August 1978

This guide explains the function of the VAX/VMS User Environment Test Package (UETP) and provides operating instructions for running the UETP on a VAX/VMS system.

VAX/VMS
UETP User's Guide

Order No. AA-D643A-TE

SUPERSESSION/UPDATE INFORMATION: This is a new document for this release.

OPERATING SYSTEM AND VERSION: VAX/VMS V01

SOFTWARE VERSION: VAX/VMS V01

To order additional copies of this document, contact the Software Distribution Center, Digital Equipment Corporation, Maynard, Massachusetts 01754

digital equipment corporation • maynard. massachusetts

First Printing, August 1978

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

The following are trademarks of Digital Equipment Corporation:

DIGITAL	DECsystem-10	MASSBUS
DEC	DECtape	OMNIBUS
PDP	DIBOL	OS/8
DECUS	EDUSYSTEM	PHA
UNIBUS	FLIP CHIP	RSTS
COMPUTER LABS	FOCAL	RSX
COMTEX	INDAC	TYPESET-8
DDT	LAB-8	TYPESET-11
DECCOMM	DECSYSTEM-20	TMS-11
ASSIST-11	RTS-8	ITPS-10
VAX	VMS	SBI
DECnet	IAS	

CONTENTS

CONTENTS (Cont.)

FIGURES

TABLES

PREFACE

MANUAL OBJECTIVES

The purpose of the VAX/VMS UETP User's Guide is to explain the function of the User Environment Test Package and to provide complete UETP operating instructions.

INTENDED AUDIENCE

The major audiences for the VAX/VMS UETP User's Guide are manufacturing technicians and DIGITAL field service and software support personnel. The guide tells them how to run the test package and how to interpret test results. The guide also allows customers to interpret what is happening while the UETP executes. In addition, the customers themselves may choose to run the UETP by following instructions contained in the guide.

STRUCTURE OF THIS DOCUMENT

This guide contains three chapters and one appendix.

- Chapter 1 is an introduction to the UETP that discusses the role of the UETP, the way it works, and the output it produces.
- Chapter 2 provides detailed operating instructions for running the UETP package automatically and for running individual UETP tests separately.
- Chapter 3 explains all the messages that the UETP tests can return.
- Appendix A is a summary description of operating instructions explained in detail in Chapter 2.

ASSOCIATED DOCUMENTS

This guide refers to several other VAX-11 documents (the VAX/VMS System Messages and Recovery Procedures Manual, in particular). For information on these and other related documents, refer to the VAX-11 Information Directory. The directory briefly describes all the VAX-11 documents and explains each document's intended audience.

CONVENTIONS USED IN THIS DOCUMENT

In examples of dialogues with the system, user input is printed in red to distinguish it from text displayed by the system. For example:

```
Username: SYSTEST
```

When demonstrating the format of UETP calls, the guide encloses optional input in brackets ([]). For example:

```
$ @UETP [/OUTPUT=filespec]
```

Note, however, that brackets are required syntax in directory specifications, such as [SYSTEST].

CHAPTER 1

INTRODUCTION TO THE UETP

This chapter is an introduction to the VAX/VMS User Environment Test Package (UETP). The UETP is a collection of tests designed to demonstrate that the hardware and software components of a VAX/VMS system are in working order. DIGITAL software support representatives run the UETP on a newly installed VAX/VMS system as the formal sample procedure. In three sections, the chapter discusses the following topics:

- The role of the UETP (Section 1.1)
- The way in which the UETP works (Section 1.2)
- The output from the UETP tests (Section 1.3)

The UETP tests are simple to run. To run all of them at once, you first type a command to invoke the UETP and then respond to several prompts. Using the information you have provided, the tests proceed automatically. Alternatively, you can choose to run only one test at a time; you can select an individual test by invoking a specific image or command procedure. Chapter 2 of this guide gives complete instructions for running the collection of tests automatically and for running each test on its own.

1.1 THE ROLE OF THE UETP

The UETP leads the system through a series of exercises; at the end of the series, most hardware and software components have been requested to perform one or more tasks. The tests show not only that individual components work but also that the various components work together as an integrated system.

The UETP attempts to simulate a normal timesharing environment; that is, the tests make demands on the system that are similar to demands that might originate from everyday use. The UETP does not attempt to test every feature exhaustively. When the UETP runs to completion without encountering nonrecoverable errors, the system under test is ready for normal use.

1.1.1 Who Uses the UETP

The primary users of the UETP are VAX/VMS manufacturing technicians and DIGITAL field service and software support representatives. The UETP is an important part of the final assembly and test (FA&T) of every VAX/VMS system. If you are a manufacturing technician, you run the UETP many times on each system to ensure that it works properly.

The UETP is the last of a thorough regimen of tests; before a system can be shipped to a customer, it must successfully complete all diagnostic tests as well as the UETP.

If you are a field service or software support representative, you run the UETP after installing a VAX/VMS system at the customer site. By running the UETP, you can ensure that the system was not damaged in transit and that the hardware and software have been correctly installed and generated. Furthermore, in its role as the sample procedure, the UETP demonstrates to the customer that the newly installed system is capable of doing useful work.

1.1.2 What the UETP Tests

With one exception, the UETP tests exercise devices and functions that are common to all VAX/VMS systems. The package does not test optional features such as high-level language compilers (other than the VAX-11 FORTRAN IV-PLUS compiler) or network devices. The system components tested include:

- All standard line printers, terminals, magnetic tapes and disks connected to the system
- Most RSX-11M utilities that run in compatibility mode on VAX/VMS
- Various native mode functions such as system services, record management services (VAX-11 RMS), native mode utilities, and, optionally, the VAX-11 FORTRAN IV-PLUS compiler
- The system's multiuser capability

The UETP explicitly tests the features listed above; moreover, in the process of running its tests, the UETP indirectly tests additional functions of the system. For example, the UETP uses command procedures to execute some of its tests. The tests themselves therefore show how command procedures can be used for indirect job processing. In addition, various system commands are issued throughout the tests. In particular, the system load test, which demonstrates the system's multiuser capability, issues many commands within a short time.

1.1.3 The UETP's Relationship to Error Logging and Diagnostics

When the UETP encounters an error, it reacts either by returning an error message and proceeding or by reporting a fatal error and terminating. In either case, the UETP does not attempt to discover the cause of the error; in other words, the tests react as normal user programs do.

If the UETP encounters an error that you cannot immediately 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 SYE error report generator, you can obtain a detailed report of hardware and system errors that occurred while the UETP was running. SYE reports optionally provide detailed information on the state of the system at the time each error occurred.

The diagnostic facilities can help in a different way. These tests exhaustively examine a device or medium to isolate the sources of any errors. If an error report covering a UETP run indicates that a particular device is consistently generating errors, you can run the appropriate diagnostic to investigate the device.

The VAX/VMS Operator's Guide contains operating instructions for the SYE report generator.

1.2 HOW THE UETP WORKS

You can run the UETP tests in either one of two ways:

- Altogether by invoking a master command procedure
- Individually by invoking a specific UETP test

The first method runs all the tests automatically. A master command procedure contains commands that initiate each test phase in turn. When you invoke the procedure, it begins by asking several questions; your responses provide information needed by various test phases. After you answer the questions, the tests run to completion without further input from you.

At certain times, you may prefer to run only one test phase. The modular design of the UETP permits you to run each test phase individually. Furthermore, some phases consist of several tests which can also be run independently. For example, the device test phase runs a different test for each type of device. You can test either all the devices at once or only the devices associated with a specific controller.

Chapter 2 explains both how to run the whole package of tests at once and how to run each test individually. The remainder of this section explains the structure of the test package and the function of each test phase.

