACKUP into memory so that you can do a partial restoration of the operating system. Then you boot the partially restored operating system into memory so that you can restore the remaining system files. Finally, when the entire operating system has been restored, you boot it into memory from the system disk so that you can begin operations.

4.2.1 Booting VAX Processors

All booting is done in console mode, that is, under the direction of the console program. On the VAX–11/725, the VAX–11/730 and the VAX–11/750, you put the system into console mode and halt the CPU by pressing CTRL/P on the console terminal keyboard. On the VAX–11/780, VAX–11/782 and VAX–11 /785, CTRL/P puts the system in console mode, but you must use the console command HALT to halt the CPU.

The system indicates it is in console mode by displaying the console prompt (>>>) on the console terminal.

With the system in console mode, you boot software from a device by typing the word BOOT (or simply the letter B) followed by the abbreviated name of the boot command procedure for the particular device.

Upon completion of the boot, the console program automatically returns the system to program mode.

If you have a VAX–11/750, you may boot software in one of two ways:

• Directly from the system disk

• Indirectly using the stand-alone BOOT58 program on the console TU58

It is best to boot directly from the system disk; use the BOOT58 program only if you cannot boot directly from the disk.

There are two ways you can boot directly from the system disk.

The first way is to go into console mode and issue a boot command in the following format:

```
>>> B ddcu
```

You specify the disk device using the *ddcu* format (see Section 4.4.)

For example, if the system disk is the first RK07 on controller A, you could use the following command to boot it:

```
>>> B DMA0
```

The second way to boot directly from the system disk is to assign the system disk a position on the BOOT DEVICE switch. With the BOOT DEVICE switch at the appropriate position, set the POWER ON ACTION switch to BOOT to boot your system disk.

You should use BOOT58 only when you cannot boot directly from the disk. The most common reason for being unable to do so is because the boot block on the disk has been corrupted. To boot the system disk using BOOT58, use the following procedure:

1 Insert the console TU58 tape cartridge into the console drive.

2 Set the keylock switch on the processor control panel to LOCAL.

3 Set the POWER ON ACTION switch to HALT.

4 Press CTRL/P to obtain the console program prompt.

5 Enter the console BOOT command to boot the TU58 tape cartridge:

```
>>> B/800 DDA0
```

6 When you get the BOOT58 prompt, enter the name of the applicable boot command procedure.

```
BOOT58> @DxyBOO.CMD
```

@   Indicates that the rest of the line contains the name of a command procedure.

x   A letter that indicates the type of disk drive holding the system disk. For example, the letter U indicates either an RA80 or RA81 disk drive (see Table 4–1).

y   A number that specifies the unit number of the disk drive holding the system disk.

You may use a short form to specify the desired boot command procedure:

```
BOOT58> BOOT Dxy
```

This specification is the same as using the long form. For example, if you want to boot from the first RA80 on controller A, you can use either of these forms:

```
BOOT58> @DUOBOO.CMD
```

```
BOOT58> BOOT DU0
```

See the *Guide to VAX/VMS System Management and Daily Operations* for more information on using the BOOT58 program.

4.3   Booting From HSC Disks

The manner in which you boot an HSC disk is in part dependent on the type of VAX processor you are using. If you have a VAX–11/750, you use the BOOT58 utility; if you have a VAX–11/780, you use console mode commands.

If you want to use a nonstop boot or are installing a new system, you use the CIBOO.CMD command procedure; if you want to change system parameters during the boot, you use the CIGEN command procedure. In either case, you must provide the selected boot command file with two pieces of information before you use it to boot the disk: the unit number of the disk drive and the node number of the HSC controlling it.

Furthermore, if you want to boot from a system root other than SYS0, you must identify the root. Note however that when installing a system, you boot from SYS0. See the *Guide to VAX/VMS System Management and Daily Operations* for more information on booting from alternate system roots and see Chapter 6 of this manual for details on creating alternate roots.

The way you provide the information to the CI boot command procedure depends on whether you have a running system or are installing a new system.

If you have a running system, you can copy the CI boot command file from the console volume to a disk directory, edit it, and copy the edited version back to the console volume. If you want to use the CI boot command file as your default boot command procedure, copy the edited version to DEFBOO.CMD on the console volume. See the *Guide to VAX/VMS System Management and Daily Operations* for details.

If you are installing a new system, you will have to use console commands (BOOT58 commands if you have a VAX–11/750) to deposit the information. Note that the form of the DEPOSIT command varies for different VAX processors.

The following sections describe four scenarios for booting HSC disks:

- Booting a VAX–11/780 during an installation
- Booting a running VAX–11/780
- Booting a VAX–11/750 during an installation
- Booting a running VAX–11/750

4.3.1 Booting a VAX–11/780 During an Installation

This section describes the steps for booting an HSC disk on a VAX–11/780 while installing a new operating system.

1 Determine the unit number of the HSC disk and the node number of the HSC controlling it.

2 Press CTRL/P to put the system in console mode.

3 Use the console HALT command to stop the processor.

4 When you get the console prompt ($>>>$), deposit the
HSC node number into Register R2 using the following
command format. Note that all numeric entries are made
using hexadecimal notation.

```
>>> DEPOSIT R2 node_number
```

For example, if the HSC is node number 12 (hexadecimal
0C) on the CI780, use this command:

```
>>> DEPOSIT R2 0C
```

5 If the disk device is accessible to two HSCs, deposit both
node numbers in hexadecimal values putting the larger
number in hex digits 3 and 2, and the lesser in digits 1 and
0. For example, if one HSC is numbered 15 (hex 0E) and
the other is numbered 10 (hex 0A), the proper command is

```
>>> DEPOSIT R2 0E0A
```

6 Deposit the unit number of the HSC disk into register R3
using a command in the following format:

```
>>> DEPOSIT R3 unit_number
```

For example, if the HSC disk is unit number 21, deposit a
hexadecimal 15 into register R3:

```
>>> DEPOSIT R3 15
```

7 Boot the HSC disk with this command:

```
>>> @CIBOO.CMD
```

4.3.2   Booting a Running VAX–11/780

This section describes how to set up DEFBOO.CMD for a
nonstop system boot of an HSC disk on a VAX–11/780 system
that is up and running. In this example, the HSC device is an
RA60 disk drive assigned unit number 03 on an HSC controller
assigned node number 12 (hexadecimal 0C). The example
assumes the user wants to boot the system from alternate root
SYSA.

1 Be sure your console drive is connected. If it is not, invoke
the System Generation Utility (SYSGEN) to connect it, as
follows:

```
$ RUN SYS$SYSTEM:SYSGEN
SYSGEN> CONNECT CONSOLE
```

2 Exit from SYSGEN.

```
SYSGEN> EXIT
```

3 Mount the console volume.

```
$ MOUNT/FOREIGN CSA1:
```

4 Copy CIBOO.CMD to your default disk directory.

```
$ EXCHANGE COPY CSA1:CIBOO.CMD *.*
```

5 Edit the disk file in your default directory as shown in the
following example. The first line you add deposits the HSC
node number in register R2, and the second line you add
deposits the HSC disk's unit number in R3.

To set the system root at SYSA, edit the line commented as
software boot control flags by changing the most significant
digit in register R5 to the letter A. This digit determines the
system root for system disks with multiple systems.

Note that all numeric entries are made using hexadecimal
notation.

```
--          .
--
--
DEPOSIT R0 20           !CI PORT DEVICE
DEPOSIT R1 E            !CI TR=E
DEPOSIT R2 0C     !HSC NODE NUMBER
DEPOSIT R3 03     !DEVICE UNIT NUMBER
DEPOSIT R4 0            !BOOT BLOCK LBN (NOT USED)
DEPOSIT R5 A0004000  !DESIGNATED ROOT IS SYSA
--
--
--
```

6 If the disk is dual-ported (accessible to two HSC controllers),
you must specify the node number of both controllers in
register R2. Further, you must specify the higher-numbered
node in bits <15:8> and the lower-numbered node in bits
<7:0>. For example, if the disk is accessible to the HSC
numbered 0C and the HSC numbered 0A, the line for R2
should appear like this:

```
DEPOSIT R2 0C0A        !DUAL-PORTED HSC NODE NUMBERS
```

7 Copy the edited boot command file to the default boot
   command procedure (DEFBOO.CMD) on your console
   volume.

   ```
   $ EXCHANGE COPY device:[directory]CIBOO.CMD CSA1:DEFBOO.CMD
   ```

   You should provide another copy of this file in case
   DEFBOO.CMD is overwritten. Use the same command,
   but provide a distinctive file name for the other file.

8 Shut down the system.

   ```
   $ @SYS$SYSTEM:SHUTDOWN
   ```

9 Press CTRL/P to put the system into console mode.

10 Use the HALT command to stop the processor.

11 Boot the HSC disk by invoking the default boot command
   procedure:

   ```
   >>> B
   ```

4.3.3   Booting a VAX–11/750 During an Installation

This section describes how to boot an HSC disk on a VAX–11
/750 while installing a new operating system.

1 At the processor control panel, set the BOOT SELECT
   switch to position A.

2 Determine the unit number of the HSC disk and the node
   number of the HSC controlling it.

3 Press CTRL/P to put the system in console mode.

4 When you get the console prompt ($> > >$), invoke BOOT58
   with this command:

   ```
   >>> B/800 DDA0
   ```

   If you fail to add the /800 qualifier, the console program
   will try to execute DEFBOO.CMD.

5 When you get the BOOT58> prompt, deposit the HSC
   node number into register R2.

   ```
   BOOT58> D/G 2 node_number
   ```

Note that all numeric entries are made using hexadecimal notation. For example, if the HSC is node number 12 (hexadecimal 0C) on the CI750, you use this command:

```
BOOT58> D/G 2 OC
```

6 If the load device is accessible to two HSCs, deposit both node numbers in hexadecimal digits putting the larger number in hexadecimal digits 3 and 2, and the lesser in digits 1 and 0. For example, if one HSC is numbered 15 (hexadecimal 0E) and the other is numbered 10 (hexadecimal 0A), the proper command is this:

```
BOOT58> D/G 2 OEOA
```

7 Deposit the unit number of the load device into register R3.

```
BOOT58> D/G 3 unit_number
```

For example, if the load device is unit number 21, deposit a hexadecimal 15 into R3:

```
BOOT58> D/G 3 15
```

8 Boot the HSC disk.

```
BOOT58> @CIBOO.CMD
```

4.3.4　Booting on a Running VAX–11/750

This section describes how to set up DEFBOO.CMD to do a nonstop system boot of an HSC disk on a VAX–11/750 system that is up and running. In this example, the HSC device is an RA60 disk drive assigned unit number 03 on an HSC controller assigned node number 12 (hexadecimal 0C). The example assumes the user wants to boot the system from alternate root SYSA.

1 Set the BOOT SELECT switch to position A.

2 Be sure your console drive is connected. If it is not, invoke the System Generation Utility (SYSGEN) to connect it, as follows:

```
$ RUN SYS$SYSTEM:SYSGEN
SYSGEN> CONNECT CONSOLE
```

3 Exit from SYSGEN.

```
SYSGEN> EXIT
```

4 Mount the console volume.

```
$ MOUNT/FOREIGN CSA1:
```

5 Copy CIBOO.CMD to your default disk directory.

```
$ EXCHANGE COPY CSA1:CIBOO.CMD *.*
```

6 Edit the disk file in your default directory. The first line you
add deposits the HSC node number in register R2, and the
second line you add deposits the HSC disk's unit number in
register R3.

To set the system root at SYSA, you edit the line commented
as *software boot control flags* by changing the most
significant digit in register R5 to the letter A. This digit
determines the system root for system disks with multiple
systems.

Note that all numeric entries are made using hexadecimal
notation.

```
--
--
--
D/G R0 20              !CI PORT DEVICE
D/G R1 E               !CI TR=E
D/G R2 0C              !HSC NODE NUMBER
D/G R3 03              !DEVICE UNIT NUMBER
D/G R4 0                   !BOOT BLOCK LBN (NOT USED)
D/G R5 A0004000        !SOFTWARE BOOT CONTROL FLAGS
--
--
--
```

7 If the disk is dualported (accessible to two HSC controllers),
you must specify the node number of both controllers in
register R2. Further, you must specify the higher-numbered
node in bits <15:8> and the lower-numbered node in bits
<7:0>. For example, if the disk is accessible to the HSC
numbered 12 (hexadecimal 0C) and the HSC numbered 10
(hexadecimal 0A), the line for register R2 should appear like
this:

```
D/G R2 0C0A        !DUAL-PORTED HSC NODE NUMBERS
```

8 Copy the edited boot command to the default boot command procedure (DEFBOO.CMD) on your console volume.

```
$ EXCHANGE COPY device:[directory]CIBOO.CMD CSA1:DEFBOO.CMD
```

You should provide another copy of this file in case DEFBOO.CMD is overwritten. Use the same command, but provide a distinctive file name for the other file.

9 Write the boot block on the TU58 using the WRITEBOOT Utility.

a Invoke WRITEBOOT.

```
$ RUN SYS$SYSTEM:WRITEBOOT
```

b When WRITEBOOT asks for the target device, make the following entry:

```
Target system device (and boot file if not VMB.EXE):? CSA1:BOOT58.EXE
```

c In response to the next question, enter the number 1:

```
Enter VBN of boot file code (default is one): 1
```

d When you are asked for the address of the primary bootstrap, enter:

```
Enter load address of primary bootstrap in HEX (default is 200): C000
```

10 Shut down the system.

```
$ @SYS$SYSTEM:SHUTDOWN
```

11 Press CTRL/P to put the system in console mode.

12 When you get the console prompt (> > >), boot the HSC disk.

```
>>> B
```

4.4 Naming Devices

When you install software on a VAX processor, you are concerned primarily with two devices—the *load* device and the *system* device. The load device is the drive that holds the distribution media; it may be either a magnetic tape drive or a disk drive. The system device is the disk drive that holds the system disk.

In addition to these devices, you may have to use the block storage device that is part of your console subsystem. The block storage device is used to hold stand-alone BACKUP when you install a system from magnetic tape. It is also used to hold the distribution media when you install updates or add optional products to your system.

When you install software, you must provide the procedure with device names. For example, when you restore the required save set, you have to specify the device names of the load device and the system device. Later, the installation process prompts you for the name of the load device when it is ready to restore the rest of the operating system. You also use the device name, or a modified version of it, when you want to bootstrap software from a device other than the default device.

As you do the installations, you will see references to the device name format, *ddcu*. When you see this, you are being told to specify a device using the following format:

dd   A 2-letter code that represents the device type. For example, an RK07 disk drive is device type **DM**, an RA60 disk drive is device type **DJ**, and a TE16 magnetic tape drive is device type **MT**. Table 4–1 lists all the device codes.

c   A 1-letter code that represents the hardware controller for the device. Controllers provide the interface for the various devices that connect to the processor. Specifically, the first controller on a bus is designated the A controller, the next is the B controller, and so forth. For example, if you are installing Version 4.0 on a system configured with dual-RA60 disk drives attached to UDA controller A, the distribution disk might be mounted on device DJA1, and the new system volume mounted on DJA0.

u   A 1-letter code that represents the unit address of the
       drive. The first drive on a controller is usually assigned
       unit address 0 and each succeeding drive is assigned the
       next higher unit address, 1, 2, and so forth. However,
       this is not a constraint and you are permitted to assign
       unit numbers as needed. You are limited to the range
       of acceptable unit numbers for a device. For example,
       most MASSBUS disks may be assigned unit numbers in
       the range 0 to 7, but UDA-supported disks may have
       unit numbers in the range 0 through 255. For installation
       purposes however, unit numbers are limited by the set of
       boot command procedures on your console media.

To assign a unit number to a device, simply insert the
appropriately inscribed logical address plug. (See Chapter 3
for details on the use of logical address plugs.)

Table 4–1 presents the device codes for all the devices you use
in doing installations and upgrades. Note that you must use
device code DB when booting a devices coded as DR.

Table 4–1   Device Codes

Code	Device
CS	Console storage devices (TU58 or RX01)
DA	RC25 disk drive
DB	RP05 and RP06 disk drives
DD	Console storage device (TU58 on VAX–11/750 only)
DL	RL02 disk drive on VAX–11/750
DM	RK06 and RK07 disk drives
DQ	RL02 and R80 disk drives on VAX–11/730
DR	RM03, RM05, RM80, and RP07 disk drives
DJ	RA60 disk drive
DU	RA80 and RA81 disk drives
MF	TU78 magnetic tape drive
MS	TS11 and TU80 magnetic tape drives
MT	TE16, TU45 and TU77 magnetic tape drives
MU	TU81 magnetic tape drive

The only part of the name you can readily modify is the
unit number. The device type is fixed and the controller
determination is made when the hardware is installed.

Some examples of device names are presented below.

Device	Name
Second RK07 disk drive on UNIBUS controller A	DMA1
First TE16 magnetic tape on MASSBUS controller A	MTA0
Eighth RA81 disk on UDA controller A	DUA7

For all VAX systems except the VAX-11/785 and the VAX-11
/750, you omit the controller designation when you specify the
boot command procedure for a particular device. For example,
if you have a VAX-11/780 and you want to boot the operating
system from the first RP06 disk drive MASSBUS controller
A, you would use the format *ddu* as shown in the following
example:

```
>>> B DB0
```

Here, the device type is specified as DB and the unit address
is 0. The primary bootstrap program uses this code to invoke
the appropriate boot command procedure. In this example, the
primary bootstrap program converts DB0 to DB0BOO.CMD and
then calls the command procedure.

4.4.1   Boot Names for the VAX-11/785

The names used to invoke bootstrap command files on
the VAX-11/785 are somewhat different from other VAX
processors. You may use one of three forms to invoke a boot
command file on the VAX-11/785.

The first form specifies the device type and the unit.

```
>>> B ddu
```

The invoked boot command procedure includes a default
controller value of A (first controller on the bus) and the
unit number you specify. For example, assume you use this
command:

```
>>> B DM5
```

The sixth RK07 on the A controller will be booted.

The second form provides the device type and controller designations, but omits the device's unit designation:

```
>>> B ddc
```

Here, the invoked boot command procedure takes the controller literal you provide, but before you use this form you must provide the unit number. You do this by using a console DEPOSIT command of this form:

```
>>> DEPOSIT R3 unit_number
```

For example, if you want to boot from an RP06 on controller C that is assigned unit number 3, you would use this sequence of commands:

```
>>> DEPOSIT R3 3
```

```
>>> B DBC
```

The third form provides neither the controller nor the unit designations:

```
>>> B dd
```

The invoked boot command procedure includes a default controller value of A (first controller on the bus) and the a default unit number of 0 (first unit on the controller). For example, if you want to boot from the first disk drive on the UDA, you could use this command:

```
>>> B DU
```

4.4.2   Boot Names for the VAX–11/750

Boot names for the VAX–11/750 require both the controller designation and the unit number.

```
>>> B ddcu
```

For example, if you want to boot the system from the third drive on the first UDA, you might use this command:

```
>>> B DUA0
```

4.4.3   VAX–11/780 Device Names

Device codes for VAX–11/780 installations (including VAX–11
/785) are listed in Table 4–2. Note that you use device code
DB when booting a device coded as DR.

Table 4–2   VAX–11/780 Device Codes

Code	Device
CS	Console storage device (RX01)
DB	RP05 and RP06 disk drives
DR	RM03, RM05, RM80, and RP07 disk drives
DM	RK07 disk drive
DJ	RA60 disk drive
DU	RA80 and RA81 disk drives
MF	TU78 magnetic tape drive
MS	TS11 and TU80 magnetic tape drives
MT	TE16, TU45 and TU77 magnetic tape drives
MU	TU81 magnetic tape drive

All the tape drives and the floppy diskette drive are used solely
as load devices. The RK07 and the RA60 may be used either
as load devices or as system devices depending on system
configuration. All of the other disk drives are used solely as
system devices.

4.4.4   VAX–11/750 Device Names

Table 4–3 lists the device codes that may be applicable when
installing various VAX–11/750 system configurations. Note
that you use device code DB when booting a device coded as
DR.

Table 4–3   VAX–11/750 Device Codes

Code	Device
DB	RP05 and RP06 disk drives
DD	TU58 cassette drive
DL	RL02 disk drive
DM	RK07 disk drive
DR	RM03, RM05, RM80, and RP07 disk drive
DJ	RA60 disk drive
DU	RA80 and RA81 disk drives
MF	TA78 and TU78 magnetic tape drives
MS	TS11 and TU80 magnetic tape drives
MT	TE16, TU45, and TU77 magnetic tape drives
MU	TU81 magnetic tape drive

All of the tape drives and the RL02 disk drive are used solely as load devices. The RK07 and the RA60 may be used either as load devices or as system devices depending on system configuration. All of the other disk drives are used solely as system devices.

4.4.5   VAX–11/730 Device Names

Table 4–4 lists the device-type codes that may be applicable to installation of various VAX–11/730 system configurations.

Table 4–4   VAX–11/730 Device-Type Codes

Code	Device Type
CS	Console storage device (TU58)
DQ	RL02 and R80 disk drives
DJ	RA60 disk drive
DU	RA80 and RA81 disk drives
MS	TS11 and TU80 magnetic tape drives
MU	TU81 magnetic tape drive

All of the tape drives are used solely as load devices. In dual-
RL02 IDC configurations, the load device and the system
device are RL02 disk drives. Similarly, the dual-RA60 UDA
configuration uses RA60 disk drives as both the load and
system devices. The R80, RA80, and RA81 are fixed-head
devices used solely as system devices.

4.4.5.1	**VAX–11/725 Device Names**

The only two devices in a VAX–11/725 system are the fixed
disk and the removable disk. Configurations of pre-Version 4.0
systems use the device name DUA0 for the removable disk,
and DUA1 for the fixed disk. For Version 4.0 systems, the
removable disk is called DAA0 and the fixed disk is DAA1.
Note however, the boot names are DU0 and DU1 for all
versions.

4.5 Backing Up a System Disk

Before you upgrade or update your system, or before you
install an optional product, you should back up your current
system disk. This provides you with a backup of your system
if a problem occurs during the installation. The recommended
method is to use BACKUP to make a copy of the system disk,
and to then use the copy as the target disk for your installation.

You may use either online BACKUP or stand-alone BACKUP
to make a copy of your system disk. Usually, stand-alone
BACKUP will take longer but is safer and less complex than
using online BACKUP. If you use online BACKUP, the system
may continue in use while you are making the copy. With
stand-alone BACKUP, all use of the system disk is discontinued
because only stand-alone BACKUP runs in memory during the
backup procedures.

This section assumes the use of stand-alone BACKUP. If
you received your operating system on magnetic tape, it was
accompanied by stand-alone BACKUP on three pieces of
console media. If you received your system on disk, you must
build a bootable stand-alone BACKUP kit (see Section 4.6).

4.5.1   Backing Up a VAX–11/780 System Disk

To back up your system disk to a scratch disk or tape, use stand-alone BACKUP floppies. The procedure is as follows:

1 Shut down your system.

2 Write-protect your system disk by pressing the WRITE PROTECT button.

3 Boot stand-alone BACKUP from the floppy diskettes.

4 Place a scratch volume on an appropriate drive.

5 Issue one of the following stand-alone BACKUP commands. The first command is for disk devices; the second for tapes.

```
$ BACKUP/VERIFY/IMAGE system_disk: scratch_disk:
$ BACKUP/VERIFY/IMAGE system_disk: tape_device:save_set_name
```

For example, if the system disk were DMA0 and the scratch disk were DMA1, you would issue this command to back up the system disk.

```
$ BACKUP/VERIFY/IMAGE DMA0: DMA1:
```

4.5.2   Backing Up a VAX–11/750 System Disk

To back up your system disk to a scratch disk or tape use stand-alone Backup on TU58 cassettes. The procedure is as follows:

1 Shut down your system.

2 Write-protect your system disk by pressing the WRITE PROTECT button.

3 Boot stand-alone BACKUP from the TU58 cassettes.

4 Place a scratch volume on an appropriate drive.

5 Issue one of the following BACKUP commands. The first command is for disk devices; the second for tapes.

```
$ BACKUP/VERIFY/IMAGE system_disk: scratch_disk:
$ BACKUP/VERIFY/IMAGE system_disk: tape_device:save_set_name
```

For example, if system disk were DMA0 and the scratch disk were DMA1, you would issue this command to back up the system disk.

```
$ BACKUP/VERIFY/IMAGE DMAO: DMA1:
```

4.5.3   Backing Up an R80 System Disk on a VAX–11/730

Because an R80 disk has a much greater storage capacity than an RL02 disk, it is usually necessary to back up the contents of an R80 disk to a group of RL02 disks that are referred to as the backup volume set. This procedure is described in the *Guide to VAX/VMS System Management and Daily Operations.*

Use this procedure to back up your R80 system disk:

1 Boot stand-alone BACKUP.

2 Assign device name DQA0 to the R80 drive and be sure it is write protected.

3 Assign device name DQA1 to the RL02 drive that will hold the scratch disks, that is, the disks you are using to back up the R80.

4 Place a scratch RL02 disk in DQA1.

5 Specify the name and file type of the save set you are creating:

```
$ BACKUP/VERIFY DQAO: DQA1:filename.type/SAVE_SET
```

When stand-alone BACKUP is ready for the next scratch disk, it will prompt you with the following message:

```
%BACKUP-I-READYWRITE, Mount Volume 2 on _DQA1: for writing
Press RETURN when ready:
```

6 Place the next RL02 scratch disk (volume 2 of the backup volume set) in the drive and press the RETURN key.

Repeat this step each time you receive the prompt for another volume. When the volume set is completed, you are returned to the DCL level (as indicated by the dollar-sign prompt ($).

4.5.4   Backing Up an RL02 System Disk

To do a full backup of a RL02 system disk, perform the following steps.

