September 1977

This document describes the installation and operation of the User Environment Test Package distributed with RSTS/E systems.

RSTS/E
User Environment Test Package

Order No. AA-5704A-TC

SUPERSESSION/UPDATE INFORMATION:	This is a new document.
OPERATING SYSTEM AND VERSION:	RSTS/E V06B-02
SOFTWARE VERSION:	V06B-03A

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, September 1977

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

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

Digital Equipment Corporation assumes no responsibility for the use or reliability of its software on equipment that is not supplied by DIGITAL.

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
DEC
PDP
DECUS
UNIBUS
COMPUTER LABS
COMTEX
DDT
DECCOMM

DECsystem-10
DECtape
DIBOL
EDUSYSTEM
FLIP CHIP
FOCAL
INDAC
LAB-8
DECSYSTEM-20

MASSBUS
OMNIBUS
OS/8
PHA
RSTS
RSX
TYPESET-8
TYPESET-10
TYPESET-11

CONTENTS

CONTENTS (CONT.)

PREFACE

The User Environment Test Package (UETP) exercises RSTS/E V06B-02 system hardware in a time sharing environment. The package is designed for use in the DIGITAL manufacturing plant. The system manager at a local installation can run UETP to find malfunctioning hardware or to isolate causes of hardware errors on the system. The UETP package can also be used to impose a known, repeatable load on the RSTS/E system.

This manual describes the procedures to install the UETP package on the system, the ways to operate UETP, and the functions of various UETP components. The system manager should refer to this manual when he generates the RSTS/E system and when he needs to test hardware components.

The manual refers to the monitor statistics option in RSTS/E. This option, described in Appendix E of the RSTS/E System Generation Manual, is a special SYSGEN feature to assist DIGITAL personnel with performance analysis. Because the statistics code is not supported software and adds storage and processing overhead to RSTS/E, DIGITAL recommends that it not be included on user systems. If the statistics code is present, however, certain UETP routines can manipulate the statistical data collected.

For more information on RSTS/E documentation for standard and optional software, consult the RSTS/E Documentation Directory. For information on hardware supplied with RSTS/E, consult the user documents that accompanied the system.

CHAPTER 1

UETP INSTALLATION

1.1 INTRODUCTION

The RSTS/E User Environment Test Package (UETP) provides a System Manager with an automated means to verify the reliability of the RSTS/E hardware. UETP consists of device exerciser programs, simulated user programs, programs that simulate user keyboard input, and suitable control programs to allow unattended operation.

UETP is written in BASIC-PLUS and may be incorporated into the user's production system; UETP can also be built on a special system to be used by the System Manager whenever hardware problems are suspected.

This manual describes how to build and execute UETP on existing RSTS/E systems and documents system generation options required for successful operation of UETP. It is assumed that the user is familiar with the RSTS/E System Generation Manual and the RSTS/E System Manager's Guide.

1.2 RSTS/E UETP HARDWARE USAGE TABLE

UETP can test the following hardware:

DEVICE	HARDWARE TESTED	COMMENTS
MEMORY	YES	
RK05	YES	UETP exercises all units, including unit 0
RK06	YES	Same as RK05
RP02/RP03	YES	Same as RK05
RP04/RP05/RP06	YES	Same as RK05
RS03/RS04	YES	Same as RK05
RF/RS11	YES	Same as RK05
RC11/RS64	NO	Used only as RSTS/E swapping device
RX11/RX01	YES	Same as RK05
TM02/TU16	YES	Same as RK05

TM11/TU10	YES	Same as RK05
TM02/TU45	YES	Same as RK05
DECtape	YES	Same as RK05
LINE PRINTER	NO	UETP uses LP0: for log output
CARD READER	YES	
P.T. PUNCH	YES	
P.T. READER	YES	
DH11,DJ11,DZ11, KL11,LC11,DL11A,DL11-B, DL11-C,DL11-D,DC11,DL11-E	YES	Used with multiple keyboards

1.3 GETTING RSTS/E UETP ON THE AIR

This section describes how to generate a system that is to use UETP, how to build the UETP files, and how to start a system when UETP is to run, if the normal startup procedures are not available or cannot be used.

1.3.1 Incorporating UETP into a System

To incorporate UETP into a RSTS/E system (production or stand-alone) you must be aware of UETP's requirements during system generation.