1.2.1 The Master Command Procedure

The file UETP.COM is the master command procedure that you invoke to run the whole package of tests. Figure 1-1 below shows the principal commands within UETP.COM; these are the commands that actually invoke the various test phases. Additional commands within the file perform such tasks as asking for input from the console, defining logical names, and manipulating files generated by the tests. The additional commands are essential to making the UETP package as automated a procedure as possible.

As the figure shows, the procedure invokes each test phase either by running an image or by invoking another command procedure.

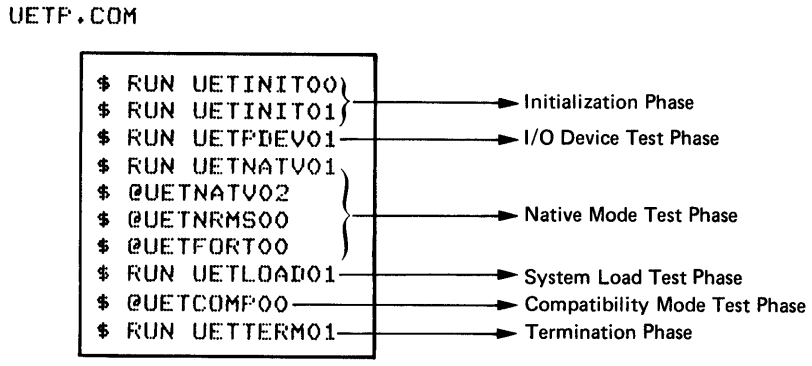

Figure 1-1 The UETP Master Command Procedure

1.2.2 The Initialization Phase

The initialization phase prepares for the actual tests in the following ways:

- Commands within the master command file prompt for input from the terminal. (Section 2.2 explains the prompts in detail.) The input defines certain variables that affect the execution of the UETP tests.

- The image UETINIT00.EXE gathers information on all the controllers in the system and on their associated devices. The image writes the information into a file called UETINIDEV.DAT and then displays the file at the terminal.

- Using the information in the file UETINIDEV.DAT, the second initialization image, UETINIT01.EXE, verifies that all the devices in the system are operable by performing a simple read/write operation. If a device fails this test, the device's entry in UETINIDEV.DAT specifies that the device is unavailable. As a result, subsequent UETP tests ignore that device.

1.2.3 The I/O Device Test Phase

The I/O device test phase includes separate tests for each type of device to be tested (disks, line printers, magnetic tapes and terminals). For each controller in the system, the image UETPDEV01.EXE creates a separate detached process and passes to it a controller name. (Section 2.4.1 explains how the controller name is passed.) Each detached process runs the appropriate device test to examine all the controller's associated devices. For example, if a system has three terminal controllers, one line printer controller and two disk controllers, the image creates six detached processes: three processes to test terminals, one to test line printers, and two to test disks.

By creating detached processes to run the individual tests, this test phase is able to exercise many devices simultaneously; all the detached processes execute concurrently. The image UETPDEV01.EXE deletes each process when the assigned controller's devices have all been tested.

The terminal tests and line printer tests perform similar exercises. Both generate 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's name, the device's name, the date, the time, and a test page number.

On each disk in the system, the test allocates a large file into which it randomly writes blocks of data. The test then checks the written data for errors and erases the file.

The magnetic tape test exercises all the tape drives in the system. The test creates a large file on each mounted tape, into which it writes multiple sequential records of varying sizes. After writing the records, the test rewinds the tape, validates the written records and erases the file.

1.2.4 The Native Mode Test Phase

The native mode test phase exercises software services provided explicitly for VAX/VMS (in contrast to RSX-11M utilities that run in compatibility mode). This phase includes four separate tests:

- UETNATV01.EXE
- UETNATV02.COM
- UETNRMS00.COM
- UETFORT00.COM

The test initiated by UETNATV01.EXE issues all the system services available to VAX/VMS programmers. For each service, the test produces both success and failure return codes. For the success codes, the test verifies that the service worked correctly. Native mode utilities such as the VAX-11 Symbolic Debugger and the Image File Patch utility, are tested by the command procedure UETNATV02.COM. The test initiated by the command procedure UETNRMS00.COM handles all the record management services (VAX-11 RMS), which are used in programs to perform I/O. The final part of the native mode test phase runs only if the system being tested includes a VAX-11 FORTRAN IV-PLUS compiler. The command procedure UETFORT00.COM tests various features of the VAX-11 FORTRAN IV-PLUS compiler and the object code it produces.

1.2.5 The System Load Test Phase

The system load test, directed by the image UETLOAD01.EXE, creates a number of detached processes, which all execute a command procedure. (When you initiate the UETP, you specify the number of detached processes to be created; the number depends on the amount of memory in your system. See Section 2.2.2.) Each process simulates a user logged in at a terminal; the commands within each procedure are the same type of commands that a user can enter 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 real users concurrently issuing commands from terminals. In this way, the load test creates an environment that is similar to normal system use.

1.2.6 The Compatibility Mode Test Phase

The compatibility mode test issues commands that call most RSX-11M utilities running in compatibility mode on VAX/VMS. Compatibility mode utilities run on VAX/VMS with the aid of the Application Migration Executive (AME). The command procedure UETCOMP00.COM issues several commands for each utility and then, in most cases, compares the output from the commands with output that is known to be correct. (Section 2.4.4 lists all the utilities that this phase tests.)

1.2.7 The Termination Phase

The termination phase signals the end of the UETP package. The master command procedure deletes temporary files and performs other cleanup activity. The image UETTERM01.EXE then displays the time at which the UETP run ended.

In addition, the master command procedure UETP.COM determines whether the UETP needs to be restarted. (You can request multiple runs when you start up the test package; see Section 2.2.3.) To automatically start up another run, the procedure passes control directly to the device test phase.

1.3 INTERPRETING UETP OUTPUT

You can monitor the progress of the UETP tests at the terminal from which you invoked them. The terminal always displays status information, such as messages that announce the beginning and ending of each test 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. For example, one log file contains the command output produced by the load test; another contains output created by the compatibility mode utilities that are tested. (Section 2.3.2 contains a list of the log files.) If the tests complete successfully, you need not refer to the log files; the terminal log provides sufficient information. However, when a test encounters an error, the log files provide further information about the nature of the error.

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

Chapter 3 of this guide lists and explains all the messages that the tests can generate. However, to clarify messages sent by the tested system components, you need to refer either to the VAX/VMS System Messages and Recovery Procedures Manual or to the manual that describes the individual component.

CHAPTER 2

UETP OPERATING INSTRUCTIONS

This chapter presents the UETP operating instructions, including:

- How to prepare the system for running the UETP (Section 2.1)
- How to define UETP variables (Section 2.2)
- How to run the whole package (Section 2.3)
- How to run individual UETP test phases (Section 2.4)

2.1 PREPARING THE SYSTEM

The images and command procedures that comprise the UETP are included in the distributed VAX/VMS system. You can run the UETP any time after the system has been generated and booted. This section describes the steps needed to set the system up for running the UETP.

2.1.1 Booting the System

To prepare for the UETP tests, boot the system in the normal manner. Bootstrap instructions are contained in the VAX-11 Software Installation Guide.

2.1.2 Logging In

You can log in from any terminal connected to the system. If you have a choice between a hard-copy terminal and a VDT terminal, consider whether you want a copy of UETP output to the console. Section 2.2.1 below discusses output to the terminal in conjunction with the UETP log files.

All UETP files reside in the [SYSTEST] directory on the system disk. To access these files, you must log in under the user name SYSTEST. The password to this account in the distributed system is UETP. Note that because the SYSTEST account has powerful privileges, its password should be changed after the VAX/VMS system has been installed at the customer site. The AUTHORIZE utility, described in the VAX/VMS System Manager's Guide, can be used to change account passwords.