1 Physically load the system disk in the RL02 drive designated DQA0.

2 Write-protect DQA0.

3 Boot stand-alone BACKUP from the TU58 cassettes.

4 Place a scratch disk in DQA1.

5 Back up the system disk to the scratch disk with the following command:

```
$ BACKUP/IMAGE/VERIFY DQA0: DQA1:
```

4.5.5   Backing Up a Fixed RC25 System Disk

Use the following procedure to back up a fixed RC25 system disk:

1 Replace the removable data disk with a removable scratch disk.

2 Boot stand-alone BACKUP from the TU58 cassettes.

3 When you get the DCL prompt ($), back up the fixed system disk to the removable scratch disk:

```
$ BACKUP/IMAGE/VERIFY DAA1: DAA0:
```

The removable disk is now a backup of the fixed system disk.

4.5.6   **Backing Up a Removable RC25 System Disk**

Use the following procedure to back up a removable system disk:

1 Replace the removable system disk (the disk you want to back up) with a removable scratch disk.

2 Boot stand-alone BACKUP from the TU58 cassettes.

3 When you get the DCL prompt ($), back up the fixed data disk to the removable scratch disk by issuing this command.

```
$ BACKUP/IMAGE/VERIFY DAA1: DAA0:
```

The removable disk is a backup of the fixed data disk. Put it aside until you are instructed to use it for restoring the fixed disk.

4 Mount the original removable system disk.

5 Back up the removable system disk to the fixed disk.

```
$ BACKUP/IMAGE/VERIFY DAA0: DAA1:
```

The fixed disk is a backup of the removable system disk.

6 Replace the the original removable system disk with another removable scratch disk.

7 Back up the fixed disk to the removable disk.

```
$ BACKUP/IMAGE/VERIFY DAA1: DAA0:
```

The removable disk is a backup of the original removable system disk. Put it aside and replace it with the removable disk that was used to back up the fixed data disk in step 3.

8 Back up the removable disk to the fixed disk.

```
$ BACKUP/IMAGE/VERIFY DAA0: DAA1:
```

The fixed disk is restored to its original state as the data disk.

9 Replace the removable disk with the original removable system disk.

4.6 Building Stand-Alone BACKUP on Console Media

To back up your system disk, you may use either online Backup or stand-alone BACKUP. If you elect to use stand-alone BACKUP and your system was not distributed on magnetic tape, you will have to build stand-alone BACKUP on either console media or on disk. This section describes how to build stand-alone BACKUP on console media.

To build a stand-alone BACKUP kit, you invoke the STABACKIT.COM command procedure in directory SYS$UPDATE. The following procedure for building stand-alone BACKUP on floppy diskettes can also be used to build the kit on TU58 cassettes, except for noted differences. The most notable difference is that the term *TU58 cassette* must be substituted for *floppy diskette*.

If you have a VAX–11/780, the console media is a diskette. For all other VAX systems, the console media is a TU58 cassette.

To build stand-alone BACKUP kit on floppy diskettes, proceed as follows:

1 Label three blank RX1 floppy diskettes SYSTEM_1, SYSTEM_2, and BACKUP, respectively.

2 Log in as system manager.

3 At the DCL prompt ($), enter the following:

    $ @SYS$UPDATE:STABACKIT

4 When STABACKIT executes, it prompts you for for the target device. Respond by entering CSA1: as shown:

    Enter the name of the device on which to build the kit: CSA1:

5 When you identify the target device, STABACKIT tells you that three blank floppies are needed to build the kit, two for the stand-alone VAX/VMS files, and the other for the BACKUP application image.

STABACKIT also gives you the option of building the VAX/VMS files as a system kit without building the application (in this case BACKUP) kit, or building both the system and application kits at this time.

4–23

```
Do you want to build the system kit? [Yes/No, default Yes]:
Do you want to build the application kit? [Yes/No, default Yes]:
```

Respond to both questions by pressing the RETURN key.

6 Next, STABACKIT gives you the option of using the Bad
Block Locator Utility (BAD) to check for bad blocks on the
target diskette. This operation requires approximately 15
additional minutes for each diskette but by finding bad
blocks that would otherwise give you a problem building
the kit, you may actually save time.

Note: **If you are building a kit on TU58 cassettes, allow 40
minutes per cassette to run BAD.**

Next, wait for the prompt that asks you to insert the first
system floppy in the console drive:

```
Please place the first system floppy diskette in drive _CSA1:.
This volume will receive the volume label SYSTEM_1.

Enter "YES" when ready:
```

7 Insert the floppy labeled SYSTEM_1 into CSA1; then press
the RETURN key. If you elected to have the bad block test
run, you get the following message:

```
Analyzing floppy diskette for bad blocks . . .
SYSTEM_1     mounted on _CSA1:
```

The test takes approximately 15 minutes (40 minutes for
TU58s). Upon completion, you are given the number of
blocks on the diskette and the number that are bad. Then
STABACKIT mounts the floppy, creates [SYS0.SYSEXE],
and copies the set of system files to it, accompanied
by appropriate messages. When the last file is copied,
STABACKIT prompts you for the next floppy:

```
Please place the first system floppy diskette in drive _CSA1:.
This volume will receive the volume label SYSTEM_2.

Enter "YES" when ready:
```

8 Replace the floppy diskette labeled SYSTEM_1 with the
floppy diskette labeled SYSTEM_2, then press the RETURN
key.

9 When you press the RETURN key, STABACKIT runs BAD
(assuming you chose to do so), mounts the floppy and
copies the rest of the system files to [SYS0.SYSEXE].

When the last file is copied, STABACKIT prompts you for the third floppy:

```
Please place the application floppy diskette in drive _CSA1:.
This volume will receive the volume label BACKUP.
Enter "YES" when ready:
```

10 Replace the floppy diskette labeled SYSTEM_2 with the floppy diskette labeled BACKUP; then press the RETURN key.

11 When you press the RETURN key, STABACKIT runs BAD, mounts the floppy and copies the executable backup image (STABACKUP.EXE) to [SYS0.SYSEXE].

12 When you get this message, the kit is built:

```
The console volume will be mounter /NOWRITE for protection.
Please replace the original console floppy diskette.
Enter "YES" when ready:
```

13 Replace the floppy labeled BACKUP with the console floppy, then press RETURN.

The procedure responds by mounting the console floppy, giving the starting and ending times, and then returning the system to DCL command level.

4.7   Using Stand-Alone BACKUP on Disk

Installing and using stand-alone BACKUP in an alternate root on your system disk saves time when backing up your system disk because you do not have to boot stand-alone BACKUP from your console volume, which is a relatively slow process.

The procedures described here apply to all VAX processors. However, you should note the following:

- You can install stand-alone BACKUP in any available disk directory root from SYS1 through SYSE; however, DIGITAL has established SYSE as the standard alternate system directory root for stand-alone BACKUP. The discussion in this section assumes you will install stand-alone BACKUP in SYSE.

- You cannot use these procedures to restore your operating system to the system disk from which stand-alone BACKUP is booted.

- There are some limitations with regard to the VAX–11/750, which are noted within the procedures.

- The console device is CSA1 on VAX systems with a single console drive and CSA2 on systems with two console drives. See Chapter 2 for help in locating the console drives.

The rest of this section explains the following:

- How to install stand-alone BACKUP in alternate system directory root SYSE

- How to create a bootstrap command procedure that will let you boot stand-alone BACKUP from SYSE

- How to boot stand-alone BACKUP from SYSE

4.7.1 Installing Stand-Alone BACKUP in Alternate System Root SYSE

You install stand-alone BACKUP in SYSE using the STABACKIT command procedure. First, log in to the SYSTEM account. Then, enter this command:

```
$ @SYS$UPDATE:STABACKIT SYS$SYSDEVICE:[SYSE.SYSEXE]
```

Be sure to enter the command exactly as shown. You will not be prompted for the destination if you forget to enter it.

After you install stand-alone BACKUP in SYSE, the next step is to create a command procedure to boot it.

4.7.2   Creating a Command Procedure to Boot Stand-Alone BACKUP

The recommended procedure for creating a command procedure to boot stand-alone BACKUP from alternate root SYSE is to modify an existing boot command procedure. Ideally, this should be the default boot command procedure, DEFBOO.CMD.

First, make a copy of DEFBOO.CMD in your default device directory. Then, edit the copy as instructed. Finally, give the edited copy an appropriate file name, and store it on the console volume.

You may choose any unique name in the form xxxBOO.CMD for the command procedure you create. However, DIGITAL suggests you use a file name identical to the copied file, except that the first letter should be an **X**.

For example, if you use a copy of DEFBOO.CMD, the new file should be named XEFBOO.CMD.

The following procedure assumes you will use a copy of DEFBOO.CMD and rename it XEFBOO.CMD.

1 If necessary, use the following command sequence to configure the console device:

```
$ RUN SYSGEN
SYSGEN> CONNECT CONSOLE
SYSGEN> EXIT
```

2 Insert the console volume into the console device.

3 Invoke the DXCOPY command procedure make a copy of DEFBOO.CMD in your default disk directory by entering this command:

```
$ @SYS$UPDATE:DXCOPY
```

4 When the command procedure asks whether the console volume is mounted, enter the letter **Y**.

5 When DXCOPY asks if you want to copy a file from the console volume to your default directory, enter the letter **Y**.

6 When DXCOPY prompts you for it, enter the name of the
file you want to copy to your default directory, for example,
DEFBOO.CMD.

```
DXCOPY> Name of file to be copied?: DEFBOO.CMD
```

7 Exit from DXCOPY using the EXIT command.

8 Rename the copy of DEFBOO.CMD in your default
directory.

```
$ RENAME DEFBOO.CMD XEFBOO.CMD
```

9 Invoke an editor to modify XEFBOO.CMD. The table below
shows how to determine the line that you must modify
for your processor; it shows the line before and after you
change it.

Processor Type	Line to Change	Line After Change
VAX–11/780	DEPOSIT R5 0	DEPOSIT R5 E0000000
VAX–11/782	DEPOSIT R5 0	DEPOSIT R5 E0000000
VAX–11/750	D/G 5 0	D/G 5 E0000000
VAX–11/730	D/G/L 5 0	D/G/L 5 E0000000
VAX–11/725	D/G/L 5 0	D/G/L 5 E0000000

10 When you have edited the file, call DXCOPY to store it on
the console volume by entering this command:

```
$ @SYS$UPDATE:DXCOPY
```

11 When the command procedure asks whether the console
storage medium is mounted, enter the letter **Y**.

12 When DXCOPY asks whether you want to copy from the
console medium, enter the letter **N**.

This response tells DXCOPY you want to copy the file from
your default directory to the console volume.

13 When the command procedure asks for it, enter the name of the file to be copied from your default directory to the console storage medium. In this example, you enter XEFBOO.CMD:

```
DXCOPY>Name of file to be copied?: XEFBOO.CMD
```

DXCOPY responds by copying the modified file to the console volume.

14 Exit from DXCOPY.

4.7.3   Booting Stand-Alone BACKUP from SYSE

To boot stand-alone BACKUP from SYSE, you must perform the following steps.

1 Shut the system down.

```
$ @SYS$SYSTEM:SHUTDOWN.COM
```

2 The procedure asks the following questions. Respond by pressing the RETURN key.

```
How many minutes until final shutdown? [0]
Reason for shutdown? [Standalone]
Do you want to spin down the disk volumes? [NO]
```

3 As the shutdown continues, the console terminal prints several shutdown messages. When the console terminal prints the following message, shutdown is completed:

```
SHUTDOWN COMPLETE - USE CONSOLE TO HALT SYSTEM
```

At this point, halt the processor using CTRL/P.

4 When the processor halts, enter the following command to boot stand-alone BACKUP from SYSE:

```
>>> B XEF
```

5 If you are not using a modified version of the default bootstrap command procedure to boot stand-alone BACKUP, use the first three characters in the name of the bootstrap command procedure you created. For example, if you are using XU0BOO.CMD, type the following command:

```
>>> B XU0
```

5 VMSINSTAL Command Procedure

You use the VMSINSTAL command procedure (VMSINSTAL.COM) to upgrade your system to the most recent major version, to install system maintenance updates, and to install many optional software products. Some optional software products, however, must be installed using the VMSUPDATE command procedure until they are adapted for VMSINSTAL. See Appendix C for information on VMSUPDATE.

This chapter describes the following:

- How to prepare your system for the installation

- How to invoke VMSINSTAL

- How to select the VMSINSTAL options appropriate to your needs

- What to do if the system fails during installation

- Special considerations for small-disk systems

- Error messsages

This chapter does not describe installation procedures specific to any upgrade, update, or product. The examples used are for illustration only. For installation details of a particular product, refer to the installation guide that accompanies it.

5.1 Preparing to Use VMSINSTAL

Before you begin the installation procedure, do the following:

1 Back up your system disk, and use the backup copy as a working copy for doing the installation. If you back up your system disk to magnetic tape, you will have to restore the tape to a Files–11 disk to get a working copy. The working copy has more contiguous space than the original system disk because of the way BACKUP creates the copy. This

additional contiguous space may be needed during the installation.

If the system fails during installation, your working copy may be left in an unusable state because VMSINSTAL may delete the older version of the product before it installs the newer version. You may have to make a new working copy of the system disk and restart the installation from the beginning if a failure occurs.

2 Log in at the console terminal under the SYSTEM account.

Note: **If SYSGEN parameters MOUNTMSG or DISMOUMSG are set to 1, you will receive a message from OPCOM each time a disk or tape is mounted or dismounted. If these messages are not disabled, and you are installing from an operator's terminal, they will appear within 30 seconds of each mount or dismount.**

3 Be sure that all users are logged out and that all batch jobs are completed.

4 Keep users off the system with the following command:

```
$ SET LOGINS/INTERACTIVE=0
```

5 Shut down DECnet–VAX (if your system includes it). First invoke the Network Control Program (NCP):

```
$ RUN SYS$SYSTEM:NCP
```

When the console terminal prints the NCP prompt (NCP>), enter the following command:

```
NCP> SET EXECUTOR STATE SHUT
```

DECnet–VAX will perform an orderly shutdown and the OPCOM facility will notify you when DECnet–VAX is off. Exit from the NCP by typing EXIT at the prompt.

6 Be sure the following limits in the SYSTEM account authorization record are equal to or greater than those specified here:

```
Buffered byte count quota (BYTLM) = 18000
Enqueue quota (ENQLM) = 30
Direct I/O limit (DIOLM) = 18
Buffered I/O limit (BIOLM) = 18
Open file quota (FILLM) = 20
AST limit (ASTLM) = 24
```

To check these limits, run the Authorize Utility
(AUTHORIZE) by entering the following command:

```
$ RUN SYS$SYSTEM:AUTHORIZE
```

When the system responds with the UAF> prompt, enter
this command:

```
UAF> SHOW SYSTEM
```

AUTHORIZE responds by displaying the current limits of
the SYSTEM account's user authorization file.

7 If the current limits need to be modified, use the
AUTHORIZE command MODIFY in the following form
to make the change:

```
UAF> MODIFY SYSTEM/limit=new_value
```

For example, to increase the DIOLM limit to 18, you use the
following command:

```
UAF> MODIFY SYSTEM/DIOLM=18
```

8 When you have completed the modifications, return to DCL
command level by entering the word EXIT when you get
the UAF> prompt. Note that the modification will not take
effect until you log out and log in again. (See the *VAX/VMS
Utilities Reference Volume* for a description of the Authorize
Utility.)

9 If your system configuration is tailored, load the library disk
onto a drive before starting. The library disk has files that
are needed to run VMSINSTAL.

10 Physically mount the distribution media containing the
software product(s).

5.2 Invoking VMSINSTAL

This section tells you how to invoke VMSINSTAL and describes options that can be used. Before you begin the installation, be sure you read and understand the installation procedures for the specific product or update.

When you invoke VMSINSTAL, it asks if you are satisfied with the backup of your system disk. Be certain that the backup copy is satisfactory before you continue with the installation.

If you have not satisfied all conditions required to begin the installation when you invoke VMSINSTAL, you will get a warning message explaining the problem together with a prompt asking if you want to continue. DIGITAL strongly recommends that you correct these conditions before you try to invoke VMSINSTAL again.

Note: If you continue without making the required corrections, the resulting installation may not be supported by DIGITAL.

To invoke VMSINSTAL, use a command with the following syntax:

```
@SYS$UPDATE:VMSINSTAL product_list source: OPTIONS option-list [root]
```

product-list   The list of the product(s) you want to install. If you use a wildcard character (*) as the product-list parameter, all versions and updates of all products from the distribution source will be installed in chronological order.

source	Identifies the source of the optional software product as one of the following:

                  • A device drive that holds the distribution medium; for example, the TU58 drive designated CSA1: on a VAX–11/750

                  • A disk directory where the product save set has been transferred from the distribution media

                  • A disk directory on another node

               You may also use a logical name to specify the source. If you do not specify the source, VMSINSTAL will prompt you for it.

OPTIONS	A keyword that precedes the list of options to be installed. If you enter an option list and the OPTIONS keyword is not included in the command, you will receive an error message and the installation will terminate.
option-list	A list of the options requested. See Section 5.3 for details.
root	You must select this option if you want to install the product in an alternate system root and it must be in the following format:

               `ddcu: [SYSn.]`

ddcu	Is the device on which the alternate root resides.
[SYSn.]	Is the top-level directory of the alternate root.

               If you wish, you may specify a previously defined logical name for the alternate root.

If you want to specify more than one item in the product list parameter, you must separate the items using commas and no intervening spaces. The format for the product list is:

`productvvu`

product	Is the product facility name (1 through 36 alphanumeric characters)
vv	Is the major version number (2 digits)
u	Is the update number (1 digit)

For example, if you update your operating system to VAX/VMS

Version 4.0, the product name (product) is VMS, the major
version (vv) is 04, and the update number (u) is 0. Therefore
the product-list parameter is VMS040.

If you omit the product-list parameter, you will be prompted
for it. If you are installing from a distribution device, the list of
products on your distribution volume is included with the bill
of materials for the distribution kit. If the list is not available,
issue a DIRECTORY command for the distribution volume to
find the products that are included.

If you are installing from a disk directory, you can obtain the
product list by entering a DIRECTORY command specifying the
disk directory in the following format. If you are accessing a
remote node, you must have READ and EXECUTE (R,E) access
to the directory.

```
$ DIRECTORY node::disk:[directory]
```

Using this format, you can install a specific version and update
of a product from a distribution volume containing several
versions and updates. If you do not include a version or update
number, then all versions and updates from the source are
installed in chronological order.

When you invoke VMSINSTAL, you receive several prompts
and messages that direct and explain the installation. These
prompts and messages differ, depending on the update or
software product you are installing. Consult the installation
documentation that comes with the update or software product
for specific instructions. If you need assistance during an
installation, typing a question mark (?) at a prompt will give
you an explanation of acceptable responses.

When the installation is completed, VMSINSTAL will do one of
two things, depending on the requirements of the product you
have installed. It will either perform an automatic shutdown
of the system and instruct you to manually reboot, or it will
simply return you to the DCL command level.

When the product is installed, you should back up the updated
system disk.

5.3 Choosing VMSINSTAL Options

The VMSINSTAL command procedure permits the use of four options:

* Auto-answer option (A)

* File log option (L)

* Alternate root option (R)

* Get option (G)

You specify each of these options by using a single letter (A, G, L, R) following the keyword OPTIONS in the command line that invokes VMSINSTAL.

Auto-answer

The auto-answer option makes it easier to *reinstall* a product or a system update by providing responses to VMSINSTAL questions and prompts during the reinstallation. The auto-answer option is used most often for reinstalling products when you upgrade the system.

If you specify the auto-answer option when you install a product, it creates an answer file of the form *product.ANS* in the SYS$UPDATE directory, where product is a file name that reflects the product-list parameter you provide when you invoke VMSINSTAL.

The file type of answer file is ANS. The answer file is used to record your responses to questions and prompts from VMSINSTAL during the installation. For example, if you install the product, NEWAID, with the auto-answer option, VMSINSTAL creates an answer file designated NEWAID.ANS.

When you subsequently reinstall the product and specify the auto-answer option (typically after upgrading your operating system), VMSINSTAL reads the answer file instead of prompting you for responses.

If you want to create a new answer file when you reinstall a product, you must delete the existing answer file first.

File Log

The file log option logs all file activity to the terminal during installation. File activity is defined as any action that alters the disposition of a file, such as creating a new file, updating a library, or deleting a file.

Alternate Root

The alternate root option permits you to install the product to a system root other than that of the running system. The VAX/VMS system in the alternate root must be complete and at the same version/update level as the running system. All files and software products that are referenced by the product installation must be present in the alternate root. Note that not all optional software products allow installation to a system root other than that of the running system. Consult the documentation of the specific product to determine whether the product can be installed to an alternate system root.

GET product

Installing products from a distribution tape or from console media directly onto your system is a time-consuming task. The GET product option saves you time by allowing you to temporarily store product save sets in a disk directory or on a magnetic tape. At your convenience the product save sets can be restored to your system from disk rather quickly in stand-alone mode using VMSINSTAL.

You may want to consider dedicating a user disk on a central node for storing product save sets to give other licensed system users fast access to the product save-set directory.

To store a product save set on a disk directory using the GET option, is a command with the following syntax:

```
$ @SYS$UPDATE:VMSINSTAL product-list source:
_OPTIONS G device:[directory]
```

You specify the disk directory immediately following the OPTIONS G parameter. For example, if you are installing a product named NEWAID from the VAX–11/750 console drive into disk directory USER1:[PRODUCTS], you would use the following command:

```
$ @SYS$UPDATE:VMSINSTAL NEWAID CSA1: OPTIONS G
_USER1:[PRODUCTS]
```

VMSINSTAL will create one or more files to store the product save set in the disk directory. The file name will have the following format:

productvvu.x

product The product name (1 to 36 alphanumeric characters)

vv   The major version number (2 digits)

u   The update number (1 digit)

x   A literal extension that is used to identify save set files, where A is the first save set, B the second, and so forth.

For example, if you are storing update 1 to Version 2.0 of the example product, NEWAID, and this requires four save sets, VMSINSTAL would create the following four files:

> NEWAID021.A
>
> NEWAID021.B
>
> NEWAID021.C
>
> NEWAID021.D

When you want to install the product on your system, you use the following format:

$ @SYS$UPDATE:VMSINSTAL product-list device:[directory]

For the example product NEWAID, you would use this command:

$ @SYS$UPDATE:VMSINSTAL NEWAID USER1:[PRODUCTS]

VMSINSTAL responds by installing the NEWAID product to your system disk.

5.4 Recovering From a System Failure

If the system fails during installation of an update or optional software product, VMSINSTAL will attempt to continue the installation upon rebooting. Depending on the point in the installation at which the system failed, one of three conditions will exist:

- The system disk will not have been altered before the system failure. In this case, VMSINSTAL will instruct you to restart the installation.

- The system disk or a library used by the installation will have been corrupted. In this case, VMSINSTAL instructs you to restore either the system disk or the corrupted library from the backup copy (see Section 5.1) and restart the installation.

- VMSINSTAL will continue the installation. In this case, VMSINSTAL performs most of the installation and then informs you that there may be some tasks you must perform manually to complete the installation. If VMSINSTAL instructs you to do so, reboot the system, log in as system manager, and perform the following tasks:

  — Purge all system files that have been replaced, even if you requested an automatic purge.

    ```
    $ PURGE/LOG SYS$SYSROOT:[*...]*.*
    ```

  — If you have a tailored system, restore all library files to the system disk because VMSINSTAL removes them during the installation. To restore your library files, mount the library disk and use the following command to make a copy request:

    ```
    $ @SYS$UPDATE:VMSTAILOR COPY VMIORIG
    ```

    After the copy command is executed, use the following command to delete the tailoring file:

    ```
    $ DELETE SYS$UPDATE:VMIORIG.TLR;*
    ```

    Both commands are necessary to return your system disk to its previous state when a failure occurs during an installation.

5.5 Special Considerations for Small-Disk Systems

A small disk is a disk whose storage capabilities (fewer than 70,000 blocks) make it impractical to use as a system disk for storing a complete operating system. Currently, small-disk systems include:

- Dual-RL02 systems

- Dual-RK07 systems

- RC25 systems

DIGITAL recommends that each of these system configurations use a tailoring environment in which two disks are used to support the operating system—a system disk and a library disk. The system disk includes all of the files required for basic operating system functions, and a limited set of library files that you can select that support additional operating system functions. The procedure for installing the operating system on a small-disk system has two parts:

- Restoring the required save set to the system disk

- Restoring the library files save set to the library disk

When the restorations are completed, you may tailor your system by copying selective files from the library disk to the system disk. (See the *Guide to VAX/VMS System Management and Daily Operations* for details on how to use the tailoring facility.)

5.5.1   Installing Optional Products on Small-Disk Systems

When you install optional products on a small-disk system, the library disk must be mounted. Before being prompted for the first optional software volume, you will receive the following message:

```
%VMSINSTAL-I-SMALLDISK, This is a small-disk system.

This is a small-disk system and must be tailored to accommodate your new
software products. The current tailoring environment will be recorded, and
the system files that are not required will be temporarily removed from the
system disk. After installation, the original environment will be restored.

%MOUNT-I-MOUNTED, VAXVMSLB4 mounted on _ddcu:
n library files recorded in SYS$SYSROOT:[SYSUPD]VMIORIG.TLR;
m revised library files recorded in SYS$SYSROOT:[SYSUPD]VMIDIFF.TLR;

The current tailoring environment has been recorded in VMIORIG.TLR.  If a
power failure or drastic error occurs during this installation, you can
restore the environment by entering the following commands after mounting
the library disk.  Please see your documentation for more details.

$ @SYS$UPDATE:VMSTAILOR COPY VMIORIG
$ DELETE SYS$UPDATE:VMIORIG.TLR

(n blocks were made available by tailoring.)
Deleting tailored software configuration.
```

The tailored software configuration is automatically restored when the installation is completed. The number of library files recorded depends on the number of library files on your system disk when the installation begins. If any revised library files

are recorded, refer to Section 5.5.2 for instructions on how to resolve file conflicts.

When you have completed the installation, you will receive a message that announces the restoration of the original software configuration.