1. If UETP is to run stand-alone (with no user programs running), the system must be generated for a minimum of 15 jobs. If run together with other production, UETP requires at least 10 additional jobs for full testing.

   STAND-ALONE: 15 jobs

2. To run the UETP package, the swap maximum must be at least 16K words.

   STAND-ALONE: 16K SWAP MAX

3. UETP controls the exercisers by using the pseudo keyboard facility of RSTS/E. The system must be generated for a minimum of 10 PK: units (one for Batch, eight for the exercisers, and one for the exerciser manager).

   Note that if pseudo keyboards have already been generated into the system (for use by the Batch program), the nine additional units will add a maximum of 288 (decimal) words to the monitor area.

   STAND-ALONE: 10 PK: units

4. It is not necessary to configure BASIC-PLUS for any special features (mathematical functions, the MAT statement, and PRINT-USING). However, if functions are not present, the CPU exerciser will not run; if the MAT statement is not present, one of the user-environment scripts will not run correctly; and if PRINT-USING is not present, the program that prints

monitor statistics (see below) will not run.

STAND-ALONE: 2-word math package
Functions and Matrix commands
PRINT-USING

5. A special monitor statistics option may be configured into the system during system generation. This is described in Appendix G of the RSTS/E System Generation Manual. This option adds about 300 (decimal) words to the monitor for code and job statistics buffers, plus 50 (decimal) words for each disk unit. Also, PRINT-USING must be configured in BASIC-PLUS to print the information.

   STAND-ALONE: Include statistics by answering the LONG OR SHORT FORM question with S/Q.

   NOTE

   Monitor statistics were added to the system to aid Software Engineering and Manufacturing in evaluation and improvement of RSTS/E software and hardware. Support for this facility is neither expressed nor implied by this document.

   The release of information in this document does not imply a commitment by DIGITAL to support this facility, to correct errors, or to maintain compatibility in future releases of RSTS/E.

6. If UETP is to exercise fixed-head disks, they must be initialized as file-structured devices, mounted, and contain enough free space for the files to be written on them by the UETP exercisers. It is also possible to implicitly exercise fixed-head disks by using them for swap space.

   STAND-ALONE: Install all swapping space on your system disk. Initialize all disks as file-structured devices.

1.4 CONVENTIONS USED IN THIS DOCUMENT

Throughout this document all responses which are to be typed by the user are underlined as in the following example:

```
     TIME:  12:45
```

All responses are terminated by typing a Carriage Return RET . The Carriage Return is shown in the examples only when it is the only user input.

1.5 BUILDING UETP

This section describes how to install UETP on your system. UETP is installed after the RSTS/E system library have been built. To build and run UETP, you must be logged into the system on a privileged account. For RK05, RK06, and magtape distribution, UETP is on the same medium as the Standard System Library. For DECtape distribution,

UETP INSTALLATION

UETP is supplied on two separate DECtapes. Follow the procedures in the RSTS/E System Generation Manual to mount the proper distribution medium. Note that disks must be mounted read-only. The UETP files are located on the following media:

MT: (7-TRACK)	DEC-11-ORS1A-E-MA7
MT: (9-TRACK)	DEC-11-ORS1A-E-MA9
DK:	DEC-11-ORSLA-E-HA1
DT:	DEC-11-ORSCA-E-UA13 and DEC-11-ORSCA-E-UA14
DM:	DEC-11-ORSLA-E-BA

If the UETP library is on a disk, it is mounted by typing:

```
MOUNT dev:ORSLAE/RO


Ready
```

Where dev: is the device name and number.

ORSLAE is the pack id.

/RO is a switch that guarantees that the pack is mounted read-only.

Although UETP can be built and run on any privileged account, it is strongly recommended that you establish a unique account for UETP. This is done by running the REACT system program as described in the RSTS/E System Manager's Guide. In this manual, account [1,44] is used for UETP. The account is installed as follows:

```
RUN $REACT

REACT V06B-03 RSTS V06B-02 Timesharing
SYSTEM ACCOUNT MANAGER

FUNCTION? E

PROJ, PROG? 1,44

DISK:PASSWORD? UETP

QUOTA? RET

CLUSTER SIZE? 8

ACCOUNT NAME? UETP

PROJ, PROG? ^Z

Ready
```