For example:

```
<RET>
Username: SYSTEST <RET>
Password:          <RET>
```

where the password is UETP or a password determined by the customer site. (The system never echoes a string typed in response to the Password: prompt.)

Note that the SYSTEST account must have the following privileges:

```
CMKRNL    PRMMBX
DETACH    PRMCEB
GRPNAM    GROUP
SYSNAM    LOG_IO
```

The SYSTEST account also must have the following quota allowances:

```
ASTLM:10
DIOLM:12
BIOLM:12
TQCNT:20
```

In the system distributed by DIGITAL, the SYSTEST account already has these privileges and quotas assigned to it. To check that they remain assigned, issue the following two commands:

```
$ SHOW PROCESS/PRIVILEGES <RET>
$ SHOW PROCESS/QUOTAS <RET>
```

In response to each command, the terminal displays all the privileges or all the quotas in effect for the current account. If the privileges and quotas listed above are not present, run the AUTHORIZE utility to add them. (The VAX/VMS System Manager's Guide describes the AUTHORIZE utility.)

The UETP tests can run while the system is processing other work; however, you obtain better results if the tests run in isolation. For example, the UETP cannot test any device that is allocated to another process.

2.1.3 Preparing Devices

After logging in, you must set up all the devices to be tested. For quick reference, each subsection below concludes with a summary in italics of the preparatory steps.

2.1.3.1 **Setting Up Disk Drives** - For each disk drive in the system, perform the following steps:

- Provide a disk that does not contain any data worth preserving (that is, a scratch disk) and start up the drive. If the disk has not been initialized, use the INITIALIZE command to do so. For example:

  ```
  $ INITIALIZE/DATA_CHECK DMA0: TEST1 <RET>
  ```

  This command initializes a disk on an RK06 or RK07 drive (DMA0:) and assigns the volume label TEST1 to the disk. Table 2-1 lists the mnemonics for the various device types.

Table 2-1
Device Names

Mnemonic	Device Type
DB	RP04, RP05, RP06 Disk
DM	RK06, RK07 Disk
DR	RM03 Disk
LP	Line Printer
MT	TE16 Magnetic Tape
TT	Interactive Terminal

Note that all volumes must have unique labels.

- Issue a MOUNT command to connect the disk to the file system. For example:

```
MOUNT/SYSTEM DMA0: TEST1 <RET>
```

This command mounts the volume labeled TEST1 on the drive DMA0:. The qualifier /SYSTEM indicates that you are making the volume available to all users in the system.

- If the volume does not contain the directory [SYSTEST], issue a CREATE command to set it up. The UETP uses this directory when testing the disk. For example:

```
$ CREATE/DIRECTORY DMA0: [SYSTEST] <RET>
```

Summary:

Physically mount a scratch disk

Start up the drive

Issue one or more of the following commands as required:

```
$INITIALIZE/DATA_CHECK device-name: label <RET>

$MOUNT/SYSTEM device-name: label <RET>

$CREATE/DIRECTORY device-name: [SYSTEST] <RET>
```

2.1.3.2 **Setting Up Magnetic Tape Drives** - For each magnetic tape drive, perform the following steps:

- Physically mount a write-enabled scratch tape. The tape volume must be at least 600 feet long. Turn power on to the drive, position the tape at the Beginning Of Tape (BOT) marker, and press the ONLINE switch.

- If the tape has not been initialized, issue the INITIALIZE command to do so. For example:

```
$ INITIALIZE MT0:  TAPE0 <RET>
```

This command initializes the tape loaded on drive unit 0 and assigns the volume label TAPE0 to the tape. Note that each tape must have a unique volume label.

- Issue a MOUNT command to connect the disk to the file system. For example:

  ```
  $
  ```

  This command mounts the tape labelled TAPE0 on drive unit 0.

Summary:

Turn power on to the device

Physically mount a write-enabled scratch tape at least 600 feet long

Position the tape at the BOT marker

Press the ONLINE switch

If necessary, initialize the tape as follows:

```
$
```

Mount the tape as follows:

```
$
```

2.1.3.3 **Setting Up Terminals and Line Printers** - In order to be tested by the UETP, terminals and line printers must be powered up and must be online to the system. In addition, check that line printers and hard copy terminals are properly loaded with paper. The amount of paper required depends on the number of UETP runs that you intend to initiate. For each run, a line printer and a terminal both require 2 pages.

In addition, check that the terminals are all set to the correct baud rate and are assigned appropriate characteristics (see the VAX-11/780 Hardware User's Guide, Order No. EK-UG-780-UG-PRE).

Summary:

Turn power on to the device

Check paper supply if the device produces hard copy

Press the ONLINE switch

Check baud rates and terminal characteristics

2.1.3.4 **Other Devices** - The UETP does not test the following devices; their status has no effect on UETP execution:

- Card readers
- Network devices (DMC11s)
- Null devices

- Mailboxes
- Dialup terminal lines
- Nonstandard devices

Furthermore, the UETP does not test the console terminal, the console floppy disk drive, and the terminal used to initiate the UETP. If you are able to boot the system, log in, and start the UETP, you have shown that these devices are usable.

2.2 DEFINING UETP VARIABLES

This section explains several variables that you must define each time you run the entire UETP package. These UETP variables determine:

- The amount of information to be output to the console
- The number of users to be simulated by the UETP in the system load test
- The number of consecutive runs to be made by the UETP
- The magnetic tape drive to be used for a test of the FLX facility and VAX-11 RMS services

You decide how much information should be output to the console by including or omitting the /OUTPUT qualifier to the call to the UETP. The remaining three variables are defined by your answers to three questions that the UETP asks when it starts up.

2.2.1 The Console Log

To initiate the UETP, you issue a call to the UETP master command procedure as follows:

```
$ @UETP [/OUTPUT=filespec] <RET>
```

If you do not specify the /OUTPUT qualifier, the UETP proceeds to send all its output to the console, which, in this context, is the terminal from which you issued the call. In most cases, however, it is more convenient for the UETP to write most of its output to a disk file. Furthermore, if you run the UETP from a hard-copy terminal, the typing of all the output considerably slows down the tests. This slowdown is multiplied if you request continuous UETP runs from a hard-copy terminal.

By appending the /OUTPUT qualifier to the UETP call, you request a short console log. The UETP then creates an output file, with the name you specified, on the system disk in the [SYSTEST] directory. During the run, the UETP displays status information at the console such as error messages and notifications of the beginning and ending 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 the output file for further information. This disk file contains most of the output generated by various phases of the UETP, plus the status information displayed at the console. Some phases have additional separate output files. For example, the load test generates a large

amount of information which in itself is not very significant. This information is written to a file called UETPLOG.LOG (see Section 2.3.2).

2.2.2 The Load Test

The UETP displays the following prompt; your answer to the prompt (n) determines the number of detached processes to be created by the load test.

```
ENTER NUMBER OF LOAD TEST USERS [0]:n <RET>
```

Each detached process executes a command procedure and thus simulates a user entering commands from a terminal. The purpose of the test is to create a situation in which each process is competing with other processes for system resources. The console displays a message when each process begins and when each process ends so you can determine the number of currently active simulated users. Each process is deleted when it finishes its command procedure.

The maximum number of users that you should specify depends on the amount of memory in your VAX/VMS system. Table 2-2 provides a guideline for selecting a number of users appropriate to the amount of memory available to you.