If you do not have enough space left on the system disk to restore your original tailored environment, you will receive the following message:

```
There are not enough free blocks left on the system disk to restore your
original tailoring environment.  The environment has been recorded and
may be restored manually (as described above) after sufficient space has
been obtained by purging and/or deleting files.
Only x blocks of the required y are available!
```

Variables x and y represent the number of free blocks on the system disk after the installation and the total number of blocks in the original software configuration, respectively.

You should free enough space on the system disk to accommodate the total software configuration, and then restore it with the tailoring command COPY, as shown below:

```
Tailor> COPY VMIORIG
```

Alternatively, you may choose to reconfigure a new system disk that is compatible with the available disk space. (See the *Guide to VAX/VMS System Management and Daily Operations* for more information about the tailoring facility.)

5.5.2 Handling Revised Files Detected During Installation

Revised files should not be detected during most optional software installations. However, if revised files are found, you will receive an appropriate message followed by a listing of the revised files. Here is an example:

```
10 revised library files recorded in SYS$SYSROOT:[SYSUPD]VMIDIFF.TLR;.
%VMSINSTAL-E-LIBDIFF, The following system and library disk files do not match.
[SYSEXE]ANALIMDMP.EXE
[SYSEXE]ANALYZBAD.EXE
[SYSEXE]AUTHORIZE.EXE
[SYSEXE]BOOT58.EXE
[SYSEXE]BOOTBLOCK.EXE
[SYSEXE]CDU.EXE
[SYSEXE]DISKQUOTA.EXE
[SYSEXE]EDT.EXE
[SYSEXE]EVL.EXE
[SYSEXE]EXCHANGE.EXE
%VMSINSTAL-F-LIBDIFF2,  You must resolve these differences before continuing.
%VMSINSTAL-F-UNEXPECTED, Installation terminated due to unexpected event.
```

As the last line of the message states, the installation is terminated and you will have to determine which of the revised files you want to keep and which you want to delete before you attempt to do the installation. Then you must remove all the named files from the system disk so VMSINSTAL can proceed when reinvoked.

5.6   Error Messages

This section describes the error messages issued by the VMSINSTAL command procedure. Each message consists of an abbreviation followed by a text message. The list includes an explanation of each message, and appropriate user action.

ACCOUNT, This installation creates an account named *name.*

Explanation: The product being installed has created (or updated) an account with the specified name. The user authorization file (SYSUAF) has been updated accordingly.

User Action: None.

ACTIVE, The following processes are still active:

Explanation: VMSINSTAL checks to ensure that no user processes are active when an installation is begun. This check has failed, and VMSINSTAL lists those processes that are still active.

User Action: You will be asked if you want to continue the installation. It is recommended that you remedy this situation before continuing.

AUTOSYNC, Auto-answer file is not synchronized to the current question

Explanation: You have specified the auto-answer option, and VMSINSTAL is using the answers from an existing answer file. The question being asked by the installation procedure does not correspond to the next question and answer in the answer file.

User Action: The installation procedure must have changed. Delete the existing answer file and begin the installation again.

BACKUPERR, An error has occured in recreating the save set

Explanation: A BACKUP error was detected while recreating the save sets in the processing of the GET option. The operation will be aborted.

User Action: Be sure the specified output device has media which can be written.

BADCONDEV, Please specify an existing console device.

Explanation: You have specified a console device on which the console is to be mounted, but the device specified is not a console device.

User Action: Specify an existing console device.

BADDISDEV, Please specify a disk or tape device.

Explanation: You have specified a device on which the distribution volumes are to be mounted, but the device is neither a disk nor a tape.

User Action: Specify a disk or tape device.

BADDISDIR, Directory *directory* **does not exist.**

Explanation: You are trying to restore the distribution kit from a nonexistent directory.

User Action: Specify the correct directory when you invoke VMSINSTAL.

BADGETDIR, Directory *directory* **does not exist.**

Explanation: You are trying to copy the specified product save set to a nonexistent directory.

User Action: Specify the proper directory when you invoke VMSINSTAL.

BADLIBDEV, Please specify an explicit disk drive.
Explanation: For tailored configurations, you have specified a device on which the library disk is to be mounted, but the device is not a disk.

User Action: Specify a disk device.

BADSPEC, File specification *specification* **cannot be parsed using default specification** *specification*

Explanation: VMSINSTAL cannot locate file because of invalid specification.
User Action: If you can determine the problem and resolve it, then simply install the product again. Otherwise, contact your DIGITAL field service representative.

CREATDIR, Creating temporary directory DIR-NAME.
Explanation: This is an informational message. VMSINSTAL creates a temporary directory if the G option is specified.

User Action: None.

CTRLY, Installation cancelled via CTRL/Y.
Explanation: This is an informational message that appears when you cancel the installation by entering a CTRL/Y.

User Action: None.

DECNET, Your DECnet network is up and running.
Explanation: VMSINSTAL checks to ensure that your DECnet

User Action: None.

DECNET, Your DECnet network is up and running.
Explanation: VMSINSTAL checks to ensure that your DECnet network has been shut down when an installation is begun. This check has failed.

User Action: You will be asked if you want to continue the installation. It is recommended that you remedy this situation before continuing.

DEVMNT, Device DEVICE-NAME will be MNT-STATUS

Explanation: This is an informational message telling you the device specified for the output of the G option will be initialized (if it is a disk) and mounted. This message is issued only if the specified output device is not mounted when VMSINSTAL is invoked.

User Action: None.

DEVNOTEXIST, Device DEVICE-NAME does not exist.

Explanation: The device specified for the output of the G option does not exist.

User Action: Specify a valid device name.

ILLSAVESET, Illegal save set name specified to RESTORE_SAVESET.

Explanation: The product installation procedure contains an incorrect call to RESTORE_SAVESET.

User Action: Contact your DIGITAL field service representative.

INSFAIL, The installation of PRODUCT-NAME has failed.

Explanation: The product installation did not exit with a successful status.

User Action: Examine the console output to determine the cause of the error. If you are unable to resolve the problem, contact your DIGITAL field service representative.

INSDFAIL, The installation of a product has failed.

Explanation: An error occurred while attempting to do deferred callbacks during the installation.

User Action: Examine the conole output to determine the cause of the error. If you are unable to resolve the problem, contact your DIGITAL field service representative.

INTEGER, Please enter an integer value.

Explanation: VMSINSTAL asked a question that must be answered by specifying an integer.

User Action: Answer the question again, specifying an integer.

INVOPTIONS, To specify options, parameter 3 must be the word OPTIONS.

Explanation: If you are specifying VMSINSTAL options, the third parameter to VMSINSTAL must be the word OPTIONS.

User Action: Invoke VMSINSTAL again, specifying the options in the correct format.

IVPFAIL, The IVP for *product* has failed.

Explanation: VMSINSTAL has invoked the installation verification procedure (IVP) for the specified product, and the IVP has failed.

User Action: If you can determine the problem and resolve it, then simply install the product again. Otherwise, contact your DIGITAL field service representative.

LIBDIFF, The following system and library disk files do not match.

LIBDIFF2, You must resolve these differences before you continue.

Explanation: The following system and library disk files do not match. You must resolve these differences before continuing. For tailored systems, VMSINSTAL temporarily removes all nonessential system files from the system disk. Before doing this, however, it checks to ensure that the copies of these files on the system disk match those on the library disk. This check has failed, and the offending files are listed.

User Action: You must resolve the differences before reinvoking VMSINSTAL.

LOWQUOTA, One or more account quotas may be too low.

Explanation: VMSINSTAL checks to ensure that your process quotas are valid when an installation is begun. This check has failed.

User Action: You will be asked if you want to continue the installation. It is recommended that you remedy this situation before continuing. See Section 5.1.

MOVEFILES, Files will now be moved to their target directories.

Explanation: This is an informational message. If sufficient disk space is available at the beginning of an installation, VMSINSTAL builds new files in a working directory for the optional product or update. After all the files are built, they are moved to their target directories.

User Action: None.

NOCONSOLE, Unknown CPU type number. Console volume cannot be remounted.

Explanation: While attempting to remount the console volume, VMSINSTAL could not determine what type of CPU you have.

User Action: Contact your DIGITAL field service representative.

NOFILE, [new] file *file* **does not exist.**

Explanation: VMSINSTAL is attempting to locate an existing (or perhaps new) file with the given name, but the file cannot be found.

User Action: If you can determine the problem and resolve it, then simply install the product again. Otherwise, contact your DIGITAL field service representative.

NOLIBDISK, Library disk could not be mounted.

Explanation: You were asked to mount the library disk so that an installation could be performed on a tailored system. For some reason, the library disk could not be mounted.

User Action: Correct the situation that prevented mounting the library disk and begin the installation again.

NOPRINT, File *file* **cannot be printed.**

Explanation: The product installation required the printing of one or more files, but the files could not be queued for printing.

User Action: This error is most likely caused by an undefined SYS$PRINT queue at your site. If you do not care that the files could not be printed, then you can ignore this message. Otherwise, define a SYS$PRINT queue and perform the installation again.

NOPROC, The kit installation procedure is missing.

Explanation: The main installation procedure for the product could not be located on the distribution volumes.

User Action: Contact your DIGITAL field service representative.

NOPRODS, None of the specified products were found.

Explanation: You have specified one or more products to be installed from the distribution volumes, but none of the products exist on the volumes.

User Action: Specify products that exist on the volumes.

NOSAVESET, Required save set *letter* **does not exist.**

Explanation: The product save set with the specified letter, which is required for the installation, does not exist on the distribution volumes.

User Action: Contact your DIGITAL field service representative.

NOSETPRV, You are not running on an account with SETPRV privilege.

Explanation: VMSINSTAL checks to ensure that your account has SETPRV privilege when an installation is begun. This check has failed.

User Action: You will be asked if you want to continue the installation. It is recommended that you remedy this situation before continuing.

NOSPACE, There are not enough blocks to restore the system disk.

Explanation: After installation on a dual-RL02 system, VMSINSTAL attempts to restore your system disk to its original tailoring configuration. However, there are not enough free blocks left on the system disk.

User Action: VMSINSTAL tells you how many additional free blocks are required. You must make more space on the system disk by purging and/or deleting files. Once this is accomplished, you can restore the original tailoring configuration, as described in Section 5.4.

NOTSYSTEM, You are not logged in to the SYSTEM account.

Explanation: VMSINSTAL checks to ensure that you are logged in to the SYSTEM account when an installation is begun. This check has failed.

User Action: You will be asked if you want to continue the installation. It is recommended that you remedy this situation before continuing.

NULSAVESET, Null save set name specified to RESTORE_SAVESET.

Explanation: A product installation procedure contained an incorrect call to RESTORE_SAVESET.

User Action: Contact your DIGITAL field service representative.

PEAKUTIL, This product requires *number* **blocks during installation.**

Explanation: The installation procedure for the product has determined that there are not enough free blocks on the system disk to install the product.

User Action: Make more space on the system disk by purging and/or deleting files, and then install the product again.

PRODSKIP, Products that have not been installed will be skipped.

Explanation: You have requested that multiple products be installed, but one of these products requires that the system be rebooted after its installation. To accomplish this, the remaining uninstalled products must be skipped.

User Action: Reinstall the remaining products after the system is rebooted.

REBOOT, This product requires that the system be rebooted.

Explanation: The product that was just installed requires that the system be rebooted to complete the installation.

User Action: Reboot the system after shutdown completes.

RECORDANS, Auto-answer file will be recorded.

Explanation: The auto-answer option was specified when VMSINSTAL was invoked. No answer file exists for the product about to be installed, so one will be recorded.

User Action: Answer the prompts and questions as usual. Your answers will be recorded.

RECOVER, *product* was being installed when the system failed.

Explanation: The specified product was being installed when the system failed due to a power failure or other problem.

User Action: You will receive additional instructions on how to proceed.

RESTORE, Restoring product save set *letter*...

Explanation: The specified product save set is about to be restored from the distribution volumes by the Backup Utility.

User Action: None.

RETAILOR, Your original tailoring environment will now be restored.

Explanation: The tailoring files that were removed from your system disk for installation, will be copied back to the system disk.

User Action: None.

REWIND, The tape will be rewound to try again.

Explanation: VMSINSTAL is attempting to restore a product save set with the Backup Utility. The required save set was not located on the currently mounted distribution tape, but VMSINSTAL will rewind the tape in case the save sets were not recorded in order.

User Action: This message will be preceded by various serious BACKUP messages, which can be ignored.

SMALLDISK, This is a small disk system.

Explanation: VMSINSTAL has determined that you are running a tailored system. It will temporarily remove all nonessential files from the system disk in order to make room for the installation.

User Action: You will receive additional instructions as to how to proceed.

SYSDIR, This product creates system directory *directory*.

Explanation: The product being installed creates the specified system directory.

User Action: None.

SYSDISK, This product creates system disk directory *directory*.
Explanation: The product being installed creates a directory on the system disk.

User Action: None.

TAMPER, File *file* **has been tampered with.**
Explanation: VMSINSTAL is attempting to update a system text file, but the file has been modified locally. The update cannot be performed.

User Action: If you modify system text files, then they cannot be updated by DIGITAL software installations.

UNEXPECTED, Installation terminated due to unexpected event.
Explanation: Some unexpected event has caused the immediate termination of VMSINSTAL. This message should be preceded by other messages that explain the problem in greater detail.

User Action: If you can determine the problem and resolve it, then simply install the product again. Otherwise, contact your DIGITAL field service representative.

UPDATED, File *file* **is already updated.**
Explanation: VMSINSTAL is attempting to update a system text file, but the file has already been updated successfully.

User Action: None.

USEANS, Auto-answer file will be used.
Explanation: The auto-answer option was specified when VMSINSTAL was invoked. An answer file exists for the

product about to be installed, so the recorded answers will
be used.

User Action: You will not have to answer prompts and
questions displayed by the product. If you want to record
new answers, then you must delete the answer file and install
the product with the auto-answer option.

VMIORIG, VMIORIG.TLR still exists; please see your documentation.

Explanation: When performing an installation on a
tailored system, VMSINSTAL records the current tailoring
configuration and then removes it from the system disk
temporarily. The configuration is recorded in the file
SYS$UPDATE:VMIORIG.TLR. After installation, VMSINSTAL
restores the configuration and deletes VMIORIG.TLR. If,
however, the system crashes during the installation, then you
must restore the configuration and delete the file, manually.
See Section 5.4.

User Action: Because VMIORIG.TLR still exists, VMSINSTAL
assumes that you have not restored your tailoring configuration.
You must do so, and then delete VMIORIG.TLR.

YESNO, Please enter YES or NO.

Explanation: VMSINSTAL asked a question that must be
answered with YES or NO.

User Action: Answer the question again, specifying YES or
NO. You may abbreviate these as Y or N.

6 Upgrading To Version 4.0

This chapter tells you how to upgrade your VAX/VMS
operating system from Version 3.4 or later to Version 4.0.
If your operating system is not currently at one of these levels,
you will have to update to Version 3.4 before you can upgrade
to Version 4.0.

When you complete the upgrade to Version 4.0, you must
perform a a mandatory update using the appropriate mandatory
update kit:

- If you have a VAX–11/780, VAX–11/782, or VAX–11/785,
  use the kit labeled *VAX/VMS V4.0 MAN/UPD RX1 1/1
  AS-Z235A-BE.*

- If you have a VAX–11/750, VAX–11/730, or VAX–11/725,
  use the kit labeled *VAX/VMS V4.0 MAN/UPD T58 1/1
  BE-Z236A-BE.*

To do the update *after* you complete the upgrade, log in to the
system manager's account; then enter the following command:

```
$ @SYS$UPDATE:VMSINSTAL VMSMUP040 CSA1:
```

Note: **Because this update includes a patch to SYS.EXE, purge the
files only *after* the reboot.**

Then follow the instructions issued by the installation command
procedure.

A major part of the upgrade is automated and does not require
that an operator be present throughout. For this reason, you
should do the upgrade from a hard-copy device to have a
record of events that occur during the upgrade.

Section 6.1 lists the steps you should take before you start
the upgrade. The actual upgrade is described in Section 6.2.
Sections 6.3 through 6.5 deal with post upgrade information.

Here is a list of materials you need to do the upgrade:

- The Version 4.0 software distribution kit

6–1

- A blank disk for building the upgraded system

- A blank console volume (floppy diskette if you have a VAX–11/780, VAX–11/782, or VAX–11/785; TU58 if you have any other VAX processor.)

- A blank library disk if you are upgrading to a tailored environment

Note: **Upgrades are not supported for VAX–11/730 users with dual-RL02 system configurations. See the installation booklet** *Installing VAX/VMS on a Dual-RL02 System* **for instructions.**

You should read this entire installation procedure before beginning the upgrade to be sure that you are aware of all upgrade requirements.

Note: **If you have changed the names of system directories in your current operating system, or deleted VAX/VMS files from them, the upgrade procedure may not work correctly. You will have to restore your operating system to a standard system before you can continue with the upgrade.**

You should also note the following before beginning the upgrade:

- As part of the upgrade, the paging, swapping, dump, and authorization files are purged back to one version.

- Everything in the [SYSERR] directory is deleted.

- All operator and accounting logs are deleted. If you want to keep any of the files, move them to a user directory before starting the upgrade.

It is recommended that you back up your system disk before upgrading because Version 4.0 makes some files obsolete and these are subsequently deleted.

Because of the HSC50 and compatibility mode utilities restriction, the upgrade procedure will fail if your system disk is connected to the HSC controller and if the numeric value of your system disk's physical device name is greater than 255, as calculated by the following equation:

```
(16 * (the numeric difference between your controller letter
       and the letter A in the ASCII collating sequence))
+ (your unit number) > 255
```

For example, a physical system device name of DUZ6 will cause the upgrade to fail because the numeric value is too high.

```
(16 * (Z - A)) + 6 = (16 * 25) + 6 = 406 > 255
```

You can work around this problem by taking the following steps:

1 Decrease the unit number of your system disk by changing its unit plug.

2 Decrease the node number of your HSC. Node numbers between 0 and 15 will result in controller letters between K and Z, respectively.

For example, changing the HSC node number from 15 to 0 in the previous example results in a physical system device name of DUK6. Therefore, the upgrade succeeds because the device name value is less than 255.

```
(16 * ('K' - 'A')) + 6 = (16 * 10) + 6 = 166
```

and

```
166 < 255)
```

If the upgrade is interrupted for any reason, it can resume from the point at which the system was most recently booted. There are three times during the upgrade procedure when the system reboots.

If you want the upgrade to resume automatically, you must boot the system exactly as requested by the upgrade procedure after the planned system shutdown in Phase 1.

To allow the upgrade procedure to resume at any point, the procedure places the new Version 4.0 system files in an alternate disk directory root (SYSF) from your current system's files, which are in SYS0. This ensures that you have at least one set of bootable system files at all times.

6.1 Preparing to Upgrade the System

Use the following procedure to prepare for the upgrade. Note that the system disk and the distribution volume cannot be moved from one device to another during the upgrade.

1 Log in to the system manager's account, SYSTEM, on the console terminal.

2 Use stand-alone BACKUP to back up your current system disk to a scratch disk. (A step-by-step procedure for backing up your system disk is given in Chapter 4).

3 If your current system disk can be removed, store it in a safe place so that it is readily available if needed. Use the backup copy to upgrade your system.

4 Boot the backup copy of your system disk (or your current system disk, if it cannot be removed).

If you are upgrading a VAX system other than a VAX–11/750, put the system in console mode, halt the processor and type the following command:

```
>>> B ddu
```

If you are upgrading a VAX–11/750, use one of the following boot methods:

- If you have a switch position on the BOOT DEVICE switch allocated to the system device, set the switch to the appropriate position and type the letter *B* at the console terminal. If you are not sure of the correct setting, contact your DIGITAL field service representative.

- If you do not have a position on the BOOT DEVICE switch for the system device and want to boot the disk directly, press CTRL/P to go into console mode, and then type following command:

```
>>> B ddcu
```

The *ddcu* code refers to the device name of your system device. See Chapter 4 for help with device names.

- If you want to boot using the TU58, insert the console TU58 in the console drive and set the BOOT DEVICE switch to position A. Then press CTRL/P to go into console mode and type this command:

  ```
  >>> B/800 DDA0
  ```

  Then when you get the BOOT58 prompt, type this command, where *ddcu* represents the physical device name of the system device.

  ```
  BOOT58> B ddcu
  ```

  If you decide to boot from the TU58 cartridge, be sure to reply YES when the upgrade procedure asks if you want to boot from the TU58.

5 When the backup copy of your system disk boots, log in under the system manager's account, SYSTEM. (You may want to consider having a second terminal logged in for doing support tasks during the upgrade.)

6 To upgrade your system, you must have a minimum of 9000 free blocks on the system disk you intend to upgrade. If you do not have this space, delete as many user files as necessary to get it. You can confirm the free block count with the following command:

```
$ SHOW DEVICES/FULL ddcu:
```

Here ddcu represents the physical device name of the drive that contains the backup copy of your system disk.

7 Make sure the restart switch on your processor control panel is set to automatic restart. The upgrade will continue automatically after shutdown(s) as long as this switch is positioned as shown in the following table.

VAX System	Restart Switch	Required Setting
VAX–11/725	AUTO/RESTART BOOT	ON
VAX–11/730	AUTO/RESTART BOOT	ON
VAX–11/750	POWER ON ACTION	(RESTART)
VAX–11/780	AUTO/RESTART	ON

8 Prevent users from logging into the system.

```
$ SET LOGINS/INTERACTIVE=0
```

9 If you are running DECnet–VAX, shut down the network. First enter this command.

```
$ RUN SYS$SYSTEM:NCP
```

When the NCP> prompt is displayed, type

```
NCP>SET EXECUTOR STATE OFF
```

10 Press CTRL/Z to return to DCL command level.

11 You should stop all queues to avoid locked-file errors during the upgrade. Note that queues are initialized during the upgrade, and any jobs pending will be lost. Check to see if there are any jobs in the queues with the following command:

```
$ SHOW QUEUE/DEVICE/BATCH/FULL/ALL
```

If no queues are set up, an appropriate message will be displayed and you may bypass this step.

Use a DCL command of the following form to stop each queue:

```
$ STOP/QUEUE/NEXT queue_name
```

6.2 Performing the Upgrade

At this point, you begin the actual upgrade procedure.

1 Place the VAX/VMS Version 4.0 distribution volume on the appropriate drive. If you are using a tape drive, put the drive on line; if you are using a disk drive, spin the drive up.

2 Execute the following command:

```
$ SET DEFAULT SYS$UPDATE:
```

3 Invoke the VMSINSTAL command procedure.

```
$ @VMSINSTAL
```

VMSINSTAL responds by announcing itself and asking if you have a satisfactory backup copy of your current system:

```
        VMS Software Product Installation Procedure
It is <date> at <time>
Enter a question mark (?) at any time for help.
*Are you satisfied with the backup of your system disk [YES]?
```

4 If you have already backed up your old system disk, press the RETURN key to continue with the upgrade.

If you have not backed up your old system disk, you may do so now by entering **NO**. VMSINSTAL will return you to DCL level so that you can do the backup. When the backup is completed, invoke VMSINSTAL to restart the upgrade.

5 Next, VMSINSTAL requests the name of the drive holding the distribution volume. Enter the device name in the *ddcu:* format.

For example, if the distribution volume is a magnetic tape mounted in a TS11 that is the first device on the UNIBUS, enter *MSA0:*. See Chapter 4 if you need more information on device names.

The upgrade procedure may abort if the device name format appears incorrect. If you specify a nonexistent device or a device that is not connected, an "invalid device" error message will be displayed. If this happens, use the DCL command SHOW DEVICES to verify device status.

6 When VMSINSTAL prompts you for the name of the product you wish to install, respond by entering VMS.

7 When your distribution volume is ready for mounting, enter **Y**.

VMSINSTAL responds with the following messages:

```
%MOUNT-I-MOUNTED, VMS040        mounted on _ddcu:
The following products will be installed:
   VMS     V4.0
        Beginning installation of VMS 4.0 at <time>
%VMSINSTAL-I-RESTORE, Restoring product save set A...
                        VAX/VMS V4.0 Upgrade Procedure
                .
                .
                .
                .
```

If you are using a tape distribution volume, there will be a pause of several minutes between the first and second lines.

8 Next, the procedure describes the upgrade, discusses the automatic restart capability.

9 This is followed by a set of messages that describe cautions and requirements related to doing the upgrade:

- First the procedure states that the kit *must* be stored in directory [000000]. The kit is stored in [000000] by default.

- Then you are cautioned about preserving files that may be deleted by the upgrade.

- Next, the procedure notes the importance of RESTAR.CMD matching your memory interleaving.

- The procedure describes how AUTOGEN may attempt to re-size your pagefile and swapfile if you do not manually override it.

- If you have DECnet–VAX installed, you must have the Version 4.0 DECnet–VAX license which is packaged in a separate distribution kit. No Version 3.0 DECnet–VAX license will work with a Version 4.0 system.

- If you are upgrading a VAX–11/785, you are told that the default bootstrap command procedure must be set to boot the system device *before* you start the upgrade.

If you indicate that you want to interrupt the upgrade to comply with any of these conditions, the procedure returns you to DCL command level and the upgrade terminates. If you do this, you must reinvoke VMSINSTAL when ready to resume the upgrade.

10 When the upgrade resumes, the procedure the next step is determined by your system configuration.

- If your distribution medium is located on an HSC disk, the procedure asks for the name of the HSC (for example, HSC001). See the *HSC50 User Guide* for details on HSC names.

- If your system includes either a CI780 or CI750, the upgrade gives you the option of creating a common disk for a VAXcluster.

- If your system does not include a CI780 or CI750 and the system disk has fewer than 70,000 blocks, the upgrade gives you the option of building a tailored system.

Note: **If you intend to tailor a system that includes only two disk drives,—that is, a dual-RK07 configuration or a VAX–11/725—you will not be able to install any optional product distributed on large-disk media. For example, if your disk storage is limited to just two RK07 disk drives, you will not be able to install any optional product distributed on an RK07 distribution kit.**

If you decide to tailor your system, the upgrade procedure will prompt you for the name of the device you will use to build the LIBRARY disk.