Next, run the BUILD program to create the UETP library as described in the RSTS/E System Manager's Guide:

```
RUN $BUILD


BUILD V06B-03 RSTS V06B-02 Timesharing

SYSTEM BUILD? NO


SOURCE INPUT DEVICE?  devn


LIBRARY OUTPUT DEVICE <SY:> ? RET


LIBRARY ACCOUNT <[1,2]>? [1,44]


CONTROL FILE IS? $UETP.CTL
```

```
Where dev is:  MT for TU10 or TS03 magtape
               MM for TU16 or TU45 magtape
               DK for RK05 disk
               DM for RK06 disk
               DT for DECtape

and n is the device number.
```

The response to the LIBRARY ACCOUNT question is the account under which UETP is to be built and run.

The UETP library is now built. Messages are printed at the console showing the actual build procedures. When BUILD is finished the following message is printed:

```
BUILD COMPLETE

Ready
```

NOTE

1. BUILD must be run once for each of the RSTS/E system library DECtapes containing UETP modules. The control file for both DECtapes is $UETP.CTL.

2. If the BASIC-PLUS system does not have the mathematical functions included during system generation, the CPEXER program will not build correctly. If the BASIC-PLUS system does not have the PRINT-USING option included during system generation, the JSTATS program will not build correctly. In each case, the "MISSING SPECIAL FEATURE" error message will be printed.

3. The UETP media must now be dismounted if it was an RK05 or RK06:

```
RUN $UTILTY


UTILTY V06B-03 RSTS V06B-02 Timesharing

#DISMOUNT dev:


#^Z

Ready
```

Where dev: is the device name and number, for example, DK1:

CHAPTER 2

UETP OPERATION

2.1 OPERATION OF UETP

If you are running UETP on your production system, start the system as you normally do and proceed to section 2.3 of this chapter.

2.2 STAND-ALONE SYSTEM STARTUP

This section describes how to start a RSTS/E system for UETP operation without using the START.CTL file.

2.2.1 Initializing the System and Starting the Error Logging Program

The system must first be bootstrapped and started as described in the RSTS/E System Manager's Guide. When timesharing is started, the INIT program runs and requests a control file. You should then stop the program by typing CTRL/C and start the various components of the system from the console terminal. It is recommended that UETP always be run from a hard-copy terminal so that you will have a record of its operation.

```
SYSTEM INITIALIZATION PROGRAM V06B-03

COMMAND FILE NAME?  ^C


RUN $ERRINT


ERRINT V06B-03 RSTS V06B-02 Timesharing
ERRLOG File is 0% Full
Change Size to < 100 >? RET

Utilize Crash File Output (Yes/No) <No>? RET


Detaching

HELLO 1/44


Password:  UETP


Ready
```

2.2.2 Adding Auxiliary Swapping Space

If the system has auxiliary disks which are to be used for swapping storage, you must run UTILTY to mount them and add any additional swapping space. If there are no auxiliary disks, proceed to section 2.2.3 in this chapter to allow programs to login and set terminal characteristics.

```
     RUN $UTILTY


     UTILTY V06B-03 RSTS V06B-02 Timesharing
```

Now, mount all scratch media for each disk device type. Note that file-structured, fixed-head disks must also be mounted. Do not mount fixed-head disks that are used for swapping only (and which have not been initialized with the RSTS/E file structure).

All packs used must have been initialized for use under RSTS/E by using the DSKINT option during system initialization.

```
     #MOUNT dev:label


     #CLEAN dev:


     #UNLOCK dev:
```

where dev: is the device name and number, for example, DK1: and label is the pack ID.

Repeat the MOUNT, CLEAN and UNLOCK commands for each disk except the system disk.

Now, add all auxiliary swapping files. The following example assumes the files are on file-structured disks. If swapping space on nonfile-structured disks is to be used, the same command is used with the exception that no file name is given; i.e., instead of

```
          ADD SWAPFILE 0 dev:SWAP0.SYS
     type ADD SWAPFILE 0 dev:

     #ADD SWAPFILE 0 dev:SWAP0.SYS
```

Where dev: is the device name and number, for example, DS0: and SWAP0.SYS is the file name if needed.