Table 2-2
Guideline for Selecting Number of Load Test Users

System	Size of Memory	Number of load test users
RP-Based	256K	10
	384K	15
	512K	20
	640K	25
	768K	30
	896K	35
	1 megabyte	40
RK-Based	256K	6
	384K	9
	512K	12
	640K	15
	768K	18
	896K	21
	1 megabyte	25

2.2.3 Single-Run Versus Continuous UETP Execution

The UETP displays the following prompt; your response (n) determines the number of runs it makes.

```
ENTER NUMBER OF COMPLETE UETP RUNS [0]:n <RET>
```

The UETP can be run as a quick check that the system is working, or it can be repeated over and over to see how the system responds to continuous use over a period of time. If you type 1 in response to the above question, the UETP stops after completing its initial run. If you specify a number greater than 1, the UETP continuously restarts itself until it completes the number of runs specified.

For example, as part of a VAX/VMS system's final assembly and test (FA&T), a manufacturing technician starts up the UETP at the end of the day, specifies 20 complete UETP runs, and lets the system run all night. In contrast, a field service representative is interested in verifying that a newly installed system works, and therefore tends to run the UETP only once or twice at a time.

When you intend to specify multiple UETP runs, be sure to request a short console log (see Section 2.2.1), especially if you are working from a hard-copy terminal. You should also ensure that all the line printers and hard-copy terminals to be tested have enough paper loaded to last through all the test runs.

2.2.4 The FLX and VAX-11 RMS Magnetic Tape Tests

In the following prompt, the UETP asks you to enter the device name of a magnetic tape drive.

```
ENTER SCRATCH MAGTAPE (E.G. MT0:) OR A <CR>: device-name <RET>
```

The drive should have a write-enabled scratch tape loaded on it. The UETP uses the specified drive for testing the compatibility mode utility FLX and the VAX-11 RMS magnetic tape services. FLX is a data transfer and conversion utility that enables the transfer of data to and from tape and disk files written in DOS-11 and RT-11 formats. VAX-11 RMS is a set of record management services used by the VAX/VMS system for data operations and record storage on disk in Files-11 format and on magnetic tape in ANSI format.

If you press <CR> instead of naming a drive, the compatibility mode phase skips the FLX test and the test of VAX-11 RMS services omits magnetic tape.

2.3 RUNNING THE ENTIRE UETP

The following dialogue shows how to initiate one or more complete UETP runs.

```
$ @UETP [/OUTPUT=filespec] <RET>

   VAX/VMS UETP STARTED: mm/dd/yy hh:mm

ENTER NUMBER OF LOAD TEST USERS [0]:n <RET>
ENTER NUMBER OF COMPLETE UETP RUNS [0]:n <RET>
ENTER SCRATCH MAGTAPE (E.G. MT0:) OR A <CR>:device-name: <RET>
```

When you have entered the first line, optionally specifying a short console log, the UETP responds by asking the three questions shown. (See Sections 2.2.2, 2.2.3, and 2.2.4 for explanations of these questions.) After you have answered the third question, the UETP initiates its sequence of tests. The tests run to completion without further input from you.

2.3.1 Using CTRL/Y and CTRL/C

The control characters CTRL/Y and CTRL/C allow you to interrupt and terminate UETP execution before it completes normally.

2.3.1.1 **CTRL/Y** - CTRL/Y interrupts the current UETP test and temporarily returns control to the command language interpreter. While the test is interrupted, you can issue a subset of system commands; this subset is defined in the VAX/VMS Command Language User's Guide. You then either terminate the test by typing STOP or continue the test from the point of interruption by typing CONTINUE. If you type STOP, the entire UETP aborts and control returns to the command language interpreter.

Note that CTRL/Y does not affect detached processes already created by the interrupted test phase. For example, the device test creates detached processes to handle individual controllers. When you press CTRL/Y during the device test, the individual tests already in progress continue uninterrupted, even if you then type STOP to abort the test phase.

2.3.1.2 **CTRL/C** - Several UETP test 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
```

You cannot continue a test phase after you press CTRL/C to stop it; the UETP continues onto the next test in the master command procedure. Note that CTRL/C also stops any detached processes already created by the current test phase.

2.3.2 UETP Log Files

At the end of a single or multiple pass of the UETP package, the [SYSTEST] directory on the system disk contains various log files that document one or more test phases. These log files record details of each phase for reference if the terminal log indicates one or more error conditions. The following list describes each log file.

- filespec (that is, @UETP/OUTPUT=filespec) - An optional log created if you request an output file in the call to the UETP. This file contains all the information displayed at the terminal as well as information that describes the progress of the tests in somewhat greater detail.

- UETPLOG.LOG - A large log file that is a concatenation of individual log files from the following tests:

  The I/O device tests
  The native mode utility tests
  The system load test
  The compatibility mode tests

- SSLOG.LOG - A log file created by the native mode system services test. This file contains information on the testing of each system service.

- UNATIVE.LOG - A log file containing the output from the native mode utility test. UETPLOG.LOG includes this file.

- UCOMP.LOG - A log file containing the output from the compatibility mode test. UETPLOG.LOG includes this file.

In addition to the files described briefly above, a file called LOAD.LOG and a file called LOGP.LOG are also present. These files originate from the system load test and the I/O device test respectively. Both these tests create a variable number of detached processes and each detached process generates its own log file, a version of LOAD.LOG or LOGP.LOG. At the end of a pass, the UETP concatenates all the LOAD.LOG files and all the LOGP.LOG files; the concatenated files become part of UETPLOG.LOG. The UETP then purges the individual load test and device test logs so that only the highest versions of LOAD.LOG and LOGP.LOG remain.

If a UETP run does not complete normally and therefore is unable to clean up its files, the [SYSTEST] directory can contain many versions of the LOAD and LOGP files, as well as other temporary files. In this case, either delete them yourself or rerun the UETP. When the UETP starts up, it checks for excess log files left behind by an interrupted run. If such files exist, the UETP deletes them before running any new tests.

2.4 RUNNING INDIVIDUAL UETP PHASES

When you run the entire UETP, it automatically steps through a sequence of phases that test various parts and functions of the system. Each phase can be run separately so that you can test a specific part or function in isolation. This section gives operating instructions for directly initiating each phase.

In summary, this section tells you how to run

- The device tests (2.4.1)
- The native mode tests (2.4.2)
- The system load test (2.4.3)
- The compatibility mode test (2.4.4)

Note that you must be logged into the SYSTEST account to run the individual tests as described in this section (see Section 2.1.2).

2.4.1 The Device Tests

The UETP device test consists of an executable image called UETPDEV01.EXE, which creates a detached process for every device controller to be tested. For example, if a system includes three terminal controllers, the device test creates three detached processes to test terminals. In parallel, the detached processes execute images that test different types of devices. If you do not want to test all the device types, you can run an image that tests only one specific controller.

Note that the magnetic tape test requires a reel containing at least 600 feet of tape.

Table 2-3 names the device test images, indicates the type of device that each tests, and shows how to invoke each test.

Table 2-3
The Device Tests

Test Image Name	Devices Tested	Command(s) to Invoke the Test
UETPDEV01.EXE	Disks Line printers Magnetic tapes Terminals	$RUN UETPDEV01
UETDISK00.EXE	Disks	$DEFINE/GROUP UETP$CTRLNAME DR×* DB× <RET> DM× $RUN UETDISK00 <RET>
UETPRIN00.EXE	Line Printers	$DEFINE/GROUP UETP$CTRLNAME LP×* <RET> $RUN UETPRIN00 <RET>
UETTAPE00.EXE	Magnetic tapes	$DEFINE/GROUP UETP$CTRLNAME MT×* <RET> $RUN UETTAPE00 <RET>
UETTTYS00.EXE	Terminals	$DEFINE/GROUP UETP$CTRLNAME TT×* <RET> $RUN UETTTYS00 <RET>

* The DEFINE command assigns a controller name to the group logical name UETP$CTRLNAME. The lower case x represents a variable letter that distinguishes like controllers from one another (for example, TTA versus TTB). See the explanation of the logical name UETP$CTRLNAME in Section 2.4.1.1.

2.4.1.1 **The Logical Name UETP$CTRLNAME** - The initial phase of the UETP creates a file called UETINIDEV.DAT, which contains data on the controllers in the system and their associated devices. UETPDEV01 uses the information in this file to find a controller name to pass to each detached process that it creates. UETPDEV01 passes the name by assigning it to the group logical name UETP$CTRLNAME. Each detached process then uses that logical name to determine which devices to test.

As long as the file UETINIDEV.DAT exists, UETPDEV01 can automatically pass the controller names to the device tests. If you have run the UETP before, the file exists unless you have explicitly deleted it.

However, when you run an image that tests a particular controller, you must explicitly define UETP$CTRLNAME. For example:

```
$ DEFINE/GROUP UETP$CTRLNAME TTB <RET>
$ RUN UETTTYS00 <RET>
```

This example assigns the controller name TTB to the logical name UETP$CTRLNAME and then runs the terminal test image UETTTYS00.EXE. If your system has more than one terminal controller and you want to test all connected terminals, you must both define UETP$CTRLNAME and run UETTTYS00.EXE separately for each controller.

When you run a device test for a specific controller, the group logical name UETP$CTRLNAME remains assigned after the test completes. To deassign this logical name, issue the following command:

```
$ DEASSIGN/GROUP UETP$CTRLNAME <RET>
```

When UETDEV01 initiates the individual device tests, it deassigns the logical name UETP$CTRLNAME at the end of the tests.

2.4.1.2 **Test Output** - When UETPDEV01 initiates the device tests, they record their results in versions of the file LOGP.LOG, which is listed in the [SYSTEST] directory on the system disk. When the tests run as part of the package, the UETP eventually concatenates all the versions of LOGP.LOG into the file UETPLOG.LOG and then purges the individual device test logs. However, when you run UETPDEV01 separately, the log files remain in versions of LOGP.LOG until you explicitly delete them.

When you run a test that exercises a specific controller (UETDISK00, for example), the test sends its output to your terminal.

2.4.2 The Native Mode Tests

The native mode test phase includes four separate tests:

- The system services test (UETNATV01.EXE)
- The native mode utility test (UETNATV02.COM)
- The VAX-11 RMS record management services test (UETNRMS00.COM)
- The VAX-11 FORTRAN IV-PLUS compiler test (UETFORT00.COM)

2.4.2.1 **The System Services Test** - The system services test consists of a collection of images, where each image tests one or more system services. To activate the service tests, issue the following command:

```
$ RUN UETNATV01 <RET>
```

The image UETNATV01.EXE initiates a detached command procedure called SSTEST.COM, which in turn calls the individual test images. Each test image issues numerous calls to the tested system services. Several calls are intended to produce a success return code. These calls test

various legal ways to invoke the service; in so doing, the image also tests how the service responds to a call in different system contexts. The remainder of the calls purposefully generate failure codes defined for the service.

The image checks the results of each call against results that are known to be correct. If all the results match, the test is successful; if the image detects any discrepancies, the test fails. In general, the success or failure of one service does not affect the testing of any other service.

Test Output - The system services test sends the following status messages to the terminal:

```
%UETP-I-BEGIN, UETNATV01 beginning at 'date' 'time'
%UETP-I-ABORTC, UETNATV01 to abort this test, type ^C
%UETP-I-TEXT, UETNATV01 # VMS system services errors found is 'n'
%UETP-I-ENDED, UETNATV01 ended at 'date' 'time'
```

The third message reports the number of system services that failed the test. If n is not equal to 0, check a file on the system disk called SSLOG.LOG. When the system services test starts up, it creates the file SSLOG.LOG for logging status information on the results of each service's test. The log can therefore tell you which service, if any, has failed. See the description of the SATSMS messages in Chapter 3.

2.4.2.2 **The Native Mode Utility Test** - The native mode utility test tests the VAX-11 Symbolic Debugger and the Image File Patch Utility. To activate the test, you run the command procedure UETNATV02.COM as follows:

```
$ @UETNATV02 [/OUTPUT=filespec] [utility] <RET>
```

where utility is one of the following values:

```
ALL
DEBUG
PATCH
```

By default, the command procedure tests both utilities; however, you can limit the test to one utility by specifying either DEBUG or PATCH.

These utility tests are similar to the compatibility mode utility tests. Section 2.4.4 describes how each utility is tested.

Test Output - When you run the native mode utility test on its own, it sends all its output to the terminal unless you specify an output file. For example:

```
$ @UETNATV02 /OUTPUT=NMCATH.LOG <RET>
```

This example directs the command procedure to write all its output to the file NMCATH.LOG on the system disk.

When the test runs as part of the UETP package, the utility output is written to a file called UNATIVE.LOG, which is duplicated in UETPLOG.LOG.

2.4.2.3 **The VAX-11 RMS Test** - The VAX-11 RMS test exercises many of the record management options provided by VAX-11 RMS services. The test runs under the control of the command procedure UETNRMS00.COM. To include magnetic tape exercises in the test, the global symbol MAGTAP must be defined. Section 2.2.4 describes the prompt that requests a magnetic tape specification when you start up a complete run of the UETP. Your response to this prompt defines the symbol MAGTAP. When you run the VAX-11 RMS test on its own, you must explicitly define the symbol MAGTAP.

The following command sequence defines the global symbol MAGTAP and runs the VAX-11 RMS test:

```
$ MAGTAP:==device-name: <RET>
$ @UETNRMS00 [/OUTPUT=filespec] <RET>
```

To test the various I/O functions provided by VAX-11 RMS, the test creates files on disk and on magnetic tape (if the global symbol MAGTAP is defined). At the end of the test, these files are deleted and the magnetic tape rewound to Beginning of Tape (BOT).

Test Output - When you run the VAX-11 RMS test separately, the test sends status messages and any error messages to the terminal or to an output file if you have requested one. (See the second command above, which shows the optional output file specification.) When the test runs as part of the UETP package, all the VAX-11 RMS test messages appear at the terminal. If you have specified an output file in the call to the UETP (that is, @UETP /OUTPUT=filespec), the messages also appear in that file.

2.4.2.4 **The VAX-11 FORTRAN IV-PLUS Compiler Test** - To run the VAX-11 FORTRAN IV-PLUS compiler test, enter the following command:

```
$ @UETFORT00  [/OUTPUT=filespec]  <RET>
```

Commands within the procedure UETFORT00.COM compile, link, and run two FORTRAN programs. The first program tests the compiler by performing functions that include:

- input/output operations
- DO loops
- integer arithmetic
- logical and arithmetic expressions
- IF statements
- arithmetic assignments
- arrays

The second program tests the compiler's floating point facility and exercises the VAX-11 FORTRAN IV-PLUS subroutine library by accessing the double precision SIN function.

The test compares the results of both programs with results that are known to be correct and reports an error for each discrepancy it finds.

Test Output - For each program, the test creates an object file, an executable image, and a listing file. The test eventually deletes the listing and object files, but the other files remain on disk in the [SYSTEST] directory. Specifically, the files that remain are UETFORT01.EXE and UETFORT02.EXE. If you do not want to keep these files, use the DELETE command to eliminate them. For example:

```
$ DELETE UETFORT01.EXE;*  <RET>
```

By putting an asterisk in the version field of the file specification, you can ensure that the system deletes versions of the file that may remain from previous test runs.

The test writes all status and error messages to the terminal or to an output file if the call to the command procedure UETFORT00.COM included an /OUTPUT qualifier. When the test runs as part of the UETP package, the messages appear in the console log and in a disk file if the call to the master command procedure UETP.COM included an /OUTPUT qualifier.

When the VAX/VMS system being tested does not include a VAX-11 FORTRAN IV-PLUS compiler, the compiler test fails after attempting to locate the necessary source files. The messages that appear in these circumstances are as follows:

```
$ SET VERIFY
$
$ !NATIVE MODE FORTRAN TESTS
$
$ SET NOVERIFY
  dd-mmm-yy hh:mm:ss
$ FORT/WARNINGS/CHECK:ALL/LIST UETFORT01
% RMS-E-FNF, FILE NOT FOUND
$ FINI:
$ SET NOVERIFY
PIP -- NO SUCH FILE(S)
SY0:[1,7]UETFORT01.OBJ;*
PIP -- NO SUCH FILE(S)
SY0:[1,7]UETFORT02.OBJ;*
  dd-mmm-yy hh:mm:ss
```

2.4.3 The System Load Test

To run the system load test, type:

```
$ DEFINE/GROUP UETP$USERS n <RET>
$ RUN UETLOAD01 <RET>
```

The value of n is the number of users to be simulated. The purpose of the load test is to simulate a number of terminal users who are simultaneously demanding system resources. When you run the UETP as a whole, it prompts for the number of users to be simulated (see Section 2.2.2). Your response to the request (ENTER NUMBER OF LOAD TEST USERS [D]:) defines the group logical name UETP$USERS. The load test uses this logical name to determine the number of detached processes to create; each detached process executes a command procedure to represent one user issuing commands from a terminal.

Before you can run the load test separately, you must explicitly define the logical name UETP$USERS. (Note that Table 2-2 in Section 2.2.2 provides a guideline for choosing a number of users based on the amount of available memory in the VAX/VMS system being tested.) For example:

```
$ DEFINE/GROUP UETP$USERS 12 <RET>
$ RUN UETLOAD01 <RET>
%UETP-I-BEGIN, UETLOAD01 beginning at 22-MAY-1978 10:51:59.52
%UETP-I-ABORTC, UETLOAD01 to abort this test, type ^C
%UETP-I-USER, UETLOAD01 1 user running
%UETP-I-USER, UETLOAD01 2 users running
%UETP-I-USER, UETLOAD01 3 users running
%UETP-I-USER, UETLOAD01 4 users running
%UETP-I-USER, UETLOAD01 5 users running
%UETP-I-USER, UETLOAD01 6 users running
%UETP-I-USER, UETLOAD01 7 users running
%UETP-I-USER, UETLOAD01 8 users running
%UETP-I-USER, UETLOAD01 9 users running
%UETP-I-USER, UETLOAD01 10 users running
%UETP-I-USER, UETLOAD01 11 users running
%UETP-I-USER, UETLOAD01 12 users running
%UETP-I-USER, UETLOAD01 11 users running
%UETP-I-USER, UETLOAD01 10 users running
%UETP-I-USER, UETLOAD01 9 users running
%UETP-I-USER, UETLOAD01 8 users running
%UETP-I-USER, UETLOAD01 7 users running
%UETP-I-USER, UETLOAD01 6 users running
%UETP-I-USER, UETLOAD01 5 users running
%UETP-I-USER, UETLOAD01 4 users running
%UETP-I-USER, UETLOAD01 3 users running
%UETP-I-USER, UETLOAD01 2 users running
%UETP-I-USER, UETLOAD01 1 user running
%UETP-I-ENDED, UETLOAD01 ended at 22-MAY-1978 11:01:17.86
$
```

In this example, the DEFINE command assigns the number 12 to the group logical name UETP$USERS. When the load test executes in response to the next command, it creates 12 detached processes which execute different command procedures. (The load test in fact uses only 10 different command procedures to simulate users. For more than 10 users, the load test runs multiple copies of the command procedures.)

To deassign the group logical name UETP$USERS after the system load test completes, issue the command

```
$ DEASSIGN/GROUP  UETP$USERS  <RET>
```

The UETP master command procedure deassigns all group logical names assigned by its tests as part of the termination phase. The group logical name UETP$USERS remains assigned only if you run the system load test separately or if the UETP package does not complete normally.

Test Output - 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 LOAD.LOG. The console displays status information only as the load test progresses. (See the example above for the types of messages displayed.)

When the load test runs as part of the entire UETP, the UETP appends the LOAD.LOG files, writes the appended output to the file UETPLOG.LOG, and deletes the individual output files. However, when the load test runs separately, the LOAD.LOG files remain, one for each simulated user; they are not appended or written to another file. You must delete them yourself.

For example, the following listing shows all the LOAD.LOG files created by the load test illustrated above.

```
$ DIRECTORY LOAD.LOG;*  <RET>

DIRECTORY DB01:[SYSTEST]
22-MAY-78 11:05

LOAD.LOG;12          2.          22-MAY-78
LOAD.LOG;11          5.          22-MAY-78
LOAD.LOG;10          18.         22-MAY-78
LOAD.LOG;9           24.         22-MAY-78
LOAD.LOG;8           39.         22-MAY-78
LOAD.LOG;7           2.          22-MAY-78
LOAD.LOG;6           40.         22-MAY-78
LOAD.LOG;5           17.         22-MAY-78
LOAD.LOG;4           5.          22-MAY-78
LOAD.LOG;3           56.         22-MAY-78
LOAD.LOG;2           2.          22-MAY-78
LOAD.LOG;1           5.          22-MAY-78

TOTAL OF 215./297. BLOCKS IN 12. FILES
```

Issue the following command to delete the load test output files:

```
$ DELETE LOAD.LOG;* <RET>
```

2.4.4 The Compatibility Mode Test

When the compatibility mode phase runs as part of the entire UETP, it tests all the utilities listed below. When you run this phase separately, you can choose to test all these utilities or only one specific utility. The command procedure UETCOMP00.COM issues several commands to each utility and then, in most cases, uses the File Compare (DIF) utility to compare the results of the test commands with results that are known to be correct.

To start up this phase, issue the following call to the UETCOMP00.COM command file, optionally specifying ALL or the name of a utility:

```
[$ MAGTAP:==device-name <RET>]
$ @UETCOMP00 [/OUTPUT=filespec] [utility] <RET>
```

where utility is one of the following values:

ALL (the default)	PIP
DMP	SLP
FLX	SOS
LBR	SRT
PAT	VFY

By default, the test applies to all the utilities listed here. Note, however, that you must explicitly define the global symbol MAGTAP to include FLX in the test.

2.4.4.1 **The FLX Test** - The FLX test uses the global symbol MAGTAP to determine which tape drive to use for testing. When you start up a complete run of the UETP, it asks you to specify the name of a tape drive if you want to test FLX (see Section 2.2.4). If you specify a drive, the UETP assigns the device name to the global symbol MAGTAP. You must define the global symbol MAGTAP yourself if you run the compatibility mode phase separately. For example:

```
$ MAGTAP:==MT0:  <RET>
$ @UETCOMP00 <RET>
```

This command sequence assigns the device name MT0: to the global symbol MAGTAP and then invokes the compatibility mode tests. If you invoke the tests without defining the symbol MAGTAP, the phase skips the FLX test.

The following commands define the global symbol MAGTAP and specifically invoke the FLX test:

```
$ MAGTAP:==MT1:  <RET>
$ @UETCOMP00 FLX <RET>
```

2.4.4.2 **Test Output** - When you run the compatibility mode test on its own, it sends all its output to the terminal unless you specify an output file. For example:

```
$ @UETCOMP00 /OUTPUT=UTILITY.LOG <RET>
```

This example directs the command procedure to write all its output to the file UTILITY.LOG on the system disk.

When the test runs as part of the UETP package, the utility output is written to a file called UCOMP.LOG, which is duplicated in the file UETPLOG.LOG.

CHAPTER 3

UETP MESSAGES

This chapter lists in alphabetical order and describes the messages returned by the UETP tests themselves. It does not describe the messages returned by components of the VAX/VMS system as a result of the testing. For explanations of the latter type of message you must refer to the VAX/VMS System Messages and Recovery Procedures Manual or to the manual that describes the part of the system that returned the message. For example, if the compatibility mode test causes a utility to detect an error, the utility itself returns an error message. To clarify this message, you must consult the manual that describes the utility.

In Section 3.2, variable parts of each message are enclosed in apostrophes. For example, in the message