- If your system has installed DECnet–VAX (as determined by the presence of the file SYS$SYSTEM:NETNODE.DAT), the upgrade procedure lets you select a DECnet–VAX area for your system. For more information on area routing, see the *Guide to Networking on VAX/VMS.*

6.2.1 Upgrade Phase 1

The upgrade is presented in five phases. Two descriptions are provided for Phase 1: one for users upgrading systems other than VAX–11/750 and one for users upgrading VAX–11/750 systems. The final four phases are common to all VAX systems.

You should now proceed to Phase 1 of the upgrade. If you are upgrading a VAX–11/750, skip the next section and go to Section 6.2.1.2.

6.2.1.1 **Upgrade Phase 1 for VAX–11/725, VAX–11/730, and VAX–11/780 Systems**

This section describes the first phase of the upgrade for users who are upgrading VAX–11/725, VAX–11/730, and VAX–11/780 systems.

1 To ensure system security, the upgrade procedure requires you to change the passwords for the SYSTEM, SYSTEST, and FIELD accounts before continuing. The first time you set these passwords, the system requires a password of six or more characters.

2 The upgrade procedure now turns off the disk quotas on the system disk and removes directory entries that point to nonexistent files. If you are not running disk quotas, an appropriate error message is displayed. You may ignore this message.

3 Your upgraded system will be built in system root SYSF.

At this point the upgrade procedure will prompt you through the building of a new console volume that will let you boot the new system from SYSF. Specifically, the procedure will change the default bootstrap command procedure (DEFBOO.CMD) on the new volume to boot from SYSF.

4 The procedure prompts you to insert your current console volume in the console drive. Your current console volume (the procedure calls it your "old" console volume) is used as a base to build the new console volume but it is *not* altered by the procedure.

5 When you indicate that you are ready to continue, the procedure copies your old console volume to a disk directory.

6 Now the procedure prompts you to set DEFBOO.CMD to boot the backup copy of your system disk (assuming you have not done so previously). Using the form dduBOO.CMD, enter the name of the boot file you want to copy to DEFBOO.CMD. For example, if the system disk is in DBA1, you enter DB1BOO.CMD. If the system disk is controlled by either an HSC or a UDA, the revised version of CIBOO.CMD or DUABOO.CMD that you built when you first installed VAX/VMS on the HSC- or the UDA-controlled system disk must be used. The selected command procedure

must be able to boot the system disk from the drive where you have placed it without any operator intervention (for example, registers R0 through R5 are correctly initialized by the command procedure).

Do not specify a conversational boot command file. The upgrade kit is set with parameters that will boot any system. The procedure does build a conversational boot file (UPGGEN.CMD) that boots from SYSF, but you should only use this file for situations prescribed by the procedure.

7 When this is done, put your old console volume away for safe keeping and insert a blank volume in the console drive. Do not remove this volume for the rest of the upgrade.

Note: **If the new console volume is a TU58, be sure the RECORD switch is set to permit writing, that is, the switch should be slid in the direction of the arrow inscribed on it.**

8 Now the procedure gives you the opportunity to have the Bad Block Locator Utility (BAD) check the new console volume before it is initialized. BAD checks the new console volume for any defective blocks, and DIGITAL recommends that you run it. If you choose to run BAD, allow an additional 30 minutes if you are using a TU58 volume. If you are using a floppy volume, an additional 15 minutes is required.

Details on running BAD can be found under the ANALYZE /MEDIA command in the *VAX/VMS DCL Dictionary* or under the description of BAD in the *VAX/VMS Utilities Reference Volume*.

9 Finally, the procedure initializes the new console volume but it will not copy the files to it from the disk until just before the first reboot.

Note: **Unless you are upgrading to a tailored system or using RL02 distribution media, you should not have to intervene in the upgrade from this point on.**

10 The upgrade procedure now does a directory cleanup and removes installed images. If you are building a tailored system, the procedure tells you that you will be needed to prepare the LIBRARY disk in Phase 2 of the upgrade.

11 Next, the procedure builds the directory tree in SYSF and and deletes all operating system files not needed to complete the upgrade.

12 Now the upgrade procedure restores the Version 4.0 *required* save set to the backup copy of your system disk and verifies that the save set files have been restored correctly.

13 When the verification pass is complete, the upgrade procedure purges the paging, swapping, dump, and authorization files and puts the most current version in the new directory tree.

14 After using AUTOGEN to save your old SYSGEN parameters and preparing the startup command file for Phase 2, the upgrade procedure copies the console files to the new volume and builds its boot block in preparation for the first reboot.

15 Now the procedure tells you it will shut down to reboot the partially installed Version 4.0 system. It also tells you that the upgrade continues automatically after the reboot but that if if you must restart the upgrade manually, simply enter the letter B.

16 Now the procedure cautions you not to move the distribution volume nor the system volume and not to change the SYSGEN parameters for the remainder of the upgrade.

17 Next, the procedure tells you how to handle a shutdown failure or a reboot failure.

- On a VAX–11/730 system, the microcode may be reloaded. If the system fails to boot correctly, power the system off and then on again so that the console will reload the microcode from the newly created TU58.

- If the system fails to boot in a CI780 environment because of insufficient nonpaged dynamic memory, use the conversational boot command procedure UPGGEN.CMD to increase the NPAGEDYN system parameter.

Note: If you need to increase this parameter, submit a Software Performance Report that describes your

complete hardware configuration and what you did to get the system to boot.

To increase this parameter, use the following sequence of console commands:

```
>>> @UPGGEN.CMD
```

This will put you in SYSBOOT where you should enter the following command:

```
SYSBOOT> SET NPAGEDYN 120000
```

Then use this command to return to the upgrade procedure:

```
SYSBOOT> CONTINUE
```

When the system reboots, it identifies itself and announces the beginning of Phase 2.

```
VAX/VMS Version 4.0 <date hh:mm>
```

Skip to Section 6.2.2, which describes Phase 2.

6.2.1.2   **Upgrade Phase 1 for VAX–11/750**
This section describes the first phase of the upgrade for users who are upgrading VAX–11/750 systems.

1 To ensure system security, the upgrade procedure requires you to change the passwords for the SYSTEM, SYSTEST, and FIELD accounts before continuing. The first time you set these passwords, the system requires a password of six or more characters.

2 The upgrade procedure now turns off the disk quotas on the system disk and removes directory entries that point to nonexistent files. If you are not running disk quotas, an appropriate error message is displayed. You may ignore this message.

You are now asked if you want to boot using the TU58 rather than directly from the disk. If you have a CI750, you *must* answer YES to this question. DIGITAL recommends that you answer YES in any case to ensure you have a console TU58 with the latest versions of the VMB and BOOT58 programs.

If you choose to boot directly from the system disk, skip the next step and go to step 4.

3 Your upgraded system will be built in system root SYSF
so that your current system in SYS0 is available if needed.
At this point, the procedure will guide you through the
building of a new console TU58 if you previously indicated
that you would be booting from the TU58. Specifically,
the procedure will change the default bootstrap command
procedure (DEFBOO.CMD) on the new volume to boot from
SYSF.

a Position the BOOT DEVICE switch on your processor
control panel to position A and leave it in position A for
the rest of the upgrade.

b The procedure prompts you to insert your current
console TU58 in the console drive. Your current console
volume (the procedure calls it your "old" console volume)
is used as a base to build the new console volume but it
is *not* altered by the procedure.

c When you indicate you are ready to continue, the
procedure copies your old console volume to a disk
directory in Files–11 format.

d Now the procedure prompts you to set DEFBOO.CMD
to boot the backup copy of your system disk (assuming
you have not done so previously). Using the form
dduBOO.CMD, enter the name of the boot file you
want to copy to DEFBOO.CMD. For example, if the
system disk is in DBA1, you enter DB1BOO.CMD. If the
system disk is controlled by either an HSC or a UDA,
the revised version of CIBOO.CMD or DUABOO.CMD
that you built when you first installed VAX/VMS on the
HSC- or the UDA-controlled system disk must be used.
The selected command procedure must be able to boot
the system disk from the drive where you have placed it
without any operator intervention (for example, registers
R0 through R5 are correctly initialized by the command
procedure).

Do not specify a conversational boot command file. The
upgrade kit is set with parameters that will boot any
system. The procedure does build a conversational boot
file (UPGGEN.CMD) that boots from SYSF, but you
should only use this file for situations prescribed by the
procedure.

e When this is done, put your old console volume away for safe keeping and insert a blank volume in the console drive. Do not remove this volume for the rest of the upgrade.

Note: Be sure the RECORD switch on the new console TU58 is set to permit writing, that is, the switch should be slid in the direction of the arrow inscribed on it.

f Now the procedure gives you the opportunity to have the Bad Block Locator Utility (BAD) check the new console volume before it is initialized. BAD checks the new console volume for any defective blocks, and DIGITAL recommends that you run it. If you choose to run BAD, allow an additional 30 minutes.

Details on running BAD can be found under the ANALYZE/MEDIA command in the *VAX/VMS DCL Dictionary* or under the BAD utility in the *VAX/VMS Utilities Reference Volume*.

g Finally, the procedure initializes the new console volume but it will not copy the files to it from the disk until just before the first reboot.

Note: Unless you are upgrading to a tailored system or using RL02 distribution media, you should not have to intervene in the upgrade from this point on.

4 The upgrade procedure now cleans up its directories and removes installed images. If you are building a tailored system, the procedure tells you that you will be needed to prepare the LIBRARY disk in Phase 2 of the upgrade.

5 Next, the procedure builds the directory tree in SYSF and and deletes all operating system files not needed to complete the upgrade.

6 Now the upgrade procedure restores the Version 4.0 *required* save set to the backup copy of your system disk and verifies that the save set files have been restored correctly.

7 When the verification pass is complete, the upgrade procedure purges the paging, swapping, dump, and authorization files and puts the most current version in the new directory tree.

8 After using AUTOGEN to save your old SYSGEN parameters and preparing the startup command file for Phase 2, the upgrade procedure copies the console files to the new volume and builds a boot block on it in preparation for the first reboot.

9 Now the procedure tells you it will shut down to reboot the the partially installed Version 4.0 system. It also tells you that the upgrade continues automatically after the reboot and how to boot the new system if you must restart the upgrade manually,

- If you are booting from the console TU58, you will reboot the system from the BOOT58 command level. To invoke BOOT58, press CTRL/P to put the system in console mode; then enter the following command:

  ```
  >>> B/800 DDA0
  ```

  When you get the BOOT58 prompt, make the following entry:

  ```
  BOOT58> B
  ```

- If you are booting from the disk, you will have to use CTRL/P to get into console mode and then reboot with this command:

  ```
  >>> B/F0000000 ddcu
  ```

10 Next the procedure tells you how to handle a reboot failure.

If the system fails to boot in a CI750 environment because of insufficient nonpaged dynamic memory, use the conversational boot command procedure UPGGEN.CMD to increase the NPAGEDYN system parameter.

Note: **If you need to increase this parameter, please submit a Software Performance Report that describes your complete hardware configuration and what you did to get the system to boot.**

11 To invoke UPGGEN.CMD when booting from a disk, use CTRL/P to get into console mode, then enter the following command:

```
>>> B/F0000001 ddcu
```

If you are booting from a TU58, invoke BOOT58 from console mode with this command:

```
>>> B/800 DDA0
```

When you get BOOT58 prompt, enter this command:

```
BOOT58> @UPGGEN.CMD
```

12 At the SYSBOOT prompt, enter the following command:

```
SYSBOOT> SET NPAGEDYN 120000
```

13 Then use this command to return to the upgrade procedure:

```
SYSBOOT> CONTINUE
```

When the system reboots, it identifies itself and announces the beginning of Phase 2.

```
VAX/VMS Version 4.0 <date hh:mm>
```

This completes Phase 1 of the upgrade. The rest of the upgrade does not require that an operator be present; however, if you are booting from the TU58, you will have to manually reboot each time the system shuts down (once in Phase 4 and once in Phase 5). If you are booting from the disk, reboots are automatic and do not require your intervention.

6.2.2   Upgrade Phase 2

This phase begins with the upgrade procedure deleting the remaining Version 3.0 files.

Then the rest of the Version 4.0 files in the LIBRARY and OPTIONAL save sets are restored from the distribution kit. This step varies, depending on your configuration.

- In a standard VAX/VMS configuration with either RK07, RA60, or magnetic tape distribution media, the LIBRARY save set is restored, and then the OPTIONAL save set is restored.

- In a standard VAX/VMS configuration with RL02 distribution media, the OPTIONAL save set is restored first. Then the procedure prompts you to remove the first volume of the distribution kit and to insert the second volume. When you comply, the LIBRARY save set is restored.

- In a tailored VAX/VMS configuration (other than one using RL02 distribution media), there are two possible scenarios:

  - If you can mount the distribution volume and the scratch library disk simultaneously, the procedure restores the LIBRARY save set to the scratch library disk. Then it dismounts the library disk, remounts it as a FILES-11 disk, and restores the OPTIONAL save set.

  - If you cannot mount the distribution volume and the scratch library disk at the same time, the procedure copies the LIBRARY and OPTIONAL save sets to the system disk, dismounts the distribution volume, and asks you to insert a scratch disk in its place. After you comply, the procedure first restores the LIBRARY save set to the scratch disk, and then the OPTIONAL save set.

When the save sets are restored, the upgrade converts your user authorization file (SYSUAF) and your network authorization file (NETUAF) to the new format.

6.2.3   Upgrade Phase 3

Phase 3 of the upgrade merges the VAX/VMS-distributed files that are commonly edited by system managers with new VAX/VMS files. Ignore any "file not found" error messages that appear at this time.

Next, the upgrade procedure merges all the miscellaneous user files that exist in the old system directories into a new set of system directories, temporarily called [SYSF.SYSEXE], [SYSF.SYSMGR], [SYSF.SYSLIB], and so on. The amount of time this takes depends on the number of user files.

6.2.4   Upgrade Phase 4

During Phase 4 of the upgrade, the new site-specific console volume is modified to allow reboot of the complete VAX/VMS Version 4.0 system. Be sure the new site-specific console volume created in Phase 1 is in the console drive (CS1: for VAX–11/780 and VAX–11/750 systems; CS2: for VAX–11/730 and VAX–11/725 systems).

When the modification is completed, you receive the following message:

`System shutting down to boot the complete V4 system.`

Leave the newly created site-specific console volume in the console drive. The system disk must also remain where it is in order to proceed to the next phase of the upgrade.

The system will attempt an automatic reboot after the shutdown.

Note: **If you are upgrading a VAX–11/750 and are booting from the console TU58, you will have to manually reboot by entering the letter B when you get the BOOT58> prompt.**

If the system fails to boot at this point because of insufficient nonpaged dynamic memory, you must use a standard conversational bootstrap to increase the NPAGEDYN (nonpaged dynamic memory) system parameter.

If you need to increase this parameter, please submit a Sofware Performance Report that describes your complete hardware configuration and what you did to get the system to boot.

If you need to invoke a conversational bootstrap at this point in the upgrade, invoke the conversational boot command procedure for your system disk. For example, if the system disk is on the first MASSBUS device, invoke DB0GEN, if the first UDA device, DU0GEN, and so forth.

When the boot is completed, the following message appears:

`VAX/VMS Version 4.0 <date hh:mm>`

6.2.5　Upgrade Phase 5

Phase 5 of the upgrade procedure creates a new site-specific SYSGEN parameter file, AUTOGEN.PAR, that combines the default values for Version 4.0 and the site-specific values you were using for your Version 3.0 system.

The command procedure cleans up several directories, announces the upgrade to Version 4.0 is complete, and makes several suggestions:

After the upgrade finishes, there are several files that you must manually edit
due to changes in the system, and other files that you may delete. See the
Guide to VAX/VMS System Management and Daily Operations for complete
information.

Among the things you should do:

- DECOMPRESS THE SYSTEM LIBRARIES - For space considerations, many of the system
libraries are shipped in a data compressed format. If you have enough disk
space, you may decompress them for faster access. Use SYS$UPDATE:LIBDECOMP.COM
to data expand the libraries. If you choose not to decompress these libraries
there will be a negative impact on the performance of the HELP and LINK
commands.

- Edit SYS$MANAGER:SYSTARTUP.COM - Your SYSTARTUP.COM will not be executed when
the system reboots. It is still present in SYS$MANAGER, as the next lowest
version.

- Edit SYS$MANAGER:SYSHUTDWN.COM - Your SYSHUTDWN.COM will not be executed when
the system shuts down. It is still present in SYS$MANAGER, as the next lowest
version.

Then the system shuts down and automatically reboots.

Note: **If you are upgrading a VAX-11/750 and are booting from the console TU58, you will have to manually reboot by entering the letter B when you get the BOOT58> prompt.**

If your system is part of a VAXcluster, you will have to execute
AUTOGEN with this command after the upgrade finishes.

```
$ @SYS$UPDATE:AUTOGEN SAVPARAMS REBOOT
```

If your system includes a CI780 or CI750, VAX/VMS waits
for several minutes to form or join a cluster. If the upgrade
procedure does not continue within 5 minutes, please submit a
Software Performance Report.

If you have upgraded a tailored system and want to decompress
the system libraries, physically mount the LIBRARY disk and
execute this command sequence. In place of ddcu:, enter the
name of the drive holding the LIBRARY disk.

```
$ @SYS$UPDATE VMSTAILOR MOUNT ddcu:
$ @SYS$UPDATE:LIBDECOMP
```

The decompressed libraries require approximately 4000
additional blocks of disk space and the decompression process
may take up to 2 hours depending on the type of processor you
are using. A general guideline for decompressing individual
library files is that HELP libraries increase by 50% and OBJECT
libraries by 25% when decompressed.

6.3 Special File Handling During Upgrade

After you have completed the upgrade, you may wish to remove files you no longer need. To do so, you need to know that certain files are handled specially during the upgrade. These include files that may have been edited by users, and shareable images that may have been linked into user images.

The distribution-kit copies of the following files become the latest version, and any pre-Version 4.0 copies become older versions of the files. Note that some of these files are new with Version 4.0 and will not be a part of your previous system.

```
[SYSLIB]LBRSHR.EXE
[SYSLIB]SUMSHR.EXE
[SYSLIB]VMSRTL.EXE
[SYSLIB]BASRTL2.EXE
[SYSLIB]BASRTL.EXE
[SYSLIB]CDDSHR.EXE
[SYSLIB]COBRTL.EXE
[SYSLIB]FORRTL.EXE
[SYSLIB]LIBRTL.EXE
[SYSLIB]LIBRTL2.EXE
[SYSLIB]MTHRTL.EXE
[SYSLIB]PASRTL.EXE
[SYSLIB]PLIRTL.EXE
[SYSLIB]RPGRTL.EXE
[SYSLIB]SCRSHR.EXE
[SYSLIB]SMGSHR.EXE
[SYSMGR]RTTLOAD.COM
[SYSMGR]SYSTARTUP.COM
[SYSMGR]SYCONFIG.COM
[SYSMGR]SYSHUTDWN.COM
[SYSMGR]STARTNET.COM
[SYSEXE]STARTUP.COM
[SYSEXE]SHUTDOWN.COM
[SYSEXE]STARTUP.MIN
[SYSHLP]HELPLIB.HLB
```

You may wish to reconcile the new versions with any local modifications to these files before deleting the older versions.

The upgrade procedure does not copy older versions of the following files to the system disk:

```
[SYSEXE]SYSUAF.DAT
[SYSEXE]NOTICE.TXT
```

The upgrade does not delete the following files, but you may delete them manually after the upgrade has completed:

```
[SYSEXE]STARTUP.UP5
[SYSEXE]UPGRADE.KIT
```

The upgrade procedure may have changed the size of the following files to better fit the system, so you may want to check these files to be sure that the sizes are appropriate:

```
[SYSEXE]SYSDUMP.DMP
[SYSEXE]PAGEFILE.SYS
[SYSEXE]SWAPFILE.SYS
```

6.4 Installed Images

As part of the upgrade, the file SYS$MANAGER:VMSIMAGES.DAT is created for your site. This is a combined list of files that are installed for enhanced system performance. Some files in this list may already be in your site's SYS$MANAGER:SYSTARTUP.COM file and should be removed.

6.5 Creating Alternate Roots on a Common System Disk

In a VAXcluster that features a common system disk, the MAKEROOT.COM command procedure is used to create system roots for nodes other than the first node of the cluster. To invoke MAKEROOT, enter this command:

```
$ @SYS$MANAGER:MAKEROOT
```

Note that MAKEROOT will abort if your system disk is not a cluster common system disk, or if the SYSGEN parameter VAXCLUSTER is 0.

When MAKEROOT executes, it requests the name of the new system root. Enter the root name using the form SYSx, where x is a hexadecimal digit in the range 1 through 9 and A through D (for example, SYS1 or SYSA). Note that system roots SYSE and SYSF are reserved for other system functions.

You may not specify the root of the currently running system (usually SYS0) and if you specify any other existing system root, you are asked if you wish to continue, that is, if you want to modify the existing system root. If you do want to modify the existing system root, MAKEROOT deletes the following files from it:

- [SYSEXE]MODPARAMS.DAT

- [SYSEXE]VAXVMSSYS.PAR

- [SYSMGR]VMSIMAGES.DAT

If a SYSCOMMON directory exists in the directory tree, it too will be deleted.

Note that MAKEROOT does not check the format of the existing system root. DIGITAL recommends you delete all the files in the directory tree, and the directory tree itself, or choose a different root.

Next, MAKEROOT asks for the new node name and its SCSSYSTEMID. The node name can be no more than six characters.

After you provide the new node name and the node's SCSSYSTEMID value, MAKEROOT prompts for the size of new root's page file and swap file. The values you provide are subsequently used by AUTOGEN.

Finally, MAKEROOT creates the new directory tree, the page file and the swap file. It also generates a VAXVMSSYS.PAR file for the new root using the SYSGEN parameters of the currently running node as the basis for the new node.

When MAKEROOT completes this, it displays the following message:

Now you must build console media for the new system. This
can be done in the following manner:

Copy all the files from the console to the disk, using
the following commands:

```
$ @SYS$UPDATE:CONSCOPY  ""  SAVE  ""  CSA1:
$ DISMOUNT CSA1:
```

Next, remove your console media, insert a scratch console media,
and enter the following commands:

```
$ @SYS$UPDATE:CONSCOPY  ""  RESTORE  ""  CSA1:
$ EXCHANGE COPY CSA1:DEFBOO.CMD *.*
```

Edit DEFBOO.CMD, and adjust the value deposited in R5 to contain
the root name in the high 4 bits. For example, if you are building
the console for root SYS5, set the value deposited in R5 to 50000000
(50004000 on a VAX-11/782).

After you edit DEFBOO.CMD, copy it to another disk file,GENBOO.CMD,
for instance, edit GENBOO.CMD to include the low bit set in R5.
For example, if R5 was set to 50000000 in DEFBOO.CMD,
set it to 50000001 in GENBOO.CMD. This will boot VAX/VMS
out of root 5, stopping in SYSBOOT.

```
$ EXCHANGE COPY DEFBOO.CMD,GENBOO.CMD CSA1:
$ DISMOUNT CSA1:
```

Remove the new console, and replace the original console
media in the drive.

Insert the newly-created console media in the console drive
on the new node.

You must now boot the target node into the newly-created root.
Use the conversational bootstrap GENBOO.CMD, which you just
created, as you must change the startup command procedure.

```
>>> @GENBOO.CMD
```

When SYSBOOT prompts, enter the following commands:

```
SYSBOOT> SET /STARTUP SYS$SYSTEM:STARTUP.COM
SYSBOOT> STARTUP_P1 "MIN"
SYSBOOT> CONTINUE
```

Following the system boot, login and invoke AUTOGEN with
this command:

```
$ @SYS$UPDATE:AUTOGEN SAVPARAMS REBOOT
```

When the system comes up, be sure to install the mandatory
update described at the start of this chapter.

7   Running the UETP

The User Environment Test Package (UETP) is a VAX/VMS software package designed to test whether the operating system itself is installed correctly. The UETP puts the system through a series of tests that simulate a typical user environment, making demands on the system that are similar to demands that might occur in everyday use.

The UETP is not a diagnostic; it does not attempt to test every feature exhaustively. When the UETP runs to completion without encountering nonrecoverable errors, the system being tested is ready for use.

The UETP exercises devices and functions that are common to all VAX/VMS systems, with the exception of optional features such as high-level language compilers. The system components tested include the following:

- Most standard peripheral devices

- The system's multiuser capability

- DECnet–VAX

- Cluster-wide file access and locks

The first time you run the UETP, you should run the entire package as described in Section 7.1, which contains a summary of operating instructions. If you need more detailed instructions, Section 7.2 of this guide describes how to prepare the UETP for testing; Section 7.3 gives complete instructions for running the entire UETP, as well as for running each phase individually.

If your UETP run results in a failure that you are unable to interpret, refer to Section 7.5 for an explanation of UETP output, error handling, and troubleshooting.

7.1 Summary of Operating Instructions

This section summarizes the procedures for running the entire UETP. You should follow these instructions when you run the test package for the first time. If you need further information, refer to Section 7.2 for detailed instructions on setup, or Section 7.3 for detailed instructions on running the UETP.

Note that the system must be bootstrapped before you can run the UETP. Complete bootstrap instructions are included in the *Guide to VAX/VMS System Management and Daily Operations.*

Do not run the UETP with other user programs or while user volumes are mounted. By design, the UETP assumes and requests the exclusive use of system resources. Unpredictable results could occur if you ignore this restriction. For example, because the UETP uses large amounts of many system resources such as memory pool space and disk space, it may interfere with applications that depend on these resources.

Logging In

Log in under the SYSTEST account as follows:

```
Username: SYSTEST
Password: UETP
```

Note: **The password on your system should have been changed by your system manager; if it has not, you should change it with the DCL command SET PASSWORD. Unauthorized use of this privileged account may compromise the security and privacy of your installation.**

Preparing Devices

After you log in, check all devices to be sure that the following conditions exist:

- All devices to be tested are powered up and on line to the system.