If more than one auxiliary swapping file exists, then type

```
     #ADD SWAPFILE 1 dev:SWAP1.SYS
```

2.2.3 Enabling Logins and Setting Terminal Characteristics

When all swapping space has been added to the system, logins are enabled by running UTILTY with the following commands.

```
RUN $UTILTY

#SET LOGINS 63

#LOGINS

#^Z
```

After enabling logins, set the characteristics of all attached terminals by editing the $TTYSET.SPD file (if necessary) and running the TTYSET program as described in the RSTS/E System Manager's Guide. The following example shows how to define KB21: as an LA36.

```
RUN $TTYSET


TTYSET V06B-03 RSTS V06B-02 Timesharing
Terminal Characteristics Program

?  KB21:


For KB21:? LA36


For KB21:? ^Z
```

Now proceed to Section 2.5 to begin running UETP.

2.3 MOUNT SCRATCH MEDIA

At this point mount all scratch media for each disk device type. Note that fixed-head, file-structured disks must also be mounted.

```
RUN $UTILTY


UTILTY V06B-03 RSTS V06B-02 Timesharing


#MOUNT dev:label


#CLEAN dev:


#UNLOCK dev:
```

Where dev: is the device name and number, for example, DK1: and label is the pack ID.

Repeat the MOUNT, CLEAN and UNLOCK commands for each disk except the system disk.

All packs used must have been initialized for use under RSTS/E by using the DSKINT option during system initialization.

2.4 LOGGING INTO THE UETP ACCOUNT

```
HELLO 1/44

Password:  UETP

Ready
```

2.5 RUNNING UETP

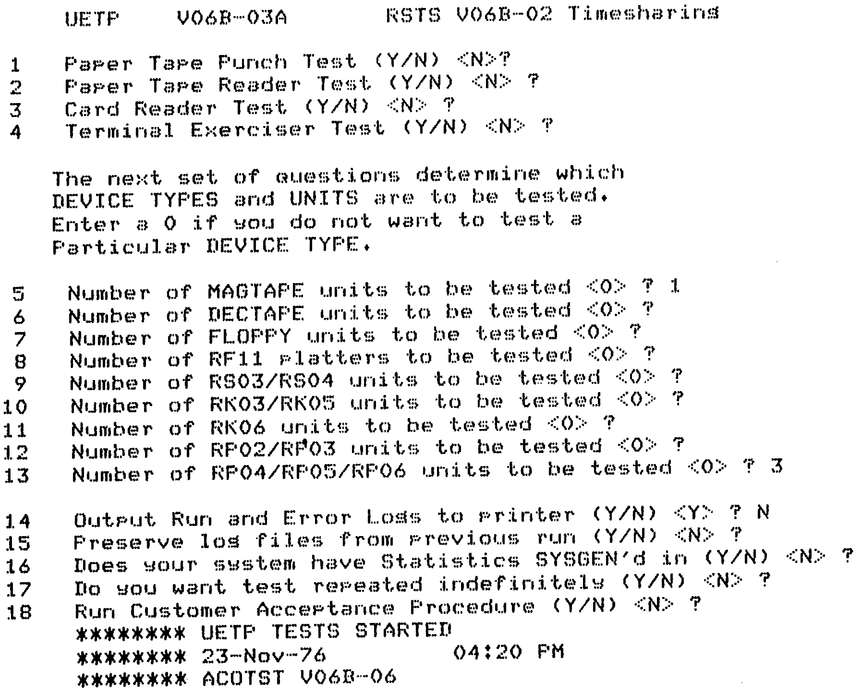
You are now ready to run UETP. UETP is controlled by a series of six programs which chain to each other in a sequence defined by the type of testing desired.

Begin by typing

```
RUN UETP
```

This begins execution of a dialogue program called UETP. Default answers are displayed in angle brackets < > and may be selected by typing the RETURN key in response to the dialogue prompt. The numbers to the left of the questions are for reference purposes only and do not appear in the dialogue.