```
READERR, error reading 'input-file'
```

the value 'input-file' is determined by the program that encountered the error.

3.1 FORMAT OF SYSTEM MESSAGES

The general format of messages displayed by the VAX/VMS operating system is:

```
%FACILITY-L-CODE, TEXT
[-FACILITY-L-CODE, TEXT]
```

where:

FACILITY is the name of the system component that issues the message.

L is a severity level indicator. It has one of the following values:

Code	Meaning
S	Success
I	Information
W	Warning
E	Error
F	Fatal or severe error

Severity levels are defined in more detail in the VAX/VMS System Messages and Recovery Procedures Manual.

CODE is an abbreviation of the message text; the message descriptions in Section 3.2 of this manual are alphabetized by this code.

TEXT is the explanation of the message.

[-FACILITY...]
is a related message.

3.2 ALPHABETICAL LIST OF MESSAGES

ABORT, 'testname' aborted [at ['date'] 'time']

Explanation: A test ended abnormally.

User action: Investigate the reason for the abnormal termination of the test.

ABORTC, 'testname' to abort this test, type ^C

Explanation: The image displaying this message responds to CTRL/C by terminating gracefully and passing control to the command language interpreter. You cannot restart the image after typing CTRL/C. Use CTRL/Y to temporarily interrupt an image.

User action: None.

ATPC, at PC 'location'

Explanation: This message displays a PC location to provide further information about an error described in one or more accompanying messages.

User action: If the error is severe or reproducible, use a Software Performance Report (SPR) to describe the error to DIGITAL.

BADFIELD, 'record' field invalid at PC 'location'

Explanation: The VAX-11 RMS test detected a discrepancy in the record, found at the PC location specified, which VAX-11 RMS itself did not detect.

User action: Rerun the test and, if the error recurs, send a Software Performance Report (SPR) to DIGITAL to report it.

BADLOGIC, internal logic error detected [at PC 'location']

Explanation: An unexpected internal software error occurred.

User Action: Collect as much information as possible and send a Software Performance Report (SPR) to DIGITAL.

BADWORD, invalid data 'xxxxxxxx' at 'location'

Explanation: An unexpected word of data was encountered.

User action: Collect as much information as possible and send a Software Performance Report (SPR) to DIGITAL.

BEGIN, 'testname' beginning at 'date' 'time'

Explanation: This message announces the beginning of a specific test.

User action: None.

CONF, the following devices are sysgened into this system

Explanation: The image UETINIT00 displays this message before listing all the controllers and devices generated into the system. The message and the list appear in a disk file if you requested a short console log (see Section 2.2.1). Otherwise, they appear at the terminal that initiated the tests.

User action: None.

DATAER, data error on 'device-type' unit 'number' byte 'location'
good data = 'xxxxxxxx' bad data = 'xxxxxxxx'

Explanation: The disk test (UETDISK00) detected a difference between the data that was written from memory to a device and the same data after it was written back from the device to memory.

User action: Investigate the device that caused the error.

DDB, UETINIT00 DDB 0 'controller' 00000000 00000000

Explanation: The message displays the name of a controller generated into the system. The message is part of a listing that describes all controllers and devices in the system.

User action: None.

DENOSU, 'testname' device 'device-type' is not supported

Explanation: The device name in the message is not known to and therefore cannot be tested by the UETP (NET, for example).

User action: None.

DESTP, 'testname' stopped testing 'controller' unit 'number' at 'time'

Explanation: A device that passed the simple read/write test in the initial phase of the UETP could not complete its test in the device test phase. For example, this problem can occur on an RK06 disk that is being used as the system disk; the disk does not have enough free space to hold the test files.

User action: If you think that the device should be tested by the UETP, investigate and fix the problem; otherwise, ignore the message.

DEUNUS, 'testname' device 'device-name' is unusable, error code
= 'xxxxxxxx'

Explanation: The specified device failed to pass the simple read/write test in the initial phase of the UETP. Subsequent tests in the UETP package will not attempt to test the device. Another message usually follows to explain why the device failed the test. Possible causes are that the device is down, offline, not mounted, not initialized, or not write-enabled.

User action: Either make the device usable or ignore the message.

ENDED, 'testname' ended at 'date' 'time'

Explanation: This message announces the completion of a specific test.

User action: None.

ERBOX,

Error count = 'nnn'

Explanation: This message assigns a sequence number to an error described in a subsequent message.

User action: None.

NOTRAN, no string translation performed

Explanation: The I/O device tests use the group logical name UETP$CTRLNAME to obtain the name of a controller whose devices are to be tested. The system load test uses the group logical name UETP$USERS to determine the number of detached processes to be created. This message appears when you run either the I/O device tests or the system load test separately and the appropriate logical name has not been defined.

User action: Define the group logical name UETP$CTRLNAME or UETP$USERS and rerun the tests. Note that you only need to define the logical names explicitly when you run the I/O device tests or system load test separately. See Section 2.4.1 for further information.

OPENIN, error opening 'input-file' as input

Explanation: Self-explanatory. This message is usually accompanied by a VAX-11 RMS message indicating the reason for the failure.

User action: Take corrective action based on the associated message.

OPENOUT, error opening 'output-file' as output

Explanation: Self-explanatory. This message is usually accompanied by a VAX-11 RMS message indicating the reason for the failure.

User action: Take corrective action based on the associated message.

READERR, error reading 'input-file'

Explanation: Self-explanatory. This message is usually accompanied by a VAX-11 RMS message indicating the reason for the failure.

RMSERROR, RMS service error

Explanation: The VAX-11 RMS test returns this message when VAX-11 RMS itself encounters an error. A subsequent RMS message then describes the actual error.

User action: If possible, correct the situation that caused the error described in the RMS message. The VAX/VMS System Messages and Recovery Procedures Manual describes all the VAX-11 RMS messages.

SATSMS, test module 'testname' 'status'

Explanation: This message announces that the testing of a specific system service has begun, ended successfully, or failed. The message appears in the file SSLOG.LOG, which is created by the native mode system service test (see Section 2.4.2.1).

User action: None if 'status' is begun or successful. However, if 'status' is failed, the test supplies a series of messages to describe the reasons for failure. In this case, see the description of the TEXT message for suggested user action.

SYSERROR, 'testname' system service error [at PC 'location']

Explanation: A test received an unexpected error return from a VMS system service.

User action: A suggestion is to run the native mode system services test (see Section 2.4.2.1); the test might reproduce the error that caused this message to appear. If the error occurred because of a quota or privilege violation, refer to Section 2.1.2, which explains how to modify privilege and quota allocations for the SYSTEST account. For other types of errors, collect as much information as possible and send a Software Performance Report (SPR) to DIGITAL.

SYSERRORPC, 'testname' system service error at PC 'location'

Explanation: A test received an unexpected error return from a VMS system service.

User action: A suggestion is to run the native mode system service test (see Section 2.4.2.1); the test might reproduce the error that caused this message to appear. If the error occurred because of a quota or privilege violation, refer to Section 2.1.2, which explains how to modify privilege and quota allocations for the SYSTEST account. For other types of errors, collect as much information as possible and send a Software Performance Report (SPR) to DIGITAL.

TEXT, 'text'

Explanation: Various UETP tests use this message to provide information, usually self-explanatory, of one kind or another. Specifically, the native mode system services test uses this type of message to explain why a system service failed its test.

User action: In most cases, none. However, note the following suggested user actions if a TEXT message written to the SSLOG.LOG file describes a system service test failure. (1) Check the quota and privilege allocations for the SYSTEST account if the message indicates a quota or privilege violation. (The message indicates such a violation by including the appropriate system service error code.) See Section 2.1.2 for information on examining and modifying account allocations for quotas and privileges. (2) Forward a listing of SSLOG.LOG to DIGITAL if other types of system service test failures are described and are repeatable.

UCB, UETINIT00 UCB 1 'unit-number' 'xxxxxxxx' 'xxxxxxxx' 'xxxxxxxx'

Explanation: This message describes a device unit on a controller named in a preceding DDB message. The three hexadecimal values are equal to the device characteristics words 1, 2, and 3. (The VAX/VMS I/O User's Guide contains information on device characteristics words.)

User action: None.

UNXPCTSTS, unexpected status detected

Explanation: The VAX-11 RMS test encountered a condition other than End of File (EOF) when it expected to find EOF. A VAX-11 RMS message follows the UNXPCTSTS message and explains the condition actually encountered.

User action: See the description of the accompanying VAX-11 RMS message in the VAX/VMS System Messages and Recovery Procedures Manual.

USER, UETLOAD01 'nnn' user[s] running

Explanation: The image UETLOAD01 issues this message to announce the number of users currently active in the system load test (see Section 2.4.3).

User action: None.

WRITEERR, error in writing 'file-spec'

Explanation: A test was unable to write to the specified file.

User action: Remove the write lock (if any), retry the test, or report the problem (if reproducible) to DIGITAL by means of a Software Performance Report (SPR).

APPENDIX A

SUMMARY OF OPERATING INSTRUCTIONS

This appendix summarizes the UETP operating instructions to provide a quick reference section for those who are already familiar with running the UETP. For further information about any instructions given below, see the appropriate section in Chapter 2, which explains the instructions in detail.

A.1 LOGGING IN

Log into the SYSTEST account as follows:

```
<RET>
Username: SYSTEST <RET>
Password: <RET>
```

Note that the system does not echo the password.

A.2 PREPARING DEVICES FOR TESTING

This section tells you how to prepare different kinds of devices for testing by the UETP.

A.2.1 Disk Drives

To prepare each disk for testing, perform the following steps:

- Physically mount a scratch disk
- Start up the drive
- Issue one or more of the following commands as required:

```
$ INITIALIZE/DATA_CHECK device-name: label
$ MOUNT/SYSTEM device-name: label
$ CREATE/DIRECTORY device-name:  [SYSTEST]
```

A.2.2 Magnetic Tape Drives

To prepare each magnetic tape drive for testing, perform the following steps:

- Turn power on to the device
- Physically mount a write-enabled scratch tape at least 600 feet long
- Position the tape at the BOT marker
- Press the ONLINE switch
- If necessary, initialize the tape by entering the command

```
$ INITIALIZE device-name: label
```

- Mount the tape by entering the command

```
$ MOUNT device-name: label
```

A.2.3 Terminals and Line Printers

Prepare terminals and line printers for testing by performing the following steps:

- Turn power on to the device
- Check the paper supply if the device produces hard copy (2 pages for each pass of the UETP)
- Press the ONLINE switch
- Check baud rates and terminal characteristics (see the VAX-11/780 Hardware User's Guide, Order No. EK-UG780-UG-PRE)

A.2.4 Other Devices

The UETP does not test the following devices:

- Card reader
- Network devices (DMC11s)
- Null devices
- Mailboxes
- The console terminal and the console floppy disk
- The terminal used to initiate the UETP tests
- Dialup terminal lines
- Nonstandard devices

A.3 RUNNING THE ENTIRE UETP

To initiate the UETP test package, enter a call to the UETP master command procedure and respond to the three prompts shown below:

```
$ @UETP [/OUTPUT=filespec] <RET>

  VAX/VMS UETP STARTED dd-mmm-yy hh:mm

ENTER NUMBER OF LOAD TEST USERS [0]: n <RET>
ENTER NUMBER OF COMPLETE UETP RUNS [0]: n <RET>
ENTER SCRATCH MAGTAPE (E.G. MTO:) OR A <CR>: device-name: <RET>
```

Sections 2.2.2, 2.2.3, and 2.2.4 explain the three prompts in detail. Table 2-2 provides a guideline for choosing the number of load test users according to the amount of memory in the VAX/VMS system being tested.

Use CTRL/Y or CTRL/C to interrupt the tests (see Section 2.3.1).

A.4 RUNNING INDIVIDUAL UETP PHASES

This section shows how to initiate individual UETP test phases.

A.4.1 The Device Tests

To test disks, line printers, magnetic tapes and terminals all at once, issue the following command.

```
$ RUN UETPDEV01 <RET>
```

Note that the file UETINIDEV.DAT must exist on the system disk; see Section 2.4.1.1.

To test disks only, issue the following commands:

```
$ DEFINE/GROUP UETP$CTRLNAME DMx
$ RUN UETDISK00
```

where x is a letter that identifies a specific controller (DMB, for example). The group logical name UETP$CTRLNAME must be defined explicitly when you run individual device tests.

To test line printers only, issue the following commands:

```
$ DEFINE/GROUP UETP$CTRLNAME LPx
$ RUN UETPRIN00
```

To test magnetic tapes only, issue the following commands:

```
$ DEFINE/GROUP UETP$CTRLNAME MTx
$ RUN UETTAPE00
```

To test terminals only, issue the following commands:

```
$ DEFINE/GROUP UETP$CTRLNAME TTx
$ RUN UETTTYS00
```

A.4.2 The Native Mode Tests

The native mode test phase includes four separate tests:

- The system services test
- The native mode utility tests
- The VAX-11 RMS record management services test
- The VAX-11 FORTRAN IV-PLUS compiler test

To run the system services test, issue the following command:

```
$ RUN UETNATV01
```

To run the native mode utility test, issue the following command:

```
$ @UETNATV02 [/OUTPUT=filespec][utility]
```

where utility is one of the following values:

```
ALL     (the default)
DEBUG
PATCH
```

To run the VAX-11 RMS test, issue the following commands:

```
[$ MAGTAP:==device-name:]
$ @UETNRMS00
```

Note that the RMS test cannot include magnetic tape tests unless you explicitly define the symbol MAGTAP as shown above.

To run the VAX-11 FORTRAN IV-PLUS compiler test, issue the following command:

```
$ @UETFORT00
```

A.4.3 The System Load Test

To run the system load test, issue the following commands:

```
$ DEFINE/GROUP UETP$USERS n
$ RUN UETLOAD01
```

Note that you must define the group logical name UETP$USERS when you run the load test separately. See Section 2.2.2 and Table 2-2 for further information.

A.4.4 The Compatibility Mode Test

To run the compatibility mode test issue the following commands:

```
[$ MAGTAP:==device-name:]
$ @UETCOMP00 [/OUTPUT=filespec] [utility]
```

where utility is one of the following values:

ALL (the default)	PIP
DMP	SLP
FLX	SOS
LBR	SRT
PAT	VFY

Note that you must define the global symbol MAGTAP to include the FLX test in the compatibility mode test phase; see Section 2.4.4.

INDEX

P

Q

R

S

T

U

V

VAX/VMS
UETP User's Guide
AA-D643A-TE

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 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

Please cut along this line.

— — — Do Not Tear - Fold Here and Tape —

digital

No Postage
Necessary
if Mailed in the
United States

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO.33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

RT/C SOFTWARE PUBLICATIONS TW/A14
DIGITAL EQUIPMENT CORPORATION
1925 ANDOVER STREET
TEWKSBURY, MASSACHUSETTS 01876

— — Do Not Tear - Fold Here —