- Scratch disks are mounted and initialized

- Disks contain a directory named [SYSTEST] with OWNER_ UIC=[1,7]. (You can create this with the DCL command CREATE/DIRECTORY.)

- Magnetic tape reels are mounted on each drive to be tested and are initialized with the label UETP.

- Magnetic tape reels contain at least 600 feet of tape.

- Line printers and hard-copy terminals are well supplied with paper.

- Terminal characteristics and baud rate are set correctly (see the user's guide for your terminal).

Note that some communications devices need to be set up by Digital Field Service (see Section 7.2).

If you do not understand the instructions above or if you encounter any problems in preparing to run the UETP, read Section 7.2 before proceeding.

Invoking the UETP

Enter the following DCL command:

```
$ @UETP
```

The UETP responds with the following prompt:

```
Run "ALL" UETP phases or a "SUBSET" [ALL]? RET
```

Press RETURN (that is, choose the default response enclosed in brackets). The UETP will respond with three more questions in the following sequence:

```
How many passes of UETP do you wish to run [1]? RET
How many simulated user loads do you want [n]? RET
Do you want Long or Short report format [Long]? RET
```

Press RETURN after each prompt; when you have answered the last question, the UETP initiates its entire sequence of tests, which run to completion without further input from you. If the run completes successfully, your VAX/VMS system is in proper working order.

Note: **After a run of the UETP, you should always run the Error Log Utility to check for hardware problems that occur during a run of the UETP. For information on running the Error Log Utility, refer to the** *VAX/VMS Utilities Reference Volume.*

If you want to run the UETP without using the default responses, refer to Section 7.3, which explains the options.

If the UETP does not complete successfully (that is, without fatal errors), refer to Section 7.5 for troubleshooting information.

7.2 Setting Up the UETP

This section presents the detailed instructions for setting up the UETP and includes sections that discuss preparing the system and devices. Although you should usually run the entire UETP, you can select and run a subset of the phases by using the instructions described in this section.

7.2.1 Logging In

You should log in at the console terminal connected to the system. Running the UETP from a terminal other than the system console will mark the terminal available for testing during the initialization phase, although the device test phase will find the terminal to be unavailable.

All UETP files reside in SYS$TEST on the system disk as described in Section 7.2.1.1, unless you are running the UETP on a tailored system. The UETP files on a tailored system reside in LIB$SYSROOT:[SYSTEST] (see Section 7.2.4).

To access the UETP files, you must log in under the user name SYSTEST; if you do not, the UETP will fail. The password to this account in the system as it is distributed is UETP. Because the SYSTEST account has powerful privileges, you should change the password after the VAX/VMS system has been installed at your site, using the DCL command SET PASSWORD. Unauthorized use of this privileged account may compromise the security and privacy of your installation.

7.2.1.1   [SYSTEST] Directories

The UETP uses directories named [SYSTEST] for two purposes:

- To hold all the files used by the UETP command procedure (UETP.COM)

- To hold temporary files used by the UETP during testing

The following explanation is intended to eliminate confusion over references to directories named [SYSTEST] and to the logical name SYS$TEST.

All the files used by the UETP command procedure reside in a single directory on the system disk (the disk from which the system was bootstrapped). This directory is a rooted directory and is assigned the logical name SYS$TEST. The DCL command SHOW LOGICAL shows the translation of the logical name SYS$TEST on a typical system:

```
$ SHOW LOGICAL SYS$TEST
   "SYS$TEST" = "SYS$SYSROOT:[SYSTEST]"   (LNM$SYSTEM_TABLE)
```

If you want the UETP to test a particular disk, such as a scratch disk, you must create either a [SYSTEST] directory or a [SYS0.SYSTEST] directory on that disk. See Section 7.2.2.2 for additional information on setting up scratch disks for testing.

7.2.2   Preparing Devices

After logging in, you must set up all the devices to be tested. The following sections provide detailed instructions for preparing devices.

7.2.2.1   Setting Up the System Disk

Make sure that the system disk has at least 1200 blocks available when the UETP starts. Note that large systems, such as systems that run more than 20 load test processes, may require a minimum of 2000 available blocks. Running multiple passes of the UETP will cause log files to accumulate in the default directory and further reduce the amount of disk space available for subsequent passes.

If you have disk quotas enabled on the system disk, you should disable them before you run the UETP.

7.2.2.2	**Setting Up Additional Disk Drives**

The disk test uses most of the available free space on each
testable disk. The UETP estimates the space to be used by
UETDISK00 for normal testing as follows:

1 For each testable disk, the test tries to create two files.
The goal is to have the combined space of the two files
eventually reach 75% of the total space on the disk.
Typically, the files are created with 5% each of the total
space on the disk and are then extended in increments
of 5%. If the disk is nearly full to begin with, the initial
allocation and the steps are reduced to 1% of the total
space. If that fails, the test aborts.

2 The files are written and read asynchronously. After every
multiple of 20 writes for each file, the test tries to extend
the file by 5% of the total disk space (or by 1% if the file
was created with 1%). If the attempt to extend the file
fails, the test proceeds with the existing allocation. By
creating two files and extending two files, the UETP tends
to create fragmented files and situations that exercise the
disk. Extending the files this way allows the test to check
for exceeded quotas or a full disk and to adapt to the local
situation. Only the initial file creation can cause the test to
fail for lack of space.

To prepare each disk drive in the system for UETP testing,
perform the following steps:

1 Physically mount a scratch disk and start up the drive. If a
scratch disk is not available, any disk with a substantial
amount of free space will suffice; the UETP does not
overwrite existing files on any volume. If your scratch disk
contains files that you want to preserve, do not initialize the
disk; skip to Step 3.

2 If the disk has not been initialized, use the DCL command
INITIALIZE to do so, as follows:

```
$ INITIALIZE DMA0[:] TEST1
```

This command initializes a disk on an RK06 or RK07 drive
(DMA0:) and assigns the volume label TEST1 to the disk.
Codes for the various device types are listed in Chapter 4.
All volumes must have unique labels.

3 Issue the DCL command MOUNT/SYSTEM to connect the disk to the file system as follows:

```
$ MOUNT/SYSTEM DMA0[:] TEST1
```

This command mounts the volume labeled TEST1 on the drive DMA0:. The /SYSTEM qualifier indicates that you are making the volume available to all users on the system.

4 The UETP uses the [SYSTEST] directory when testing the disk. If the volume does not contain the directory [SYSTEST], issue the DCL command CREATE/DIRECTORY to create it.

```
$ CREATE/DIRECTORY/OWNER_UIC=[1,7] DMA0:[SYSTEST]
```

This command creates a [SYSTEST] directory on the drive DMA0: and assigns a user identification code (UIC) of [1,7]. You must have a UIC of [1,7] in order to run the UETP.

If the disk you have mounted contains a rooted directory structure, you can create the [SYSTEST] directory in the [SYS0.] tree.

7.2.2.3 **Setting Up Magnetic Tape Drives**

To set up each magnetic tape drive in the system, perform the following steps:

1 Mount a scratch volume with at least 600 feet of magnetic tape, making sure that the write-enable ring is in place; then switch on the device power.

2 Position the magnetic tape at the beginning-of-tape mark (BOT) and make sure the drive is on line.

3 Once you are logged in to the SYSTEST account, you must initialize each device on which you have mounted a scratch magnetic tape. For example, assume you have mounted a scratch magnetic tape on device MTA1:. Initialize the device as follows:

```
$ INITIALIZE MTA1[:] UETP
```

Only the magnetic tapes labeled UETP are written during the test run.

If you encounter a problem initializing the magnetic tape, or if
the test has a problem accessing the magnetic tape, refer to the
description of the DCL command INITIALIZE in the *VAX/VMS
DCL Dictionary*.

7.2.2.4 **Setting Up Terminals and Line Printers**
To be tested by the UETP, terminals and line printers must be
powered up and on line. If the device is a line printer, press
the ONLINE switch. Check that line printers and hard-copy
terminals are properly loaded with paper. The amount of paper
required depends on the number of UETP passes that you
intend to execute. Each pass requires two pages for each line
printer and hard-copy terminal.

Check that all terminals are set to the correct baud rate and
are assigned appropriate characteristics (see the user's guide for
your terminal).

Spooled devices and devices allocated to queues will fail the
initialization phase of the UETP and therefore will not be
tested.

Note that if you run the UETP from a terminal other than the
system console, the device test phase will display a message
that the terminal is unavailable for testing.

7.2.2.5 **Preparing the DEUNA for UETP Testing**
Make sure that no other processes are sharing the DEUNA
during a run of the UETP. The UETP procedure automatically
shuts down DECnet and the LAT-11 server for the duration
of the device tests, and restarts them when the device tests
are completed. However, you must shut down any local
applications.

7.2.2.6   Preparing the DR11–W for UETP Testing

The DR11–W using an internal logical loopback mode that tests all functionality except that of module connectors, cables, and transceivers. Since random external patterns are generated during this operation, the user device or other processor may need to be isolated from the DR11–W being tested until the testing is complete.

Note: Only DIGITAL field service personnel should set up the DR11-W for UETP testing.

To properly test the DR11–W, the E105 switchpack must be set as follows:

Switch 1	Switch 2	Switch 3	Switch 4	Switch 5
Off	On	Off	Off	On

When UETP testing is completed, a field service representative will restore the DMR11 to the proper configuration for operations.

7.2.2.7   Preparing the DR750 or the DR780 for UETP Testing

The following steps are necessary to prepare the DR750 or the DR780 for UETP testing:

Note: Only DIGITAL Field Service personnel should set up the DR750 or DR780 for UETP testing.

1 Copy the Version 1 DR780 microcode file, XF780.ULD, from the diagnostic medium (QE005–AY) to SYS$SYSTEM:, using the procedure described in the Installation Procedures (AA–J949A–TE) provided with the DR780 Microcode Kit.

2 Turn off the DR780 power.

3 Make the following DR780 backplane jumper changes:

- Remove the jumper from W7 and W8.
- Add a jumper from E04M1 to E04R1.
- Add a jumper from E04M2 to E04R2.

4 Disconnect the DDI cable from the DR780. This cable is either a BC06V–nn cable, which can be disconnected, or a BC06R–nn cable, which requires that you remove its paddle card from the backplane of the DR780.

5 Restore power to the DR780.

When UETP testing is completed, a field service representative will restore the DR750 or the DR780 to the proper configuration for operations.

7.2.2.8 **Preparing the MA780 for UETP Testing**

Make sure that the MA780 is set up according to the guidelines for shared memory in the *Guide to VAX/VMS System Management and Daily Operations*.

If you run the MA780 device test individually, the logical name CTRLNAME must be defined as MPM, regardless of the memory name. As an alternative, you can enter "MPM" in response to the controller designation prompt.

7.2.2.9 **Preparing a Second LPA11-K for UETP Testing**

If you have two LPA11-Ks, be sure that each is given the system-wide logical name in the SYS$MANAGER:LPA11STRT.COM file. The logical name for the second LPA11 should be LPA11$1.

7.2.2.10 **Devices Not Tested**

The UETP does not test the following devices; their status has no effect on UETP execution:

- Devices that require operator interaction (such as card readers)

- Software devices (such as the null device and local memory mailboxes)

The UETP does not explicitly test UDA, HSC, or CI devices; they are tested implicitly by the disk, magnetic tape, and DECnet tests.

The UETP does not test the console terminal or the console block storage device. If you are able to bootstrap the system, log in, and start the UETP, you have shown that these devices are usable.

7.2.3   Preparing for VAXcluster Testing

The UETP requires little additional preparation for the cluster-integration test phase, beyond the requirements for other UETP test phases and regular system management. The requirements for cluster integration testing are as follows:

1 Your system must be a member of a VAXcluster. If it is not, the UETP displays a message and does not attempt to run the test.

2 Your system must be consistent with other systems in the VAXcluster. The cluster-integration test lists each VAX/VMS system to be included in the lock tests along with the version of VAX/VMS that it is running. It checks to see that those systems use the same deadlock detection interval, and it takes advantage of the fact that a VAXcluster is a single entity for security purposes.

3 The SYSTEST_CLIG account must be present in the user authorization file, exactly as distributed by DIGITAL on each system in your VAXcluster. The SYSTEST_CLIG account parallels SYSTEST except that it is dedicated to the sole purpose of running the cluster-integration test.

4 All parts of the cluster-integration phase must reside in SYS$TEST on each system. The files UETCLIG00.COM and UETCLIG00.EXE are necessary for each system to be included in the test.

5 DECnet must be set up between the VAXcluster nodes; the UETP uses DECnet to create a process on those nodes. Virtually all checks that the test makes are dependent on the abilities to create the SYSTEST_CLIG processes and to communicate with them using DECnet.

6 There must be a [SYSTEST] or [SYS0.SYSTEST] directory on some disk available to the VAXcluster for each node (both VAX/VMS and HSC) in the cluster. The test uses the same directory as the UETP disk test to create a file on each cluster node and to see if some VAX/VMS node in the cluster can share access to that file. The requirement is only that there be one such directory per node; the test continues with the next cluster node once it has finished with a file.

7.2.4 Preparing a Small-Disk System

If you are running the UETP on a small-disk system, such as a VAX–11/730, you should be aware of the disk space limitations. Because of the space limitations, you need to perform special procedures to prepare for a UETP run. You can run the UETP in either of two ways on a small-disk system:

- From the library disk (LIB$SYSROOT)

- From the system disk, after copying the appropriate files

The following two sections describe the methods of preparing a small-disk system for UETP testing.

7.2.4.1 Running the UETP from the Library Disk

Running the UETP from the library disk is the easier of the two methods and leaves more space on your system disk; however, it allows the UETP to write to your library disk. If you plan to run the UETP from LIB$SYSROOT, use the following procedure:

1 Log in under the SYSTEST account.

2 Place the library disk in drive DQA1.

3 Invoke the tailoring procedure and mount the library disk as writeable with the following command:

```
$ @SYS$UPDATE:VMSTAILOR MOUNT/WRITE DQA1:
```

4 Start the UETP with the following command:

```
$ @LIB$TEST:UETP
```

7.2.4.2 **Running the UETP from the System Disk**

As an alternative, you can run the UETP from the system disk. This method of running the UETP allows you to reserve your library disk as read-only, but it is more complicated and uses more space on your system disk. (See Section 7.2.2.1 for a discussion of system disk space.)

If you plan to run the UETP from your system disk, you must first mount the library disk with the tailoring command MOUNT, then copy the UETP file group to your system disk. If you want to remove the library disk during a run of the UETP, you must tailor additional files to the system disk before you invoke the UETP.

1 Log in under the SYSTEST account.

2 Place the library disk in drive DQA1.

3 Invoke the tailoring procedure with the following command:

```
$ @SYS$UPDATE:VMSTAILOR
```

4 Enter the following sequence of tailoring commands:

```
Tailor> MOUNT DQA1:
Tailor> COPY UETP
```

5 If you want to remove the library disk during the UETP run, enter the additional tailoring commands:

```
Tailor> COPY /FILE CONVERT
Tailor> COPY /FILE [SYSLIB]CONVSHR
Tailor> COPY /FILE CDU
Tailor> COPY /FILE DIFF
Tailor> COPY /FILE SEARCH
Tailor> COPY /FILE [SYSHLP]HELPLIB.HLB
Tailor> EXIT
```

6 End tailoring with the EXIT command:

```
Tailor> EXIT
```

After you have completed these preparations, run the UETP according to the instructions in Section 7.3.

7.3 Interacting with the UETP

Each time you invoke the UETP, you specify the following
variables, either explicitly or implicitly:

- What part of the UETP to run (or whether to run the entire
  UETP)

- The number of consecutive passes to be made by the UETP

- The number of users to be simulated by the UETP in the
  system load test

- The amount of information to be output to the console

These variables are defined by your answers to the questions
that the UETP asks in the startup dialog. The following section
describes the startup dialog.

7.3.1 Startup Dialog

When you initiate the UETP, it responds with the following
prompt:

```
Run "ALL" UETP phases or a "SUBSET" [ALL]?
```

Throughout the startup dialog, brackets indicate the default
value, which you can choose by pressing RETURN. If you
choose ALL, the UETP responds with three more questions, as
described in Sections 7.3.1.1 through 7.3.1.3. If you want to
choose a subset, however, refer to Section 7.3.1.4.

7.3.1.1 Single Run Versus Multiple Passes

When you choose the default of ALL, the UETP prompts you
as follows:

```
How many passes of UETP do you wish to run [1]?
```

You can repeat the test run as many times as you wish, without
further input. If you enter 1 in response to the prompt (or press
RETURN for the default), the UETP stops after completing a
single run. If you specify a number greater than 1, the UETP
restarts itself continuously until it completes the number of
passes (runs) specified.

The UETP can be run once to check that the system is working, or it can be repeated many times to evaluate the system's response to continuous use. For example, a field service technician who is interested only in verifying that a newly installed system works might run the UETP once or twice, whereas a manufacturing technician might let the system run all night as part of the system integration and test.

When you specify multiple UETP runs, you may wish to request a short console log (see Section 7.3.1.3). Make certain that all line printers and hard-copy terminals have enough paper; each run requires two pages.

7.3.1.2 **Defining User Load for Load Test**

After you determine the number of passes, the UETP prompts you as follows:

```
How many simulated user loads do you want [n]?
```

Your response to this prompt provides information necessary to run the load test. The maximum number of users that you should enter will depend upon the amount of memory and the swapping and paging space in your VAX/VMS system. The UETP takes these factors into account when computing the default value.

If you want to see the equation the UETP uses to compute the default value, define the logical name MODE and run the UETINIT00 program as follows:

```
$ DEFINE MODE DUMP
$ RUN UETINITOO
        Welcome to VAX/VMS UETP Version V4.0
%UETP-I-ABORTC, UETINITOO to abort this test, type ^C
You are running on an 11/750 CPU with 8704 pages of memory.
The system was booted from _DRAO:[SYSO.].
Run "ALL" UETP phases or a "SUBSET" [ALL]? RET
How many passes of UETP do you wish to run [1]? RET
The default number of loads is the minimum result of
1) CPU_SCALE * ((MEM_FREE + MEM_MODIFY) / (WS_SIZE * PER_WS_INUSE))
        0.80 * ((    8704 +        323) / (    350 *          0.20)) = 8704
2) Free process slots                                               = 56
3) Free page file pages / Typical use of page file pages per process
                  18040 /                                   1000 = 18
How many simulated user loads do you want [18]?
Do you want Long or Short report format [Long]?
UETP starting at  2-OCT-1985 09:08:26.71 with parameters:
DEVICE LOAD DECNET CLUSTER  phases, 1 pass, 18 loads, long report.
$
```

This program does not initiate any phase; it serves only to display the equation used by the UETP to determine user load, and the specific factors that are employed in the current run.

You should respond to the dialog questions by pressing RETURN. After you respond to the first prompt, the program displays the expressions that determine the default number of simultaneous processes; the following definitions apply:

- CPU_SCALE refers to the relative processing power of the CPU. CPU_SCALE is determined as follows:

  VAX–11/730 = 0.5

  VAX–11/750 = 0.8

  VAX–11/780 = 1.0

  VAX–11/782 = 1.4

  VAX–11/785 = 1.5

- MEM_FREE represents memory in pages available to users

- MEM_MODIFY represents memory pages on the modified page list

- WS_SIZE represents working set size

- PER_WS_INUSE represents typical percentage of the working set in active use per process

UETINIT00 also displays the specific values represented by the expressions. In this example, the UETP selects 18 as the default for simulated user loads, because 18 is the minimum result of the three expressions.

You should deassign the logical name MODE before running the UETP, unless you prefer to see the above breakdown each time you run the UETP.

The purpose of the load test is to simulate a situation in which a number of detached processes (users) are competing for system resources. Each detached process executes a command procedure, thereby simulating a user entering commands from a terminal. When the command procedure is finished, the detached process logs out. You can monitor the number of currently active processes at the console, where both the beginning and the end of each process are displayed.

The load count is also used to limit the number of simultaneous tests during the device and DECnet test phases. Although the given default value is the best choice, you can increase or decrease the user load by entering your own response to the prompt. Be aware, however, that an increase might overload the system and cause failure through insufficient resources.

7.3.1.3   **Long and Short Report Format**
The following prompt allows you to choose one of two console report formats:

Do you want Long or Short report format [Long]?

If you choose the long report format (the default), the UETP sends to the console all error messages as well as information on the beginning and end of all phases and tests. The UETP records all of its output in the UETP.LOG file (see Section 7.5.3), regardless of your response to this question.

In many cases, it may not be convenient to have the UETP write the bulk of its output to the terminal. For example, if you run the UETP from a hard-copy terminal, the printing of all the output can slow the progress of the tests. This delay may not be a problem if you have requested only one run, but you

may prefer to use the short format if you intend to run multiple passes of the UETP from a hard-copy terminal.

If you request the short format, the UETP displays status information at the console, such as error messages and notifications of the beginning and end of each phase. This information enables you to determine whether the UETP is proceeding normally. If the short console log indicates a problem, you can examine UETP.LOG for further information. This disk file contains all of the output generated by the various phases, as well as the status information displayed at the console.

7.3.1.4 Running a Subset or Individual Phase

You can run a single phase by entering SUBSET or S in response to the following prompt:

```
Run "ALL" UETP phases or a "SUBSET" [ALL]?
```

The UETP prompts you for the desired phase as follows:

```
You can choose one or more of the following phases:
      DEVICE, LOAD, DECNET, CLUSTER
Phases(s):
```

There is no default; enter one or more phase names from the list. If you choose two or more phases, separate them with spaces or commas.

If your choice is LOAD, the UETP responds with three prompts, as in running the entire UETP. To run the LOAD phase, refer to Sections 7.3.1.1 through 7.3.1.3 and proceed as if you were running the entire test package.

If you choose a phase other than LOAD, the UETP responds with two prompts:

```
How many passes of UETP do you wish to run [1]?
Do you want Long or Short report format [Long]?
```

These questions are discussed in Sections 7.3.1.1 and 7.3.1.3 respectively. After you answer both questions, the phase you have selected runs to completion.

7.4 Using CTRL/Y and CTRL/C

The control characters CTRL/Y and CTRL/C allow you to terminate a UETP run before it completes normally. Normal completion of a UETP run, however, includes the deletion of miscellaneous files that have been created by the UETP for the purpose of testing. These cleanup procedures may be interrupted or prevented by the use of CTRL/Y or CTRL/C.

The effect of these control characters depends on what part of the UETP you are executing. For an explanation of the organization of the UETP and its components, refer to Section 7.6.

7.4.1 Using CTRL/Y

You can abort a run of the UETP command procedure by pressing CTRL/Y. Note, however, that cleanup of files and network processes in the [SYSTEST] directory may not be complete.

If you are running an individual test image (as described in Section 7.6.2.2), CTRL/Y interrupts the current UETP test and temporarily returns control to the command interpreter. While the test is interrupted, you can issue a subset of DCL commands that are performed within the command interpreter and thus do not cause the current image to exit. The *VAX/VMS DCL Dictionary* contains a table of commands performed within the command interpreter. In addition, you can issue any of the following:

- CONTINUE to continue the test from the point of interruption (except during execution of the cluster test).

- STOP to terminate the test; the test aborts and control returns to the command interpreter.

- EXIT to perform cleanup procedures and terminate the test (except during execution of the cluster test); control returns to the command interpreter.

- Any DCL command other than the subset mentioned above; the test performs cleanup procedures and terminates, and the DCL command executes.

Note: Use of the STOP command may prevent cleanup procedures that normally are executed. You should use the EXIT command if you want the image to perform its cleanup procedures before terminating.

7.4.2   Using CTRL/C

You can interrupt the command procedure by pressing CTRL/C, but you cannot continue a phase after you press CTRL/C. The UETP automatically goes to the next phase in the master command procedure.

Some UETP phases react to CTRL/C by cleaning up all activity and terminating immediately. The tests that have enabled CTRL/C in this way display the following message as they start to run:

```
%UETP-I-ABORTC, 'testname' to abort this test, type ^C
```

The phases that do not display the above message will terminate all processes they have started, but the processes may not have a chance to perform normal cleanup procedures.

If you are running an individual test image, however, you can use CTRL/C to terminate the execution of the image and perform cleanup procedures.

Note, however, that CTRL/C does not perform cleanup procedures for the cluster test.

7.5 Troubleshooting

This section explains the role of the UETP in interpreting operational errors in a VAX/VMS system. Section 7.5.4 discusses the most typical problems that can appear in a UETP run and describes how to identify them.

7.5.1 Relationship of the UETP to Error Logging and Diagnostics

When the UETP encounters an error, it reacts as a typical user program would react. It either returns an error message and continues, or it reports a fatal error and terminates the image or phase. In either case, the UETP does not attempt to discover the cause of the error.

The UETP assumes that the VAX/VMS hardware is correctly installed and operating as verified by a previous diagnostic run. If the UETP reports an error that you cannot diagnose, you can turn to the VAX/VMS error logging and diagnostic facilities. The system continuously maintains data on various kinds of errors. By running the Error Log Utility, you can obtain a detailed report of hardware and system errors that have occurred. Error log reports provide information about the state of the hardware device and I/O request at the time each error occurred. The diagnostic facilities can exhaustively examine a device or medium to isolate the sources of any errors.

For information about running the Error Log Utility, refer to the *VAX/VMS Utilities Reference Volume*.

7.5.2 Interpreting UETP Output

You can monitor the progress of the UETP tests at the terminal from which they were invoked. The terminal always displays status information, such as messages that announce the beginning and end of each phase and messages that signal an error. The tests send other types of output to various log files (which can include the issuing terminal) depending on how you invoked the tests. The log files contain output generated by the actual test procedures. Even if the UETP completes successfully, with no errors visibly displayed at the terminal, it is good practice to check these log files for errors. Furthermore, when errors are displayed at the terminal, check the log files for more information about their origin and nature.