The example that follows shows a UETP run to test one magtape drive and three RP04 disks. Logs are not printed and the system does not have monitor statistics. To simplify the printout, responses are not underlined, and the default responses (the RETURN key typed) are not explicitly indicated.



```
        UETP     V06B-03A        RSTS V06B-02 Timesharing

  1     Paper Tape Punch Test (Y/N) <N>?
  2     Paper Tape Reader Test (Y/N) <N> ?
  3     Card Reader Test (Y/N) <N> ?
  4     Terminal Exerciser Test (Y/N) <N> ?

       The next set of questions determine which
       DEVICE TYPES and UNITS are to be tested.
       Enter a 0 if you do not want to test a
       particular DEVICE TYPE.

  5     Number of MAGTAPE units to be tested <0> ? 1
  6     Number of DECTAPE units to be tested <0> ?
  7     Number of FLOPPY units to be tested <0> ?
  8     Number of RF11 platters to be tested <0> ?
  9     Number of RS03/RS04 units to be tested <0> ?
 10     Number of RK03/RK05 units to be tested <0> ?
 11     Number of RK06 units to be tested <0> ?
 12     Number of RP02/RP03 units to be tested <0> ?
 13     Number of RP04/RP05/RP06 units to be tested <0> ? 3

 14     Output Run and Error Logs to printer (Y/N) <Y> ? N
 15     Preserve log files from previous run (Y/N) <N> ?
 16     Does your system have Statistics SYSGEN'd in (Y/N) <N> ?
 17     Do you want test repeated indefinitely (Y/N) <N> ?
 18     Run Customer Acceptance Procedure (Y/N) <N> ?
        ******** UETP TESTS STARTED
        ******** 23-Nov-76          04:20 PM
        ******** ACOTST V06B-06
```

A Y answer to any of the first four questions results in execution of AC5TST which requires manual intervention to perform the selected test.

A Y answer to question 4 results in execution of a program which broadcasts test patterns to all terminals on the system. Its use is not recommended if you are running UETP concurrently with timesharing users.

Questions 5-13 should be answered as follows:

Type 0 or the RETURN key if you do not have that device type, or if you do not wish to test that device type. Otherwise, enter the number of units on your system. If you do not wish to test a specific unit number, then you should leave the unit write protected or otherwise unavailable (i.e., off-line). UETP cannot selectively inhibit testing of a specific unit number.

Questions 12,13. UETP cannot test both RP02/03 and RP04/05/06 at the same time. If you have both types of disks, you will have to run UETP twice (once for RP02/03 disks, and once for RP04/05/06 disks).

Alternatively, you can start UETP for one disk type (answering 0 to the other) and run the exerciser programs for the other disk type from another terminal. The programs to run are NEWRP for RP02/03 disks and NEWRB for RP04/05/06 disks. Their operation is described in sections 3.4.11 and 3.4.13 respectively.

Question 14. Answer Y if you have a line printer.

Question 15. Answer Y to append log output from this run to any existing log files.

Question 16. The monitor statistics option is described in Appendix G of the RSTS/E System Generation Manual. If you answer Y and the option is present, job and disk statistics are collected during UETP operation, and summaries are printed in the log. (The printout program requires the PRINT-USING option to be present.) The disk statistics printout may be used together with the error log report to determine whether disk error rates exceed established criteria. The monitor statistics option is not necessary for the operation of UETP.

Question 17. Answer Y to cycle UETP continuously. Answer N to obtain a single cycle (which can exceed 12 hours).

Question 18. Answer Y to execute AC0TST and AC3TST only. This minimal test takes about two hours. See sections 3.2.2 and 3.2.5 respectively for descriptions of AC0TST and AC3TST.

NOTE

> If the UETP questions are answered such that the disk structure consists of two or less RK05s, UETP prints a warning message indicating that there may not be room to run UETP.

2.6 CONSOLE LOG EXAMPLE

Current job statistics and all RSTS/E system errors detected by the individual test programs are printed on the console keyboard. The example below was extracted from a UETP test run. It illustrates the simultaneous execution of four copies of CPEXER.

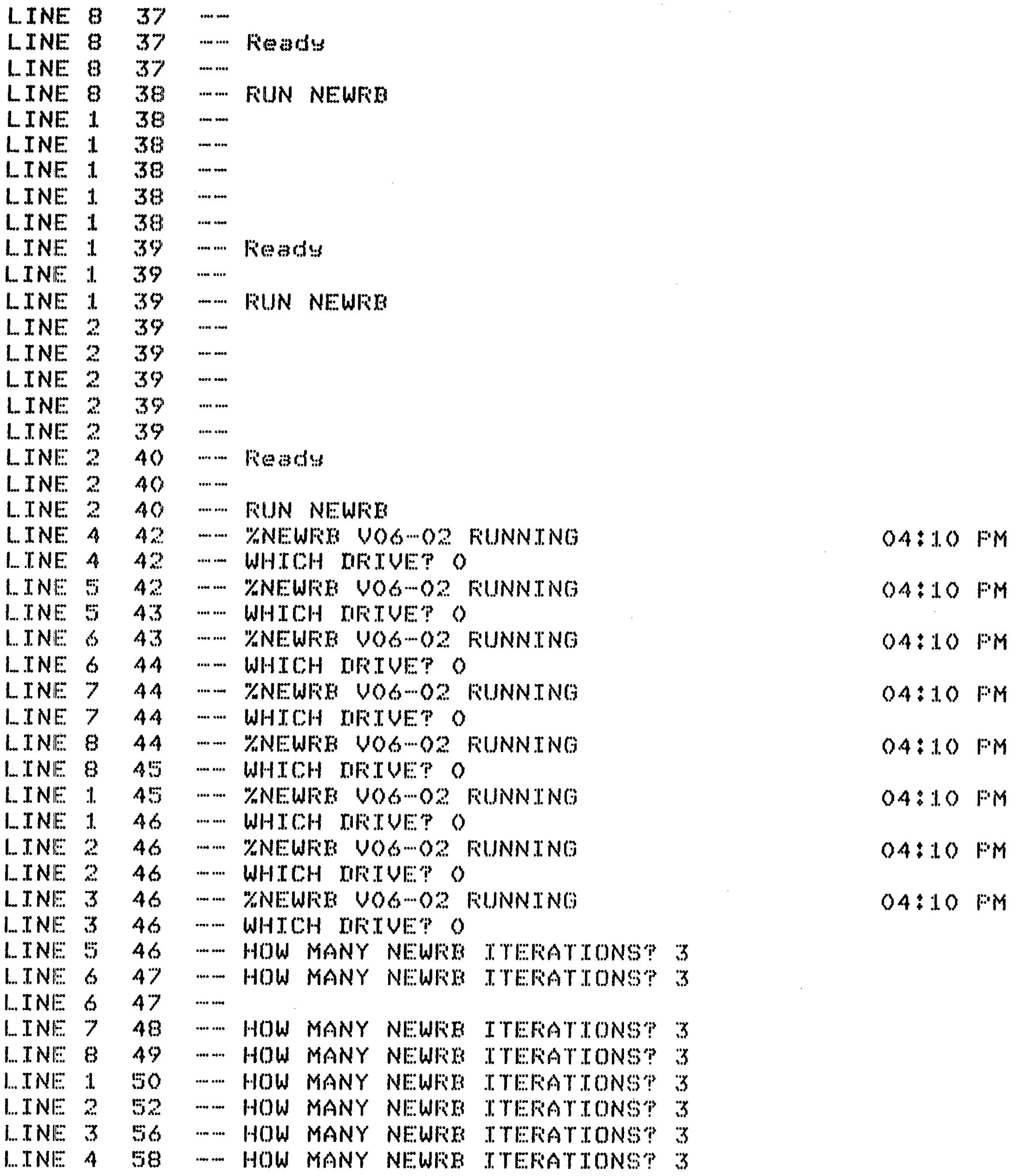
```
********* SYSTEM RELIABILITY TESTS        23-NOV-76    04:24 PM
********* CPU EXERCISER TESTS             04:24 PM
04:24 PM  RUN SCRIPT
04:24 PM  *SY:ACOLOG.LOG<Z5CTL.TMP
04:24 PM  LINE 1 -- CPEXER V06-03 RUNNING     04:24 PM
04:24 PM
04:39 PM  LINE 1 -- CPEXER FINISHED           04:39 PM
04:39 PM
04:39 PM  FROM SCRPTB -- 1 IS FINISHED
04:39 PM  FROM SCRPTB -- ALL CHNLS DONE
04:39 PM
04:39 PM  Ready
04:39 PM
04:39 PM  RUN SCRIPT
04:39 PM  *SY:ACOLOG.LOG<Z6CTL.TMP
04:40 PM  LINE 4 -- CPEXER V06-03 RUNNING              04:40 PM
04:40 PM
04:40 PM  LINE 1 -- CPEXER V06-03 RUNNING              04:40 PM
04:40 PM
04:40 PM  LINE 2 -- CPEXER V06-03 RUNNING              04:40 PM
04:40 PM
04:40 PM  LINE 3 -- CPEXER V06-03 RUNNING              04:40 PM
04:40 PM
04:55 PM  LINE 4 -- CPEXER FINISHED                    04:55 PM
04:55 PM
04:55 PM  FROM SCRPTB -- 4 IS FINISHED
04:55 PM  LINE 3 -- CPEXER FINISHED                    04:55 PM
04:55 PM
04:55 PM  FROM SCRPTB -- 3 IS FINISHED
04:55 PM  LINE 1 -- CPEXER FINISHED                    04:55 PM
04:55 PM
04:55 PM  FROM SCRPTB -- 1 IS FINISHED
04:55 PM  LINE 2 -- CPEXER FINISHED                    04:55 PM
04:55 PM
04:55 PM  FROM SCRPTB -- 2 IS FINISHED
04:55 PM  FROM SCRPTB -- ALL CHNLS DONE
04:55 PM

04:55 PM  Ready
```