The error messages that appear at the terminal and within the log files have two basic sources:

- The UETP tests themselves
- The system components that are tested

To interpret the messages, you may need to refer either to the *VAX/VMS System Messages and Recovery Procedures Reference Manual* or to the manual that describes the individual system component.

Several parts of the UETP, such as some device tests, UETINIT00.EXE, UETCLIG00.EXE, and UETDNET00.COM, allow you to obtain additional information concerning the progress of the test run or the problems it encounters. Because this information is usually insignificant, it is suppressed; however, if the tests can translate the logical name MODE to the equivalence string DUMP, they issue their messages with this information. Section 7.3.1.2 gives an example of this output for UETINIT00.EXE.

7.5.3 The Log Files

At the end of a run of the UETP, SYS$TEST contains a log file named UETP.LOG. This file contains all information generated by all UETP tests and phases. If the run involves multiple passes, you will find a version of the UETP.LOG for each pass.

Each time you run the test package, either in its entirety or by individual phase, in either single or multiple passes, the log file(s) named UETP.LOG will contain information from the latest run only. Each version of the UETP.LOG file from the previous run is renamed OLDUETP.LOG; the file(s) named OLDUETP.LOG will contain information from only one previous run.

The cluster test creates a NETSERVER.LOG file in SYS$TEST for each pass on each system included in the run. If the test is unable to report back errors (for example, if the connection to another node is lost), the NETSERVER.LOG file on that node will contain the result of the test run on that node. The UETP does not purge or delete NETSERVER.LOG files; therefore, you must delete them occasionally to recover disk space.

If a UETP run does not complete normally, SYS$TEST may contain other log files. Ordinarily, These log files are concatenated and placed within UETP.LOG. If, however, they appear on the system disk, they may be used for error checking, but they must be deleted before you run any new tests. You

)

may delete the other log files yourself or rerun the entire UETP, which checks for old UETP.LOG files and deletes them.

7.5.4 Possible UETP Errors

This section is designed to help you identify and solve problems that you may encounter during a run of the UETP. You should refer to this section if you need help in understanding a system failure and isolating its cause. This section is not intended as a repair manual and cannot be expected to diagnose any flaws in your system. It should, however, help you to interpret error messages and determine what action to take when you encounter them.

If you are unable to correct an error after following the steps in this guide, you should contact your DIGITAL Field Service representative. Any information that you can supply in this event, regarding the measures you have taken to isolate the problem, will help to diagnose the failure quickly.

The typical failures listed below appear in an order that roughly reflects the frequency with which they might occur in running the UETP:

1 Wrong quotas, privileges, or account

2 UETINIT01 failure

3 Insufficient disk space

4 Incorrect VAXcluster setup

5 Problems during the load test

6 DECnet error

7 Errors logged but not displayed

8 No PCB or swap slots

9 Hangs

10 Bugchecks and machine checks

The following sections describe how the UETP shows the above errors, and offer the best course of action for dealing with each problem.

7.5.4.1 Wrong Quotas, Privileges, or Account

If your assigned quotas or privileges do not match those that are standard for the SYSTEST account, UETP issues an error message as follows:

```
*********************
*  UETINIT00         *
*  Error count =  1  *
*********************
-UETP-W-TEXT,  The following:
        OPER privilege,
        BIOLM  quota,
        ENQLM  quota,
        FILLM  quota,
are non-standard for the SYSTEST account and may result in UETP errors.
```

In this example, the message informs you that the OPER privilege and the BIOLM, ENQLM, and FILLM quotas are either not assigned correctly or are not assigned at all.

Solution

In the system distributed by DIGITAL, the SYSTEST account has the following privileges and quotas assigned to it. If these are altered, the UETP does not run correctly.

Privileges

CMKRNL	CMEXEC	NETMBX	DIAGNOSE
DETACH	PRMCEB	PRMMBX	PHY_IO
GRPNAM	TMPMBX	VOLPRO	LOG_IO
SYSNAM	SYSPRV	SETPRV	GROUP

Quotas

BIOLM: 12	PRCLM: 8
DIOLM: 55	ASTLM: 55
FILLM: 20	BYTLM: 20480
TQELM: 20	CPU: no limit
ENQLM: 20	PGFLQUOTA: 10000

To check that these privileges and quotas remain assigned, use

the DCL commands SHOW PROCESS/PRIVILEGE and SHOW PROCESS/QUOTA. The terminal displays all privileges and quotas that are in effect for the current account:

```
$ SHOW PROCESS/PRIVILEGES
2-OCT-1985 18:06:02.89   OPA0:   User : SYSTEST
Process privileges :
    CMKRNL     may change mode to kernel
    CMEXEC     may change mode to exec
    SYSNAM     may insert in system logical name table
    GRPNAM     may insert in group logical name table
    DETACH     may create detached processes
    DIAGNOSE   may diagnose devices
    LOG_IO     may do logical I/O
    GROUP      may affect other processes in same group
    PRMCEB     may create permanent common event clusters
    PRMMBX     may create permanent mailbox
    SETPRV     may set any privilege bit
    TMPMBX     may create temporary mailbox
    NETMBX     may create network device
    VOLPRO     may override volume protection
    PHY_IO     may do physical I/O
    SYSPRV     may access objects via system protection
$ SHOW PROCESS/QUOTAS
2-OCT-1985 18:06:03.36          OPA0:         User : SYSTEST
Process Quotas:
    Account name: SYSTEST
    CPU limit :                     Infinite   Direct I/O limit :     55
    Buffered I/O byte count quota :    20480   Buffered I/O limit:    12
    Timer queue entry quota :             20   Open file quota :      20
    Paging file quota :                10000   Subprocess quota :     8
    Default page fault cluster :          32   AST limit :            55
    Enqueue quota :                       20
```

If any privileges or quotas are incorrect, you can run the Authorize Utility (AUTHORIZE) to add them (AUTHORIZE is explained in the *VAX/VMS Utilities Reference Volume*). As an alternative, you can temporarily assign the correct privileges with the DCL command SET PROCESS/PRIVILEGES.

If you are logged in to the wrong account, an error message informs you of the fact and asks you to log in to the SYSTEST account.

```
$ @UETP
**********************
*  UETINIT00        *
*  Error count =  1  *
**********************
-UETP-E-ABORT, UETINIT00 aborted at  2-OCT-1985 14:24:10.13
-UETP-E-TEXT, You are logged into the wrong account.
              Please login to the SYSTEST account.
$
```

7.5.4.2   **UETINIT01 Failure**

UETINIT01 failures are related to peripheral devices; this type of error message may indicate any of the following:

- Device failure

- Device not supported or not mounted

- Device allocated to another user

- Device write locked

- Lost vacuum

- Drive off line

In some cases, the course of action you should take is explicit in the error message. For example, you may receive a message from the Operator Communication Facility (OPCOM) process informing you of a problem and recommending a corrective measure:

```
%OPCOM,  2-OCT-1985 14:10:52.96, request 1, from user SYSTEST
Please mount volume UETP in device _MTA0:
%MOUNT-I-OPRQST, Please mount volume UETP in device _MTA0:
```

Other error messages may relate information in which the solution is implicit:

```
%UETP-S-BEGIN, UETDISK00 beginning at 2-OCT-1985 13:34:46.03
**********************
*  DISK_DRA         *
*  Error count =  1  *
**********************
-UETP-E-TEXT, RMS file error in file DRA0:DRA00.TST
-RMS-E-DNR, device not ready or not mounted
%UETP-S-ENDED, UETDISK00 ended at  2-OCT-1985 13:34:46.80
```

This message tells you that a disk device is either not ready or not mounted. From this information, you know where to look for the cause of the failure—at the disk device itself. If you cannot see the cause of the problem immediately, then check the setup instructions in Section 7.2.2.

In other cases, the cause of a failure may not be obvious from the information in the message. It is possible that the problem is related to hardware rather than software. For example, the DEUNA test may produce one of the following messages if the UETP does not have exclusive access:

* Inter-module cable unplugged

* Self-test failure code 0000000

To run the self-test diagnostic on the DEUNA successfully, the UETP needs exclusive access. Either DECnet or the LAT terminal server may also wish to use the DEUNA, which is a shareable device. The UETP shuts down DECnet and the LAT terminal server for the duration of the device tests and restarts them when the tests are completed.

Solution

To locate the source of the failure, you should try to determine where or when it occurs in the execution of the software:

* Run the device test individually (see Section 7.6.2.2). One reason for doing this is to determine whether the failure can be re-created. Also, you will find it easier to isolate the cause of the problem if you reproduce it using the least amount of software possible. For example, if the failure occurs only when you run the entire device phase, and not when you run the affected device test individually, then you can conclude that a device-interaction problem is indicated. Conversely, if you can re-create the error by running the single device test, then you have proven that the error is not related to device interaction.

* Run the device test with different media. If your run of the single device test succeeded in reproducing the error, the magnetic tape or disk media could be defective. Running the same test with new media will determine whether this is the problem.

(

- Call DIGITAL Field Service. If you have tried all of the above without solving the problem, you should contact your field service representative.

7.5.4.3   **Insufficient Disk Space**

When you run continuous passes of the UETP, log files accumulate on the disk from which the UETP was run and reduce the amount of free disk space available for each successive pass. If the available amount of disk space becomes too small for the current load, an error message indicates insufficient disk space:

```
%UETP-S-BEGIN, UETDISK00 beginning at  2-OCT-1985 08:12:24.34
%UETP-I-ABORTC, DISK_DMA to abort this test, type ^C

*********************
*  DISK_DMA         *
*  Error count = 1  *
*********************
-UETP-F-TEXT, RMS file error in file DMA0:DMA00.TST
-RMS-F-FUL, device full (insufficient space for allocation)

*********************
*  DISK_DMA         *
*  Error count = 2  *
*********************
-UETP-F-TEXT, RMS file error in file DMA0:DMA01.TST
-RMS-F-FUL, device full (insufficient space for allocation)
%UETP-E-DESTP, DISK_DMA stopped testing DMA unit 0 at 08:12:36.91
%UETP-S-ENDED, UETDISK00 ended at  2-OCT-1985 08:12:37.98
```

Solution

Make more space available on the disk; you can do this by using one or more of the following techniques:

- Delete unnecessary files to create more space.

- Purge files, if multiple versions exist.

- Mount a volume with sufficient space.

- Check for disk quotas that may be enabled on the disk. If they are enabled, either disable or increase them (see the *VAX/VMS Utilities Reference Volume* for a description of the Disk Quota Utility).

See Sections 7.2.2.1 and 7.2.2.2 for a further discussion of disk space.

)

7.5.4.4   **Incorrect Setup of a VAXcluster**

Most problems that can occur during the cluster-integration test are related to improper setup of the VAXcluster or of the UETP on the VAXcluster. These problems are most likely to occur at the following stages of the VAXcluster test:

- Near the beginning, when processes on VAX/VMS nodes are started

- Toward the end, when cluster file access is checked

The cluster test phase shows that various VAX/VMS nodes in your cluster can simultaneously access files on each node in the cluster. For each node in the cluster, the test looks for some disk device that is visible to all nodes in the cluster and that is set up so that the test can create a file on it. The requirements for creating a file in the cluster test phase are similar to those of the UETP disk test (UETDISK00.EXE):

- There must be a [SYSTEST] directory on the disk in either the master file directory (MFD) or in the root directory [SYS0.].

- The [SYSTEST] directory must be protected so that the SYSTEST account can create a file in it.

The cluster test phase need only create a small file. If there is not such a disk on some cluster node, the test gives a warning and proceeds to the next cluster node.

Whenever you suspect a problem, you should try to recover the SYS$TEST:NETSERVER.LOG file that was created when the SYSTEST_CLIG process was created. This file may contain additional error information that could not be transmitted to the node running the test. If it was not possible to create the SYSTEST_CLIG process on some node, the system accounting file for that node may contain a final process status in a process termination record.

An error message may indicate a problem with any of the following during a run of the cluster test:

- Logging in at other nodes—This problem is due to incorrect setup for the cluster test at the remote VAX/VMS node; refer to Section 7.2.3 for information on preparing for VAXcluster testing.

- Communicating with other nodes—A message indicates a DECnet problem. Check the NETSERVER.LOG file on the affected node to determine the cause.

- Taking out locks or detecting deadlocks—The most likely cause of this problem is that you are not logged in to the SYSTEST account. Otherwise, your cluster is not properly configured; you should send an SPR.

- Creating files on VAXcluster nodes—This problem is due to incorrect setup for the cluster test; refer to Section 7.2.3 for information on preparing for VAXcluster testing.

7.5.4.5   **Problems During the Load Test**
A variety of errors can occur during the load test, because the command procedures that are invoked during the tests run several utilities and perform many functions. Tracking a problem can be difficult because the UETP deletes the log files that are generated during the load test (see Section 7.6.3).

If a problem occurs during the load test and the cause is not obvious, you can modify UETP.COM to preserve the log files.

1 Append /NODELETE to the following line:

```
$ TCNTRL UETLOAD00.DAT/PARALLEL_COUNT='LOADS/REPORT_TYPE='REPORT
```

2 Delete the following line:

```
$ DELETE UETLO*.LOG;*
```

Rerun the load test with these changes to try to re-create the problem.

If you re-create the problem, examine the contents of the appropriate log file. To determine which log file to read, you need to understand the scheme by which the load test names its processes and log files. (The log file names are derived from the process names.)

The load test creates processes that are named in the following format: UETLOADnn_nnnn

For example:

```
%UETP-I-BEGIN, UETLOAD00 beginning at 22-AUG-1985 15:45:08.97
%UETP-I-BEGIN, UETLOAD02_0000 beginning at 22-AUG-1985 15:45:09.42
%UETP-I-BEGIN, UETLOAD03_0001 beginning at 22-AUG-1985 15:45:09.63
%UETP-I-BEGIN, UETLOAD04_0002 beginning at 22-AUG-1985 15:45:10.76
%UETP-I-BEGIN, UETLOAD05_0003 beginning at 22-AUG-1985 15:45:11.28
%UETP-I-BEGIN, UETLOAD06_0004 beginning at 22-AUG-1985 15:45:12.56
%UETP-I-BEGIN, UETLOAD07_0005 beginning at 22-AUG-1985 15:45:13.81
%UETP-I-BEGIN, UETLOAD08_0006 beginning at 22-AUG-1985 15:45:14.94
%UETP-I-BEGIN, UETLOAD09_0007 beginning at 22-AUG-1985 15:45:16.99
%UETP-I-BEGIN, UETLOAD10_0008 beginning at 22-AUG-1985 15:45:19.32
%UETP-I-BEGIN, UETLOAD11_0009 beginning at 22-AUG-1985 15:45:19.94
%UETP-I-BEGIN, UETLOAD02_0010 beginning at 22-AUG-1985 15:45:20.20
%UETP-I-BEGIN, UETLOAD03_0011 beginning at 22-AUG-1985 15:45:21.94
%UETP-I-BEGIN, UETLOAD04_0012 beginning at 22-AUG-1985 15:45:22.99
```

Note that if more than ten processes are created, the numbering sequence for the UETLOADnn portion of the process name starts over at UETLOAD02; however, the four digits of the _nnnn portion continue to increase.

Each load test process creates two log files. The first log file is created by the test controller; the second log file is created by the process itself. The log file that you need to examine for error information on any given load test process is the one that was created by the test controller (the first log file).

The load test log file derives its file name from the process name, appending the last four digits of the process name (from the _nnnn portion) to UETLO. The test-controller log file and the process log file for each process use the same file name; however, the process log file has the higher version number of the two. For example, the log files created by the process UETLOAD05_0003 would be named as follows: UETLO0003.LOG;1 (test-controller log file)

UETLO0003.LOG;2 (process log file)

Make certain that you read the log file with the lower version number, as that file is the one that contains the load test commands and error information.

After you have isolated the problem, restore UETP.COM to its original state and delete the log files that remain from the load test (UETLO*.LOG;*); failure to delete these files may result in disk space problems.

7.5.4.6   **DECnet Error**
A DECnet error message may indicate that the network is not available.

Solution

- If the DECnet license is included in your system, turn it on (see the installation documentation for DECnet–VAX).

- If DECnet is not included in your system, ignore the message; this is normal and the UETP run will not be affected.

If you encounter other DECnet-related errors, you should do the following:

- Run DECnet as a single phase (see Section 7.3.1.4) to determine whether the error can be re-created.

- Refer to the *VAX/VMS System Messages and Recovery Procedures Reference Manual* or the installation documentation for DECnet–VAX.

7.5.4.7   **Errors Logged But Not Displayed**
If no errors are shown at the console terminal or reported in the UETP.LOG file, you should run the Error Log Utility to see if any errors were logged in the ERRLOG.SYS file. See the *VAX/VMS Utilities Reference Volume* for information about running the Error Log Utility.

7.5.4.8   **No PCB or Swap Slots**
An error message indicates that no process control block (PCB) or swap slots are available. For example:

```
%UETP-I-BEGIN, UETLOAD00 beginning at  2-OCT-1985 07:47:16.50
%UETP-I-BEGIN, UETLOAD02_0000 beginning at  2-OCT-1985 07:47:16.76
%UETP-I-BEGIN, UETLOAD03_0001 beginning at  2-OCT-1985 07:47:16.92
%UETP-I-BEGIN, UETLOAD04_0002 beginning at  2-OCT-1985 07:47:17.13
%UETP-I-BEGIN, UETLOAD05_0003 beginning at  2-OCT-1985 07:47:17.35
%UETP-I-BEGIN, UETLOAD06_0004 beginning at  2-OCT-1985 07:47:17.61
%UETP-W-TEXT, The process -UETLOAD07_0005- was unable to be created,
  the error message is
-SYSTEM-F-NOSLOT, no pcb or swap slot available
%UETP-W-TEXT, The process -UETLOAD08_0006- was unable to be created,
  the error message is
-SYSTEM-F-NOSLOT, no pcb or swap slot available
%UETP-W-TEXT, The process -UETLOAD09_0007- was unable to be created,
  the error message is
-SYSTEM-F-NOSLOT, no pcb or swap slot available
%UETP-W-TEXT, The process -UETLOAD10_0008- was unable to be created,
  the error message is
-SYSTEM-F-NOSLOT, no pcb or swap slot available
%UETP-W-TEXT, The process -UETLOAD11_0009- was unable to be created,
  the error message is
-SYSTEM-F-NOSLOT, no pcb or swap slot available
%UETP-W-ABORT, UETLOAD00 aborted at  2-OCT-1985 07:47:54.10
-UETP-W-TEXT, Aborted via a user CTRL/C.
 *************************************************
 *                                               *
    END OF UETP PASS 1 AT  2-OCT-1985 07:48:03.17
 *                                               *
 *************************************************
```

Solution

To solve this problem, do the following:

- Rerun individually the phase that caused the error message (which would be the LOAD phase in the above example), to see if the error can be reproduced.

- Increase the size of the page file, using either the command file SYS$UPDATE:SWAPFILES.COM or the System Generation Utility (see the *VAX/VMS Utilities Reference Volume.*)

- Increase the MAXPROCESSCNT SYSGEN parameter, if necessary, and reboot the system.

- Increase both the page file size and the MAXPROCESSCNT, if necessary.

7.5.4.9 Hangs

If there is no keyboard response and/or system disk activity, the system may be hung.

Solution

A system hang can be difficult to trace; you should always save the dump file for reference purposes. To learn why the system hung, run the System Dump Analyzer as described in the *VAX/VMS Utilities Reference Volume*. Possible reasons include the following:

- Insufficient pool space—Reboot the system with a larger value for NPAGEVIR.

- Insufficient page file space—Increase the page file space using the System Generation Utility as described in the *VAX/VMS Utilities Reference Volume*.

- I/O device failure causing driver-permanent loop—Call DIGITAL Field Service.

7.5.4.10 Bugchecks and Machine Checks

When the system aborts its run, a bugcheck message appears at the console.

Solution

Call DIGITAL Field Service. This is often a hardware problem and there is no easy way to solve a bugcheck or machine check. It is important, however, that you save the SYS$SYSTEM:SYSDUMP.DMP and ERRLOG.SYS files so that they are available for examination. It is also important to know whether the failure can be re-created; you can verify this by running the UETP again.

7.6   UETP Tests and Phases

This section explains the organization of the UETP and the individual components within the test package. Section 7.6.2.2 provides instructions for testing specific devices.

You run the UETP by invoking a master command procedure, which contains commands that in turn invoke each test phase. The procedure begins by prompting you for responses that will provide information needed by the various test phases. (See Section 7.3 for a detailed description of invoking the UETP.)

The master command procedure, UETP.COM, contains commands that initiate each test phase in turn and perform such tasks as defining logical names and manipulating files generated by the tests.

The UETP.COM procedure also issues commands to invoke the phase controlling program, UETPHAS00.EXE, which in turn controls each test phase. The phase controller starts up multiple detached processes and reports their completion status and other information the processes relate back to it.

The following sections describe the various UETP test phases.

7.6.1   Initialization Phase

The initialization phase prepares for the actual UETP tests in the following ways:

- The image UETINIT00.EXE prompts for input from the terminal (see Section 7.3). The input defines certain variables that affect the execution of the UETP tests.

- The image UETINIT01.EXE gathers information on all the controllers in the system and on their associated devices. This image writes the information into a file called UETINIDEV.DAT.

- Using the information in UETINIDEV.DAT, UETINIT01.EXE verifies which devices in the system are operable by running the appropriate device test, which performs a simple read /write operation to each device. If a device fails this test, the device's entry in UETINIDEV.DAT specifies that the device cannot be tested. As a result, subsequent UETP tests ignore that device.

7–35

- For each testable controller, UETINIT01.EXE writes a line
  into a file called UETCONT00.DAT. The line associates a
  test file with the controller it is to test.

A summary of UETINIDEV.DAT always exists in UETP.LOG,
and UETINIT01.EXE outputs that summary to the console if
you have requested the long report format.

7.6.2 Device Test Phase

The device test phase includes separate tests for each type
of device, such as disks, magnetic tapes, line printers, and
terminals. This section explains the device test phase and
presents instructions for running a single device test. If you
want to run the entire device test phase individually, refer to
Section 7.3.1.4.

7.6.2.1 How the Device Phase Works

The UETP device test phase contains an executable image,
the phase controller UETPHAS00, which creates a detached
process for every device controller to be tested. For example,
if a system includes three terminal controllers, one line printer
controller, and two disk controllers, the image creates six
detached processes. In parallel, the detached processes execute
images that test the various types of devices.

The initialization phase of the UETP creates a file called
UETINIDEV.DAT and a file called UETCONT00.DAT.
UETINIDEV.DAT contains data on the VAX/VMS-supported
controllers in the system and their associated devices, while
UETCONT00.DAT associates a device test image with each
testable controller. UETPHAS00 uses the information in
UETCONT00.DAT to find a device controller name to pass to
each detached process that it creates. UETPHAS00 passes the
controller name by writing it to a mailbox that is SYS$INPUT
to individual tests. Each detached process uses that data to
determine which controller to test. The test image then searches
UETINIDEV.DAT for the device controller and for all testable
units on that controller. The phase controller terminates when
all devices on all controllers have completed testing.

Because UETCONT00.DAT is deleted automatically at the end of a UETP run, you cannot run the device phase unless you invoke UETP.COM; you can run only individual test images. UETINIDEV.DAT exists in SYS$TEST unless you explicitly delete it.

7.6.2.2   **Running a Single Device Test**
You must be logged in to the SYSTEST account to run the individual tests as described in this section. Also, a copy of UETINIDEV.DAT must exist; if it is not present from a previous run (a run of the entire UETP or a run of the device test phase creates UETINIDEV.DAT), you can create it yourself. Note that when you run a single test, no log file is created; the test sends all its output to your terminal.

If you do not want to test all the device types, you can test a specific controller by choosing a test image name from Table 7–1 and executing it as in the following example:

```
$ RUN UETTTYS00
Controller designation?: TTB
```

The UETP prompts you for the controller designation, which you supply together with the device code. Unless you are testing your own terminal, you must explicitly designate a controller name. If you are running the terminal test, you can press RETURN to test your terminal only.

If you plan to repeat the run a number of times, you may find it more convenient to use the DEFINE command as follows:

```
$ DEFINE CTRLNAME TTB
$ RUN UETTTYS00
```

When you define the controller name in this way, the logical name CTRLNAME remains assigned after the test completes. To deassign this logical name, use the DCL command DEASSIGN as follows:

```
$ DEASSIGN CTRLNAME
```

Format of UETINIDEV.DAT

The UETINIDEV.DAT file is an ASCII sequential file that you
can type out or edit if necessary. The contents of this file are as
shown in the following command sequence:

```
$ TYPE UETINIDEV.DAT
DDB X DDD
UCB Y UUUUU
END OF UETINIDEV.DAT
```

The symbols in this example are defined as follows:

Symbol	Definition
X	T, if there are any testable units for this controller
X	N, if this controller is not to be tested
Y	T, if this unit is testable
Y	N, if this unit is not testable
DDD	device controller name
UUUUU	device unit number

UETINIDEV.DAT contains a DDB (device data block) line for
each controller connected or visible to your system. Following
the DDB line is a UCB (unit control block) line for each unit
connected to that controller. In addition, if your system uses
MA780 memory in a loosely coupled CPU configuration,
UETINIDEV.DAT includes one UCB line for each MA780
memory. A device test will test a particular device only if
both the DDB line and the UCB line indicate that the device is
testable.

Running a Test in Loop Mode

If you want to put extra stress on a device, you can run
the device test in loop mode, which causes the test to run
indefinitely. For example:

```
$ DEFINE MODE LOOP
$ RUN UETDISK00
Controller designation?: DRA
%UETP-I-TEXT, End of pass 1 with 980 iterations at 26-OCT-1985 16:18:51:03
^C
```

You must use CTRL/C to terminate the test run. If you use CTRL/Y, the UETP will not perform cleanup procedures.

Functions of Individual Device Tests

For each disk in the system, the disk test allocates two files into which it randomly writes blocks of data. The test then checks the data, reports any errors to SYS$OUTPUT, and deletes the disk files.

When you run the disk test phase in a cluster environment, the test will access all disks that are mounted by the system being tested, and users of the disk being tested may encounter an insufficient disk space problem. Unless you issue a warning to the users on a remote node, they will be unaware that the UETP is testing a disk that they may be using.

The magnetic tape test exercises all the magnetic tape drives in the system. The test creates a large file on each mounted magnetic tape, into which it writes multiple sequential records of varying sizes. After writing the records, the test rewinds the magnetic tape, validates the written records, and finishes by reinitializing the magnetic tape.

The terminal and line printer test generates several pages or screens of output, where each page or screen contains a header line and a test pattern of ASCII characters. A header line contains the test name, the device name, the date, and the time.

For the laboratory peripheral accelerator (LPA11-K), the test image determines the configuration on the LPA11-K's I/O bus. The image loads all types of microcode to LPA11-K and reads or writes data for each device on the LPA11-K I/O bus.

The communications device tests fill the transmit message buffer with random data; then, using loopback mode, they transmit and receive the message several times. To check that the looped-back data is correct, an AST routine is associated with $QIO read to compare the received message against the transmitted message. The procedure is repeated using messages of different lengths.

The interface device tests put their respective devices in maintenance mode, write random data, and then verify the data.

The MA780 device test creates and modifies mailboxes, common event flags, and global sections in shared memory; it then verifies that modifications can be made. You can run MA780 tests in parallel from separate systems so that the tests interact with each other through common MA780 memories.

The DEUNA test performs self-test diagnostics on the device and then performs read and write operations with test data that uses various DEUNA modes (such as internal loopback, external loopback, and broadcast).

Table 7–1 lists the device test images and the devices to be tested.

Table 7–1   The Device Tests

Test Image Name	Devices Tested
UETDISK00.EXE	Disks
UETTAPE00.EXE	Magnetic Tapes
UETTTYS00.EXE	Terminals and Line Printers
UETLPAK00.EXE	LPA11-K
UETCOMS00.EXE	DMC-11, DMR-11
UETDMPF00.EXE	DMF-32, DMP-11
UETDR1W00.EXE	DR11–W
UETDR7800.EXE	DR780, DR750
UETMA7800.EXE	MA780
UETUNAS00.EXE	DEUNA

7.6.3   System Load Test Phase

The purpose of the system load test is to simulate a number of terminal users who are simultaneously demanding system resources. The system load tests, directed by the file UETLOAD00.DAT, create a number of detached processes that execute various command procedures. Each process simulates a user logged in at a terminal; the commands within each procedure are the same type of commands that a user enters from a terminal. The load test creates the detached processes in quick succession, and generally the processes execute their command procedures simultaneously. The effect on the system is analogous to an equal number of users concurrently issuing

commands from terminals. In this way, the load test creates an environment that is similar to normal system use.

The load test uses the logical name LOADS to determine the number of detached processes to create. When you initiate the UETP command procedure, it prompts for the number of users to be simulated (see Section 7.3.1.2) and consequently the number of detached processes to be created. Your response, which depends on the amount of memory and the swapping and paging space in your system, defines the group logical name LOADS.

The UETP master command procedure deassigns all group logical names assigned by its tests as part of the termination phase. The group logical name LOADS remains assigned only if the UETP package does not complete normally.

The command procedures executed by the load test can generate a large amount of output, depending on the number of detached processes created. For each detached process (or user), the test creates a version of an output file called UETLOnnnn.LOG ("nnnn" represents a string of numeric characters). The console displays only status information as the load test progresses.

Whether the load test runs as part of the entire UETP or as an individual phase, the UETP excerpts the UETLOnnnn.LOG files, writes the output to the file UETP.LOG, and deletes the individual output files.

You can run the system load test as a single phase by selecting LOAD from the choices offered in the startup dialog (see Section 7.3.1.4).

7.6.4   DECnet Test Phase

If DECnet is included in your VAX/VMS system, a run of the entire UETP automatically tests DECnet hardware and software. Because communications devices are allocated to DECnet and the DECnet devices cannot be tested by the UETP device test, the UETP shuts down DECnet for the duration of the initialization and device test phases (approximately 3 to 5 minutes each). It then turns DECnet on again after those phases are completed. The DECnet node and circuit counters

are zeroed at the beginning of the DECnet test to allow for
failure monitoring during the run.

As with other UETP phases, you can run the DECnet phase
individually by following the procedure described in Section
7.3.1.4.

7.6.4.1 Environment

The DECnet test will work successfully on VAX/VMS systems
connected to all DECnet-supported node types, including
routing and nonrouting type nodes and several different types
of operating systems (such as RSTS, RSX, TOPS, and RT).
There must be some sort of default access on remote systems in
order to copy files between systems. The DECnet phase tests
the following:

- The node the UETP is running on

- All circuits in sequence

- All adjacent or first-hop nodes, and all circuits in parallel

There is no limitation on the number of communication lines
supported by the tests. A test on one adjacent node should last
no more than two minutes at normal communications transfer
rates.

7.6.4.2 How the DECnet Phase Works

Controlled by UETPHAS00.EXE, the UETP reads the file
UETDNET00.DAT and performs the following steps to
complete the DECnet phase:

1 Executes a set of NCP LOOP EXECUTOR commands to test
the node on which the UETP is running.

2 Uses NCP to execute the command SHOW ACTIVE
CIRCUITS. The results are placed in UETININET.TMP, from
which the UETP creates the data file UETININET.DAT. The
UETININET.TMP file contains the following information for
any circuit in the ON state but not in transition:

- The circuit name

- The node address

- The node name (if one exists)

7–42

The UETININET.TMP file is used throughout the DECnet phase to determine which devices to test.

3 Uses the UETININET.TMP file to create an NCP command procedure for each testable circuit; each command procedure contains a set of NCP commands to zero the circuit and node counters and to test the circuit and adjacent node by copying files back and forth.

Note: **If you do not want the counters zeroed, do not test DECnet.**

4 Executes the command procedures from step 3 in parallel to simulate a heavy user load. The simulated user load is the lesser of the following values:

- The number of testable circuits, multiplied by 2

- The maximum number of user-detached processes that can be created on the system before it runs out of resources (determined by UETINIT00)

5 Executes a program, UETNETS00.EXE, that uses the UETININET.DAT file to check the circuit and node counters for each testable circuit. If a counter indicates possible degradation (by being nonzero), its name and value is reported to the console. All counters are reported in the log file, but only the counters that indicate degradation are reported to the console. Following is an example of UETNETS00 output:

```
%UETP-S-BEGIN, UETNETS00 beginning at  2-OCT-1985 13:45:33.18
%UETP-S-TEXT, Circuit DMC-0 to (NODENAME1) OK.
%UETP-I-TEXT, Node  (NODENAME2) over DMC-1 response timeouts = 1.
%UETP-I-TEXT, Circuit DMC-1 to (NODENAME2) local buffer errors = 34.
%UETP-I-TEXT, Node  (NODENAME3) over DMP-0 response timeouts = 3.
%UETP-S-ENDED, UETNETS00 ended at  2-OCT-1985 13:45:36.34
```

Since degradation is not necessarily an error, only you can determine the success of the test. The following counters are considered to indicate possible degradation:

For Circuits

* Arriving congestion loss
* Corruption loss
* Transit congestion loss
* Line down
* Initialization failure
* Data errors inbound
* Data errors outbound
* Remote reply timeouts
* Local reply timeouts
* Remote buffer errors
* Local buffer errors
* Selection timeouts
* Remote process errors
* Local process errors
* Locally initiated resets
* Network initiated resets

For Nodes

* Response timeouts
* Received connect resource errors
* Aged packet loss
* Node unreachable packet loss
* Node out of range packet loss
* Oversized packet loss
* Packet format error
* Partial routing update loss
* Verification reject

7.6.5   Cluster-Integration Test Phase

The cluster-integration test phase consists of a single program and a command file, and depends heavily on DECnet. It uses DECnet to create SYSTEST_CLIG processes on each VAX/VMS node in the cluster and to communicate with each node. SYSTEST_CLIG is an account that is parallel to SYSTEST, but limited so that it can only be used as part of the cluster-integration test. The following restrictions on the SYSTEST_CLIG account are necessary for a correct run of the cluster test phase:

- The account password must be empty.

- The UIC must be the same as that of the SYSTEST account.

- The account can allow login only via DECnet.

- The account must be locked into running UETCLIG00.COM when it logs in.

The latter two items are necessary to ensure the security and privacy of your system. If the test cannot create a SYSTEST_CLIG process on some VAX/VMS node, it gives the reason for the failure and ignores that node for the lock tests and for sharing access during the file test. The test makes no attempt to report information relating to a failure at the node where creation was attempted; that is, any possible log file is not copied to the node running the test. If there is a problem communicating with a SYSTEST_CLIG process after it has been created, the test excludes it from further lock and file sharing tests. At the end of the cluster-integration test, an attempt is made to report any errors seen by that node.

UETCLIG00.EXE has two threads of execution. The first, or master thread, checks the cluster configuration; that is, the VAX/VMS and HSC nodes and the disks attached to each of them that can be seen from the node running the test. For each VAX/VMS node, it attempts to start up a SYSTEST_CLIG process via DECnet. Those nodes where it succeeds run the command file, UETCLIG00.COM, which starts up UETCLIG00.EXE and runs the second (slave) execution thread.

The process running the master thread checks to see that it can communicate with the processes running the slave threads. It then instructs them to take out locks so that a deadlock situation is created.

The master tries to create a file on some disk on each
VAX/VMS and HSC node in the cluster. It writes a block,
reads it back and verifies it. The master selects one VAX/VMS
node at random and asks that node to read the block and verify
it. The master extends the file by writing another block and has
the slave read and verify the second block. The file is deleted.

The slave processes exit. They copy to the master process
the contents of their SYS$ERROR files, so that the UETP
log file and console report show all problems in a central
place. DECnet automatically creates a NETSERVER.LOG in
SYS$TEST as the test is run, so that if necessary, you can read
that file later from the node in question.

During the test run, the master process announces the
beginning and ending of the test using cluster $BRKTHRU
to each VAX/VMS node's console.

You can define the logical name MODE to the equivalence
string DUMP to trace most events as they occur.

7.6.6   Termination of the UETP

At the end of a UETP pass, the master command procedure
UETP.COM displays the time at which the pass ended. In
addition, UETP.COM determines whether the UETP needs to
be restarted. (You can request multiple passes when you start
up the test package; see Section 7.3.1.1.)

At the end of an entire UETP run, UETP.COM deletes
temporary files and performs other cleanup activity.

A Files of the VAX/VMS System

This appendix contains the names and brief descriptions of
the files provided by DIGITAL on the VAX/VMS system
distribution medium. The files on this medium are cataloged in
eleven directories, three of which are reserved for system use.
The names of the remaining eight directories and descriptions
of their contents follow.

1 [SYSCBI]

This directory is reserved for EDTCAI files.

2 [SYSERR]

This directory is reserved for the error log file
(ERRLOG.SYS).

3 [SYSEXE]

This directory, listed in Table A–1, contains commonly used
executable images of the operating system.

4 [SYSHLP]

This directory, listed in Table A–2, contains text libraries for
the Help Utility and other components.

5 [SYSHLP.EXAMPLES]

This subdirectory, listed in Table A–3, contains sample
driver programs, user-written system service programs, and
other source code examples of interest.

6 [SYSLIB]

This directory, listed in Table A–4, contains various macro
and object libraries as well as other files used for general
reference.

7 [SYSMAINT]

This directory is reserved for system hardware diagnostic programs.

8 [SYSMGR]

This directory, listed in Table A-5, contains files used in managing the operating system. This directory is the default directory for the system manager's account.

9 [SYSMSG]

This directory, listed in Table A-6, contains system message text files.

10 [SYSTEST]

This directory, listed in Table A-7, contains files used to run the User Environment Test Package (UETP).

11 [SYSUPD]

This directory, listed in Table A-8, contains files used in applying system updates.

Table A-1   Files Contained in Directory [SYSEXE]

File Name	Description
ACC.EXE	Accounting Utility
ACLEDT.EXE	Access Control List (ACL) Editor
ANALIMDMP.EXE	ANALYZE/PROCESS_DUMP image
ANALYZBAD.EXE	ANALYZE/MEDIA image
ANALYZOBJ.EXE	ANALYZE/IMAGE and ANALYZE/OBJECT image
ANALYZRMS.EXE	ANALYZE/RMS_FILE image
AUTHORIZE.EXE	User authorization program
AUTOGEN.PAR	Parameter file for the AUTOGEN command procedure
BACKUP.EXE	Backup Utility
BADBLOCK.EXE	Dynamic bad block Files-11 ACP subprocess
BOOT58.EXE	Reserved for future use
BOOTBLOCK.EXE	Reserved for future use
CDU.EXE	Command Definition Utility

Table A-1 (Cont.)   Files Contained in Directory [SYSEXE]

File Name	Description
CHECKSUM.EXE	Used during installation of VAX/VMS updates
CLUSTRLOA.EXE	Loadable VAXcluster support code
CNDRIVER.EXE	DECnet CI datalink driver
CONFIGURE.EXE	Dynamic device configure process
CONINTERR.EXE	Connect-to-Interrupt driver
CONVERT.EXE	Convert Utility
COPY.EXE	Copy utility
CRDRIVER.EXE	Card reader driver
CREATE.EXE	File and directory creation utility
CREATEFDL.EXE	CREATE/FDL image
CSP.EXE	Cluster server process image
CTDRIVER.EXE	CTERM driver
CVTNAF.EXE	Convert NETUAF.DAT utility
CVTUAF.EXE	Convert SYSUAF.DAT utility
DBDRIVER.EXE	RP05 and RP06 disk driver
DCL.EXE	Command interpreter
DCLDEF.STB[1]	Global definitions for DCL structures
DDDRIVER.EXE	TU58 driver
DELETE.EXE	File deletion/purge utility
DIFF.EXE	File compare utility
DIRECTORY.EXE	Directory utility
DISKQUOTA.EXE	Disk Quota Utility
DISMOUNT.EXE	Volume dismount utility
DLDRIVER.EXE	RL02 disk driver
DMDRIVER.EXE	RK07 disk driver
DQDRIVER.EXE	RB730 driver
DRDRIVER.EXE	RM03 disk driver
DSRINDEX.EXE	RUNOFF/INDEX image
DSRTOC.EXE	RUNOFF/CONTENTS image
DTR.COM	DTRECV.EXE server initiating procedure

[1]Not supported by DIGITAL

Table A-1 (Cont.)   Files Contained in Directory [SYSEXE]

File Name	Description
DTRECV.EXE	DTSEND server
DTSEND.EXE	DECnet logical links test program
DUDRIVER.EXE	UDA disk driver
DUMP.EXE	Dump utility
DXDRIVER.EXE	RX01 console floppy diskette driver
DYDRIVER.EXE	RX02 floppy diskette driver
DZDRIVER.EXE	DZ11 port driver
EDF.EXE	File definition language editor
EDT.EXE	EDT text editor
ENCRYPFAC.EXE	ENCRYPT command image
ERF.EXE	ANALYZE/ERROR image
ERFBRIEF.EXE	ANALYZE/ERROR brief report generator
ERFBUS.EXE	ANALYZE/ERROR bus display generator
ERFDISK.EXE	ANALYZE/ERROR disk display generator
ERFINICOM.EXE	ANALYZE/ERROR initialize routines
ERFPROC1.EXE	ANALYZE/ERROR processing routines
ERFPROC2.EXE	ANALYZE/ERROR processing routines
ERFPROC3.EXE	ANALYZE/ERROR processing routines
ERFPROC4.EXE	ANALYZE/ERROR processing routines
ERFPROC5.EXE	ANALYZE/ERROR processing routines
ERFRLTIM.EXE	ANALYZE/ERROR real-time device routines
ERFSUMM.EXE	ANALYZE/ERROR summary display routines
ERFTAPE.EXE	ANALYZE/ERROR tape display routines
ERRFMT.EXE	Error logging facility
EVL.EXE	DECnet event logging program
EVL.COM	Command file used by DECnet error logging
EXCHANGE.EXE	RT-11/DOS file transfer utility
F11AACP.EXE	Files-11 Structure Level 1 ancillary control process
F11BXQP.EXE	File system extended QIO processor
FAL.COM	FAL startup procedure
FAL.EXE	DECnet file access listener

Table A–1 (Cont.) Files Contained in Directory [SYSEXE]

File Name	Description
FILESERV.EXE	File system cache flush server
FPEMUL.EXE	Floating point emulation for F- ,D- ,G- and H-floating point
HLD.COM	Command procedure used by HLD.EXE
HLD.EXE	Downline task loading program
IMGDEF.STB[1]	Global definitions for image activator structures
INIT.EXE	Disk device initialization utility
INPSMB.EXE	Card reader input symbiont
INSTALL.EXE	Utility that installs known images
JOBCTL.EXE	Job controller/symbiont manager
LADRIVER.EXE	LPA-11 driver
LALOAD.EXE	Accepts commands from or sends requests to LALOADER to load LPA-11 microcode
LALOADER.EXE	Loads LPA-11 microcode upon power recovery or upon request from LALOAD
LATCP.EXE	LAT-11 control program
LCDRIVER.EXE	DMF-32 lineprinter driver
LIBRARIAN.EXE	Librarian
LINK.EXE	Linker
LOGINOUT.EXE	Login/logout utility
LPDRIVER.EXE	Line printer driver
LTDRIVER.EXE	LAT-11 driver
MACRO32.EXE	VAX MACRO assembler
MAIL.EXE	Mail Utility
MAIL.COM	Command procedure used by DECnet mail
MAILEDIT.COM	Default MAIL editing command procedure
MBXDRIVER.EXE	Shared memory mailbox driver
MERGE.EXE	Merge Utility
MESSAGE.EXE	Message compiler
MIRROR.COM	MIRROR startup procedure
MIRROR.EXE	DECnet node loopback server

[1]Not supported by DIGITAL

Table A-1 (Cont.) Files Contained in Directory [SYSEXE]

File Name	Description
MODPARAMS.DAT	Site modifications to AUTOGEN procedures
MOM.COM	Maintenance operations module DECnet command procedure
MOM.EXE	Maintenance operations module image
MONITOR.EXE	Monitor Utility
MP.EXE	VAX-11/782 multiprocessing code
MP.STB[1]	Symbol table for MP.EXE
MSCP.EXE	MSCP server
MTAAACP.EXE	Magnetic tape ancillary control process
NCP.EXE	Network control program
NDDRIVER.EXE	DECnet pseudo-datalink driver
NETACP.EXE	DECnet ancillary control process
NETDEF.STB[1]	Symbol table for network definition
NETDRIVER.EXE	DECnet logical link driver
NETSERVER.COM	Network server DECnet command procedure
NETSERVER.EXE	Network server image
NICONFIG.COM	Ethernet configurator DECnet command procedure
NICONFIG.EXE	Ethernet configurator image
NML.COM	NML server startup procedure
NML.EXE	DECnet network management listener
NODRIVER.EXE	Asynchronous DECnet driver
NOTICE.TXT	Text file that can contain announcements to system users
OPCCRASH.EXE	System shutdown utility
OPCOM.EXE	Operator communications utility
PADRIVER.EXE	CI780 port driver
PAGEFILE.SYS	System paging file
PARAMS.DAT	Data file for parameter values
PATCH.EXE	Patch Utility
PHONE.COM	PHONE startup procedure

[1]Not supported by DIGITAL

Table A-1 (Cont.) Files Contained in Directory [SYSEXE]

File Name	Description
PHONE.EXE	Phone Utility
PRTSMB.EXE	Print symbiont
PUDRIVER.EXE	CI UDA port driver
QUEMAN.EXE	Queue managing utility
RECLAIM.EXE	CONVERT/RECLAIM image
REMACP.EXE	Remote device ACP
RENAME.EXE	File rename utility
REPLY.EXE	Message broadcasting facility
REQUEST.EXE	Operator request facility
RMS.EXE	Record Management Services
RMS.STB[1]	RMS symbol table
RMSDEF.STB[1]	Global definitions for VAX RMS structures
RTB.EXE	Utility that writes an RT-11 bootstrap on disk
RTPAD.EXE	Remote terminal command interface
RTTDRIVER.EXE	Remote terminal driver
RUNDET.EXE	Facility that runs detached images
RUNOFF.EXE	DIGITAL Standard Runoff text formatting utility
SCSDEF.STB[1]	Symbol table for loadable routines
SCSLOA.EXE	Loadable routines used by SCS
SDA.EXE	System Dump Analyzer
SDLNPARSE.EXE	SDL; Used for installing optional software
SEARCH.EXE	File search utility
SET.EXE	SET command processor
SETP0.EXE	SET MESSAGE command processor
SETSHOACL.EXE	SET and SHOW ACCESS CONTROL LIST commands
SHOW.EXE	SHOW command processor
SHUTDOWN.COM	System shutdown command procedure
SHWCLSTR.EXE	SHOW CLUSTER command

[1]Not supported by DIGITAL

Table A-1 (Cont.) Files Contained in Directory [SYSEXE]

File Name	Description
SMGBLDTRM.EXE	Compiler for TERMTABLE definition file
SMGMAPTRM.EXE	Termtable global section. Run at system startup
SMGTERMS.TXT	ASCII source file for DIGITAL terminal definitions
SORTMERGE.EXE	SORT and MERGE commands
SRTTRN.EXE	SORT specification file translator image
STABACCOP.EXE	Copy program for building standalone BACKUP kit
STABACKUP.EXE	stand-alone BACKUP Utility
STACONFIG.EXE	HSC system disk configurator image
STANDCONF.EXE	stand-alone BACKUP configure image
STARTUP.COM	System startup command procedure
STASYSGEN.EXE	Stand-alone System Generation Utility
STOPREM.EXE	Stop REMACP utility
SUBMIT.EXE	Batch job submission utility
SUMSLP.EXE	Source file editor
SWAPFILE.SYS	System swap file
SYS.EXE	Operating system image file
SYS.MAP	Map of the operating system
SYS.STB	Global symbol table of operating system
SYSBOOT.EXE	System bootstrap utility
SYSDEF.STB[1]	Global definitions for executive structures
SYSDUMP.DMP	System dump file
SYSGEN.EXE	System Generation Utility
SYSINIT.EXE	Operating system initialization image
SYSLOA730.EXE	VAX-11/730 system image file
SYSLOA750.EXE	VAX-11/750 system image file
SYSLOA780.EXE	VAX-11/780 system image file
SYSUAF.DAT	User authorization data file
SYSUAF.RL2	Unmodified copy of SYSUAF.DAT
TECO.EXE	TECO text editor
TERMTABLE.EXE	Binary terminal definitions file

[1]Not supported by DIGITAL

Table A–1 (Cont.) Files Contained in Directory [SYSEXE]

File Name	Description
TERMTABLE.TXT	Terminal definitions source file
TFDRIVER.EXE	TU78 driver
TMDRIVER.EXE	Magnetic tape driver
TSDRIVER.EXE	TS11 Magnetic tape driver
TTDRIVER.EXE	Terminal driver
TTRDVFY.EXE	Auxiliary terminal driver module
TUDRIVER.EXE	Magnetic tape class driver
TYPE.EXE	Type utility
UNLOCK.EXE	File unlock utility
VAXVMSSYS.PAR	System parameter file
VERIFY.EXE	ANALYZE/DISK_STRUCTRURE image
VMB.EXE	VAX/VMS primary bootstrap
VMOUNT.EXE	Volume mount utility
VMSHELP.EXE	Help Utility
WRITEBOOT.EXE	System volume bootblock writing utility
XADRIVER.EXE	Reserved for future use
XDDRIVER.EXE	DECnet DMP-11 datalink driver
XEDRIVER.EXE	UNA driver
XFDRIVER.EXE	DR32 system interconnect interface driver
XFLOADER.EXE	DR32 microcode loader utility
XGDRIVER.EXE	DECnet DMF datalink driver
XMDRIVER.EXE	DMC-11 Synchronous Communications Line Interface driver
XWDRIVER.EXE	DUP-11 device driver
YCDRIVER.EXE	DMF32 asynchronous port driver
YFDRIVER.EXE	DHU port driver

Table A–2   Files Contained in Directory [SYSHLP]

File Name	Help Library
ACLEDT.HLB	Access Control List Editor
ANLRMSHLP.HLB	ANALYZE/RMS_FILE command
DEBUGHLP.HLB	Debugger
DISKQUOTA.HLB	Disk Quota Utility
EDFHLP.HLB	File Definition Language
EDTHELP.HLB	EDT
EDTVT100.DOC	EDT keypad layout for VT100
EDTVT52.DOC	EDT keypad layout for VT52
EXAMPLES.DIR	Examples directory
EXCHNGHLP.HLB	Exchange Utility
HELPLIB.HLB	Default (DCL) help library
INSTALHLP.HLB	Install Utility
LATCP.HLB	LAT Control Program
MAILHELP.HLB	Mail Utility
MNRHELP.HLB	Monitor Utility
NCPHELP.HLB	Network Control Program
PATCHHELP.HLB	Patch Utility
PHONEHELP.HLB	Phone Utility
SDA.HLB	System Dump Analyzer
SHWCLHELP.HLB	Show Cluster Utility
SYSGEN.HLB	System Generation Utility
TECO.HLB	TECO
UAFHELP.HLB	User authorization file
VMSTLRHLP.HLB	Tailoring facility

Table A-3   Files Contained in Directory [SYSHLP.EXAMPLES]

File Name	Description
ADDRIVER.MAR	Example device driver for AD11-K
CONNECT.COM	Command procedure that connects device for LABIO system
DOD_ERAPAT.MAR	Example loadable erase pattern generator
DRCOPY.PRM	Parameter file for DRCOPY routines
DRCOPYBLD.COM	Command procedure to build DRCOPY.EXE
DRMAST.MAR	VAX RMS interface for DRMASTER.FOR
DRMASTER.FOR	Master subroutines for DRCOPY
DRSLAVE.FOR	Slave subroutines for DRCOPY
DRSLV.MAR	VAX RMS interface for DRSLAVE.FOR
DTE_DF03.MAR	SET HOST/DTE dialer support
GBLSECUFO.MAR	Opens file used as a global section for LABIO system
LABCHNDEF.FOR	Defines information associated wth each A/D for LABIO system
LABIO.OPT	Linker options file for linking modules to be used in LABIO
LABIOACQ.FOR	Acquires data for LABIO system
LABIOCIN.MAR	Contains connect-to-interrupt call for LABIO system
LABIOCIN.OPT	Linker options file for linking LABIO_DATA_ACQ
LABIOCOM.FOR	Attaches a LABIO user program to the LABIO system modules of the LABIO system
LABIOCOMP.COM	Command procedure to compile and assemble the modules of the LABIO system
LABIOCON.FOR	Handles user requests and modifies the database for LABIO system
LABIOLINK.COM	Command procedure to link LABIO system
LABIOPEAK.FOR	Samples channel for peak data in LABIO system
LABIOSAMP.FOR	Samples channel in intervals, reporting date, time, and average value on logical device for LABIO system
LABIOSEC.FOR	Places LABIO_SECTION on page boundary
LABIOSTAT.FOR	Displays A/D channel status for LABIO system
LABIOSTRT.COM	Command procedure to start LABIO system
LABMBXDEF.FOR	Defines mailbox block for LABIO system

A-11

Table A–3 (Cont.) Files Contained in Directory [SYSHLP.EXAMPLES]

File Name	Description
LBRDEMO.COM	Command procedure to create Librarian DEMO.EXE
LBRDEMO.FOR	Librarian demo (first part)
LBRMAC.MAR	Librarian demo (second part)
LPATEST.FOR	LPA11-K test program
LPMULT.B32	Example program for line printer
MAILCOMPRESS.COM	Sample procedure to compress mail files
MAILCVT.COM	Sample procedure to convert Version 3.0 mail files
MAILUAF.COM	Sample procedure to manipulate SYS$SYSTEM:VMSMAIL.DAT
MSCPMOUNT.COM	Example cluster disk mount procedure
PEAK.FOR	Peak selection routine in LABIO system
SCRFT.MAR	Optional screen package (SCR$...in RTL) extension to handle foreign terminals
SYSGTTSTR.MSG	Sample SYSGEN TERMINAL/ECHO message file
TDRIVER.MAR	Template for user-written driver
TESTLABIO.FOR	Tests LABIO sytem
USSDISP.MAR	Sample user system service dispatch and service examples
USSLINK.COM	Link command procedure for USSDISP
USSTEST.MAR	Sample program to invoke one of the example user services implemented in USSDISP
USSTSTLNK.COM	Link command procedure for USSTEST
XADRIVER.MAR	DR-11 driver
XALINK.MAR	Sample DR11W to DR11W link program
XAMESSAGE.MAR	DR-11 test program
XATEST.COM	Used to set up XALINK.MAR
XATEST.FOR	Companion program for XAMESSAGE
XIDRIVER.MAR	Example driver for parallel port on DMF32

Table A-4   Files Contained in Directory [SYSLIB]

File Name	Description
ACLEDIT.INI	Access Control List Editor initialization file
BASRTL.EXE	VAX BASIC Run-Time Library
BASRTL2.EXE	VAX BASIC Run-Time Library (part 2)
CDDSHR.EXE	Dummy CDD image for layered products
COBRTL.EXE	VAX COBOL Run-Time Library
CONVSHR.EXE	CONVERT and CONVERT/RECLAIM shareable image
CRFSHR.EXE	Cross-reference shareable image
DBGSSISHR.EXE	VAX DEBUG system service intercept handler
DCLTABLES.EXE	DCL command tables
DCXSHR.EXE	Data compression support
DEBUG.EXE	VAX/VMS Symbolic Debugger
DELTA.EXE	DELTA multimode debugging tool image
DELTA.OBJ	Alternate debugging tool
DISMNTSHR.EXE	DISMOUNT shareable image
DTE_DF03.EXE	SET HOST/DTE support for DF03 dialer
EDTSHR.EXE	EDT editor
ENCRYPSHR.EXE	Dummy VAX Encryption support module
ERFCOMMON.EXE	ANALYZE/ERROR common data structures
ERFLIB.TLB	ANALYZE/ERROR device descriptions
ERFSHR.EXE	ANALYZE/ERROR common routines
FDLSHR.EXE	FDL parsing shareable image
FORDEF.FOR	FORTRAN INCLUDE file: FOR$ symbols
FORIOSDEF.FOR	FORTRAN INCLUDE file: IOSTAT error codes
FORRTL.EXE	VAX FORTRAN Run-Time Library
IMAGELIB.OLB	System default shareable image library
IMGDMP.EXE	Image dump procedures
LBRSHR.EXE	Librarian shareable image
LIB.MLB	Operating system macro library
LIB.REQ	Structure definitions of executive internals for use by BLISS programs
LIBDEF.FOR	FORTRAN program utility INCLUDE files

Table A–4 (Cont.)   Files Contained in Directory [SYSLIB]

File Name	Description
LIBRTL.EXE	VAX Common Run-Time Library
LIBRTL2.EXE	VAX Common Run-Time Library (part 2)
MCRTABLES.EXE	MCR command tables
MOUNTSHR.EXE	MOUNT shareable image
MTHDEF.FOR	FORTRAN INCLUDE files: MATH$ symbols
MTHRTL.EXE	VAX math support Run-Time Library
NMLSHR.EXE	DECnet management listener shareable image
PASRTL.EXE	VAX Pascal Run-Time Library
PLIRTL.EXE	VAX PL/I Run-Time Library
RPGRTL.EXE	VAX RPG Run-Time Library
SCRSHR.EXE	RTL terminal screen procedures shareable image
SECURESHR.EXE	Rights database (RIGHTSLIST.DAT) service routines
SIGDEF.FOR	FORTRAN program utility INCLUDE files
SMBSRVSHR.EXE	Print symbiont service routines
SMGSHR.EXE	VAX Screen Management Run-Time Library
SORTSHR.EXE	VAX Sort/Merge Run-Time Library
STARLET.MLB	System macro library
STARLET.OLB	System object library and Run-Time Library
STARLET.REQ	User interface structures for use by BLISS programs
STARLETSD.TLB	Text library of STARLET definitions; Used during layered product installations.
SUMSHR.EXE	Source update merge shareable image
TPAMAC.REQ	Structure definitions for BLISS programs using TPARSE
TRACE.EXE	VAX/VMS error traceback facility
VMSRTL.EXE	Run-Time Library shareable image
XFDEF.FOR	Definitions available for programs using DR780 support routines

Table A–5   Files Contained in Directory [SYSMGR]

File Name	Description
ACCOUNTNG.DAT	Accounting data file
ALFMAINT.COM	Command procedure to maintain SYS$SYSTEM:SYSALF.DAT
LOADNET.COM	Command procedure to create network ACP process
LPA11STRT.COM	LPA11 site-specific startup command procedure
LTLOAD.COM	Command procedure to load and start LAT
MAKEROOT.COM	Command procedure to add new roots to cluster common system disk.
NETCONFIG.COM	Command procedure to configure network database.
RTTLOAD.COM	Remote terminal loader
STARTNET.COM	DECnet startup procedure
SYSHUTDWN.COM	Site-specific system shutdown command procedure
SYSTARTUP.COM	Site-specific system startup command procedure
VMSIMAGES.COM	Invoked at startup to install known images
VMSIMAGES.DAT	Data file for VMSIMAGES.COM

Table A–6   Files Contained in Directory [SYSMSG]

File Name	Description
CLIUTLMSG.EXE	ANALYZE/MEDIA,EXCHANGE,MAIL,PHONE,PRINT,SUBMIT, RUN,SET,SHOW,SEARCH
DBGTBKMSG.EXE	DEBUG, TRACE
FILMNTMSG.EXE	ANALYZE/OBJECT,ANALYZE/IMAGE,EDIT/FDL,ANALYZE /DISK
NETWRKMSG.EXE	NCP, SET HOST
PASMSG.EXE	PASCAL Run-Time Library
PLIMSG.EXE	PL/I Run-Time Library
PRGDEVMSG.EXE	CDU,DIFF,DUMP,LIBRARY,LINK, MACRO,MESSAGE,PATCH,ANALYZE/SYSTEM,ANALYZE /CRASH
RPGMSG.EXE	RPG Run-Time Library
SHRIMGMSG.EXE	CONVSHR, DCXSHR, FDLSHR, SORTSHR, SMGSHR, EDTSHR

Table A–6 (Cont.)   Files Contained in Directory [SYSMSG]

File Name	Description
SYSMGTMSG.EXE	ACC,EDIT/ACL,BACKUP,INSTALL,MONITOR,AUTHORIZE
SYSMSG.EXE	System message file

Table A–7   Files Contained in Directory [SYSTEST]

File Name	Description
TCNTRL.CLD	Defines commands to invoke the UETP Test Controller
UETCLIG00.COM	Command procedure to run the cluster-integration phase
UETCLIG00.DAT	Used by UETP Test Controller to start cluster-integration phase
UETCOMS00.EXE	DMC and DMR device test
UETDISK00.EXE	Disk device test
UETDMPF00.EXE	DMP and DMF-32 device test
UETDNET00.COM	Command procedure for the DECnet phase
UETDNET00.DAT	Used by UETP Test Controller to start DECnet phase
UETDR1W00.EXE	DR11–W device test
UETDR7800.EXE	DR780 and DR750 device test
UETFORT01.DAT	FORTRAN data file used by UETFORT01
UETFORT01.EXE	Artificial load in load test
UETFORT02.EXE	Artificial load in load test
UETFORT03.EXE	Artificial load in load test
UETINIT00.EXE	Checks UETP environment and sets overall parameters
UETINIT01.EXE	Quick checks devices and build UETINIDEV.DAT
UETLOAD00.DAT	Used by UETP Test Controller to start Load Test phase
UETLOAD02.COM	User script for load test
UETLOAD03.COM	User script for load test
UETLOAD04.COM	User script for load test
UETLOAD05.COM	User script for load test
UETLOAD06.COM	User script for load test
UETLOAD07.COM	User script for load test
UETLOAD08.COM	User script for load test

Table A–7 (Cont.) Files Contained in Directory [SYSTEST]

File Name	Description
UETLOAD09.COM	User script for load test
UETLOAD10.COM	User script for load test
UETLOAD11.COM	User script for load test
UETLPAK00.EXE	LPA11-K device test
UETMA7800.EXE	MA780 device test
UETMEMY01.EXE	Artificial load for load test
UETNETS00.EXE	Reports nonzero counters as part of DECnet phase
UETP.COM	Main command procedure for entire UETP
UETPHAS00.EXE	Test controller
UETRSXFOR.EXE	Artificial load for load test
UETSUPDEV.DAT	Supported device data file
UETTAPE00.EXE	Magnetic tape device test
UETTTYS00.EXE	Terminal and line printer device test
UETUNAS00.EXE	DEUNA device test

Table A–8 Files Contained in Directory [SYSUPD]

File Name	Description
AUTOGEN.COM	Command procedure to determine parameter values
BLISSREQ.TLR	List of files in the BLISS tailoring group
BOOTBLDR.COM	Multiprocessing console floppies command procedure
BOOTUPD.COM	Command procedure to update VAX/VMS bootstrap file on console floppy diskette
CONSCOPY.COM	Command procedure that copies console floppy diskette
CVTNAF.COM	Command procedure to convert NETUAF.DAT
CVTUAF.COM	Command procedure to convert SYSUAF.DAT
DECNET.TLR	List of files in the DECnet tailoring group
DEVELOP.TLR	List of files in the DEVELOP tailoring group
DXCOPY.COM	Command procedure that copies files from console floppy diskette and restores files to floppy diskette
EXAMPLES.TLR	List of files in the EXAMPLE tailoring group

Table A-8 (Cont.) Files Contained in Directory [SYSUPD]

File Name	Description
FILETOOLS.TLR	List of files in the TOOLS tailoring group
HELP.TLR	List of files in the HELP tailoring group
LIBDECOMP.COM	Command procedure to expand libraries shipped in data-reduced format.
LIBRARY.TLR	List of files in the LIBRARY tailoring group
MANAGER.TLR	List of files in the MANAGER tailoring group
MISCTOOLS.TLR	List of files in the MISCTOOLS tailoring group
QUEUES.TLR	List of files in the QUEUES tailoring group
REQUIRED.TLR	List of files in the REQUIRED tailoring group
SETDEFBOO.COM	Command procedure that sets default boot command file
SPKITBLD.COM	Command procedure to build software product kits.
STABACKIT.COM	Command procedure that builds stand-alone BACKUP to media
SWAPFILES.COM	Command procedure that creates swapping, paging,and system dump files of appropriate size for system being installed
TEXTTOOLS.TLR	List of files in the TEXTTOOLS tailoring group
UETP.TLR	List of files in the UETP tailoring group
VMSINSTAL.COM	Command procedure to install EDTCAI and maintenance updates
VMSKITBLD.COM	Command procedure that builds and copies VAX/VMS distribution disk
VMSKITBLD.DAT	List of files in VAX/VMS system that drives VMSKITBLD.COM
VMSTAILOR.COM	Tailoring facility command procedure (supported only on VAX-11/730)
VMSUPDATE.COM	System update command procedure

B   Restoring the VAX/VMS Source Kit

This material is included for users who purchase the source license. It is limited to telling you how to invoke the command procedure that directs the building of the source kit and references to appropriate supporting documentation.

The VAX/VMS source kit is shipped on a set of magnetic tapes that contain the source files, object files, command files, language processors, and utilities you need to build any standard VAX/VMS component. The kit may be restored to any system that has sufficient storage capabilities.

Note: **The languages and utilities used to build operating system components are not supported by DIGITAL.**

Before you restore the source kit to a disk storage system, be sure you have enough free disk space. A single component may as use as much as 40,000 free blocks. To restore the entire kit, you will need a disk drive with a storage capacity of at least 110,000 blocks. When doing a complete system build, you will need a result disk device with a storage capacity of approximately 900,000 blocks.

To restore the source kit, you must first mount the tape labeled VMSRC1 and restore the file [SYSBLDCOM]SRCINSTAL.COM. When you invoke SRCINSTAL.COM, it asks you for information it needs to define the system building environment.

1 Allocate a tape drive with the following command:

    $ALLOCATE ddcu:

Note: **See Chapter 4 for help with device names.**

2 Load the tape into the allocated drive.

3 Mount the tape as foreign with this command:

    $MOUNT/FOREIGN ddcu: VMSRC1

4 Restore the source installation command procedure with this command:

```
$BACKUP ddcu:SYSBLDCOM/SELECT=(SRCINSTAL.COM) ddcu:[...]
```

5 After you restore the file, dismount the tape.

6 Invoke the source installation command procedure by entering:

```
$@ddcu:[SYSBLDCOM]SRCINSTAL.COM
```

The command procedure creates a system build account containing the logical name definitions and private command definitions needed to set up the proper environment for a system build. To log in to this account, specify the username SYSTEMBUILD and the password CAVEAT_EMPTOR.

To obtain a general description of the source kit, you can print a copy of the file [SYSBLDCOM]SOURCEKIT.DOC.

C Using VMSUPDATE

This appendix outlines the procedure for installing optional products using VMSUPDATE.

Note: **VMSUPDATE is *not* supported on tailored systems nor on cluster configurations that share a common system disk.**

The procedure for installing optional software products with VMSUPDATE requires little involvement on your part beyond setting up the proper conditions for the installation and responding to queries and prompting messages displayed as the installation proceeds.

When you invoke VMSUPDATE it uses command procedures contained in the distribution media to direct the installation. These command procedures direct the installation by means of queries and instructions sent to the terminal. Most queries require only that you provide a YES or NO response.

C.1 Preparing to Install Optional Software

Optional software products are distributed on one or more pieces of console media depending on the product(s). Each piece of media is labeled with a unique name and serial number that differentiates it from others in the distribution kit. Check that your kit contains all the console media listed in the bill of materials.

To prepare for the installation, do the following:

1 Back up your system disk (see Chapter 4). Because optional software products may delete older product versions before installing new ones, it is advisable to back up your system disk before attempting any software installation. It is also advisable because a system failure at a critical point of the installation could leave unusable files.

2 Log in at the console terminal under the system manager's account, SYSTEM.

Note: If the SYSGEN parameters MOUNTMSG or DISMOUMSG are set to 1, you will receive a message from OPCOM each time a disk, tape, or piece of console media is mounted or dismounted. These messages are normally disabled, but if they have been activated and you are installing from a console terminal, they will appear from 1 to 30 seconds after each mount or dismount.

3 Prevent users from gaining access to the system by executing the SHUTDOWN.COM command procedure. Then reboot the system.

4 Set the login quota to 0.

```
$ SET LOGINS/INTERACTIVE = 0
```

5 Be sure that you have set your default to the disk that is to receive the product. This is typically the system disk with the logical name SYS$SYSROOT.

If you are installing optional software on a system disk other than the disk from which you booted, you must define a logical name for the target disk. For example, if you are installing software on an RK07 disk on the first drive, issue the following commands:

```
$ DEFINE TARGET DMA1:[SYS0.]
$ SET DEFAULT TARGET:[SYSUPD]
```

Note: Check the documentation for your product to be sure it can be installed on a device other than the system disk.

6 If you are installing to the current system disk, establish the following default:

```
$ SET DEFAULT SYS$UPDATE
```

C.2 Performing the Installation

Enter the following command to begin the installation:

```
$ @VMSUPDATE
```

You will receive the following message text at your terminal:

```
VMS Update Procedure
This command procedure performs VAX/VMS software updates and optional
software installations for VAX/VMS Release 3. During this sequence, the
standard console medium will not be present in the console drive.
Therefore, the system may be vulnerable to a power failure or other fatal
crash. If a system crash should occur during this period the update
sequence can be restarted at the beginning of the first incomplete
update.
Dismount the current console medium.
Please place the first floppy in the console drive
```

> **Note:** **The examples used here relate to the use of floppy diskettes in console drive CSA1 on a VAX–11/780 system. If your system is not a VAX–11/780, the console media is TU58 cartridges and the console drive is CSA1. See Chapter 2 for help in locating the console drives.**

After placing the first piece of media in the CSA1 drive, the installation procedure asks:

```
Are you ready to continue?:
```

If you enter the letter **Y**, the installation proceeds.

If you type N, the request to insert the first piece of media will be repeated together with the question "Are you ready to continue?" .

When the installation resumes, the following message is displayed:

```
%MOUNT-I-MOUNTED , <label> mounted on _CSA1:
```

The remaining questions and prompts depend on the product you are installing.

(

C.3 Completing the Installation

When the installation is completed, control is returned to
VMSUPDATE.COM, which asks:

Are there more kits to process?:

If you enter the letter **Y**, the installation procedure resumes
with the following request:

Please place the first floppy in the console drive.

If you have no more products to install, enter the letter N.
VMSUPDATE responds by sending the you the following
message and then returning you to command level:

Requested update sequence complete.

To use the new product(s), you will have to shut down the
processor and reboot the system.

D Operating System Distribution Kits

This appendix lists the software components of the distribution kits for the various VAX processors. Each kit also contains appropriate supporting documentation.

D.1 VAX-11/780 Distribution Kit

VAX/VMS software for the VAX-11/780 is distributed in one of three media:

- Magnetic tape

- RK07 disk

- RA60 disk

In addition to the media and other materials in the software distribution kit, you will need a console floppy before you can install or upgrade a VAX/VMS system. If you are installing a new system, the console floppy was shipped with the hardware. If you are upgrading an existing system, your current console floppy will be upgraded as part of the upgrade procedure.

D.1.1 Magnetic Tape Kit

The VAX-11/780 magnetic tape software distribution kit contains the following materials:

- The VAX/VMS Version 4.0 distribution tape

  Part number: BB-BT05A-BE
  Part description: VAX/VMS V4.0 BIN 16MT9

- Three floppy diskettes containing stand-alone BACKUP. The asterisk (*) denotes the latest version.

  Part number: AS-CT94*-BE
  Part description: VAX/VMS V4.0 S/A BKUP RX1 1/3
  Part number: AS-CT95*-BE
  Part description: VAX/VMS V4.0 S/A BKUP RX1 2/3

(

Part number: AS-CT96∗-BE
Part description: VAX/VMS V4.0 S/A BKUP RX1 3/3

D.1.2 RK07 Kit

The VAX–11/780 RK07 software distribution kit contains the
following materials:

- The RK07 Version 4.0 distribution disk

    Part number: AY-BT10A-BE
    Part description: VAX/VMS V4.0 BIN RK7

D.1.3 RA60 Kit

The VAX–11/780 RA60 software distribution kit contains the
following:

- The RA60 Version 4.0 distribution disk

    Part number: BD-BT09A-BE
    Part description: VAX/VMS V4.0 BIN RA6

D.2 VAX–11/750 Distribution Kit

VAX/VMS software for the VAX–11/750 is distributed in one
of four kits:

- Magnetic tape
- RK07 disk
- RA60 disk
- RL02 disks

The contents of each kit are listed in a bill of materials that
accompanies each kit.

In addition to the materials each kit, you will need a console
TU58 cartridge before you can proceed with an installation or
an upgrade. If you are installing a newly purchased system, the
console TU58 cartridge was shipped with the hardware. If you
are upgrading an existing system, your current console TU58

cartridge will be upgraded as part of the VAX/VMS upgrade procedure.

D.2.1   Magnetic Tape Kit

The VAX–11/750 magnetic tape software distribution kit contains the following materials:

- The VAX/VMS Version 4.0 distribution tape

    Part number: BB-BT05A-BE
    Part description: VAX/VMS V4.0 BIN 16MT9

- Three TU58 cartridges that contain stand-alone BACKUP for installing your system. The asterisk (∗) denotes latest version.

    Part number: BE-CT97∗-BE
    Part description: VAX/VMS V4.0 S/A BKUP T58 1/3
    Part number: BE-CT98∗-BE
    Part description: VAX/VMS V4.0 S/A BKUP T58 2/3
    Part number: BE-CT99∗-BE
    Part description: VAX/VMS V4.0 S/A BKUP T58 3/3

D.2.2   RK07 Kit

The VAX–11/750 RK07 software distribution kit contains the following materials:

- The RK07 Version 4.0 distribution disk

    Part number: AY-BT10A-BE
    Part description: VAX/VMS V4.0 BIN RK7

D.2.3   RA60 Kit

The VAX–11/750 RA60 software distribution kit contains the following materials:

- The RA60 Version 4.0 distribution disk

    Part number: BD-BT09A-BE
    Part description: VAX/VMS V4.0 BIN RA6

D.2.4 RL02 Kit

If you have an an RL02/RA80 system configuration, the
operating system is distributed on an RL02 kit that include
two disks.

The first disk contains two save sets: a *required* save set and an
optional files save set. The *required* save set includes all the files
you need to build a bootable system. The *optional files* save set
includes additional pieces of the VAX/VMS operating system;
VAX BLISS files, programming examples and the system map.

The second distribution disk contains the library save set that
includes the rest of the operating system files.

Index

D

O

OLDUETP.LOG • 7-22
OPCOM message • 7-26
    during product installation • 5-2, C-2
Open file quota (FILLM) • 5-2
Operating instructions
    summary • 7-2
Operating modes
    console subsystem • 2-1
OPTIONS keyword • 5-5
Output • 7-1
    console report • 7-14, 7-17, 7-36,
      7-37
    interpreting • 7-21
    terminal and line printer • 7-39
    UETP.LOG • 7-17
Overview of installation • 1-1

P

Page file size
    use in making alternate root • 6-23
Password • 7-4
Peripheral devices
    list of • 3-1
Phase
    cluster • 7-18
    DECnet • 7-18, 7-23, 7-32, 7-41,
      7-42
    device • 7-18, 7-36
    initialization • 7-35
    load • 7-18, 7-41
    running individual • 7-18
Phase controller
    See UETPHAS00.EXE
POWER ON ACTION switch • 2-7
Preparing a tailored system for UETP •
    7-12
Preparing devices • 7-5
    disk • 7-2, 7-6
    line printer • 7-3, 7-8
    magnetic tape • 7-3, 7-7
    summary • 7-2
    terminal • 7-3, 7-8
    VAXcluster • 7-11
Privilege • 7-4, 7-23
    error message • 7-24

Privilege (cont'd.)
    for running UETP • 7-24
Processor control panel
    VAX–11/730,VAX–11/725 • 2-9
    VAX–11/750 • 2-5
    VAX–11/780 • 2-2
Product installation
    preparation • 5-2
Product list
    VMSINSTAL parameter • 5-4
      syntax • 5-5
Program mode
    console subsystem • 2-1

Q

Question mark (?)
    to get help from VMSINSTAL • 5-6
Queues
    during upgrade • 6-6
Quota • 7-23
    error message • 7-24
    for running UETP • 7-24

R

R80, RM80
    controls and indicators • 3-13
    disk drive • 3-13
R80/RL02
    backup • 4-20
RA60, RA80, RA81
    controls and indicators • 3-15
    disk drive • 3-15
RC25 backup
    fixed system disk • 4-21
    removable system disk • 4-22
RC25 controls and indicators • 3-17
Rebooting
    from BOOT58 level • 6-16
    to restart upgrade • 6-12, 6-16
Removing files after upgrade
    instructions for • 6-21
RESET switch • 2-9
Restart
    on VAX–11/730 • 6-12
    switch positions • 6-5

S

W

READER'S COMMENTS

Note: This form is for document comments only. DIGITAL will use comments submitted on this form at the company's discretion. If you require a written reply and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Did you find this manual understandable, usable, and well organized? Please make suggestions for improvement.

_____

_____

_____

_____

_____

_____

Did you find errors in this manual? If so, specify the error and the page number.

_____

_____

_____

_____

_____

_____

Please indicate the type of user/reader that you most nearly represent:

☐ Assembly language programmer
☐ Higher-level language programmer
☐ Occasional programmer (experienced)
☐ User with little programming experience
☐ Student programmer
☐ Other (please specify) _____

Name _____ Date _____

Organization _____

Street _____

City _____ State _____ Zip Code _____
                                              or Country