2.7 LINE PRINTER OUTPUT

While the tests are running, a log file is written by the UETP system. If question 14 was answered Y or <CR>, the file is written to the line printer (LP0:) during UETP execution. If the question was answered N, log files are written on the public structure in the current account. The files are named ACnTST.LOG (where n is the test component).

The log contains run and error information for all jobs run by the UETP monitor. The third column in the printout contains the number of seconds since a job step was started. The following printout shows a portion of a run when eight copies of the RP04/05/06 exerciser were run on RP04 unit 0.



```
LINE 8  37  --
LINE 8  37  -- Ready
LINE 8  37  --
LINE 8  38  -- RUN NEWRB
LINE 1  38  --
LINE 1  38  --
LINE 1  38  --
LINE 1  38  --
LINE 1  38  --
LINE 1  39  -- Ready
LINE 1  39  --
LINE 1  39  -- RUN NEWRB
LINE 2  39  --
LINE 2  39  --
LINE 2  39  --
LINE 2  39  --
LINE 2  39  --
LINE 2  40  -- Ready
LINE 2  40  --
LINE 2  40  -- RUN NEWRB
LINE 4  42  -- %NEWRB V06-02 RUNNING                      04:10 PM
LINE 4  42  -- WHICH DRIVE? 0
LINE 5  42  -- %NEWRB V06-02 RUNNING                      04:10 PM
LINE 5  43  -- WHICH DRIVE? 0
LINE 6  43  -- %NEWRB V06-02 RUNNING                      04:10 PM
LINE 6  44  -- WHICH DRIVE? 0
LINE 7  44  -- %NEWRB V06-02 RUNNING                      04:10 PM
LINE 7  44  -- WHICH DRIVE? 0
LINE 8  44  -- %NEWRB V06-02 RUNNING                      04:10 PM
LINE 8  45  -- WHICH DRIVE? 0
LINE 1  45  -- %NEWRB V06-02 RUNNING                      04:10 PM
LINE 1  46  -- WHICH DRIVE? 0
LINE 2  46  -- %NEWRB V06-02 RUNNING                      04:10 PM
LINE 2  46  -- WHICH DRIVE? 0
LINE 3  46  -- %NEWRB V06-02 RUNNING                      04:10 PM
LINE 3  46  -- WHICH DRIVE? 0
LINE 5  46  -- HOW MANY NEWRB ITERATIONS? 3
LINE 6  47  -- HOW MANY NEWRB ITERATIONS? 3
LINE 6  47  --
LINE 7  48  -- HOW MANY NEWRB ITERATIONS? 3
LINE 8  49  -- HOW MANY NEWRB ITERATIONS? 3
LINE 1  50  -- HOW MANY NEWRB ITERATIONS? 3
LINE 2  52  -- HOW MANY NEWRB ITERATIONS? 3
LINE 3  56  -- HOW MANY NEWRB ITERATIONS? 3
LINE 4  58  -- HOW MANY NEWRB ITERATIONS? 3
```

The log file also contains the output of the error display program. The UETP control programs automatically dump this report three times during the test run. This log is identical to the log output by the standard RSTS/E ERRDIS program. The logging of errors during UETP and their subsequent analysis is the key towards using UETP to verify system integrity. The following printout shows a portion of a UETP log file, including an error log printout.

```
LINE 1  1  -- HELLO1/44;UETP
LINE 1  2  -- Please say HELLO
LINE 1  2  --
LINE 1  2  -- Bye
LINE 1  2  --
LINE 1  2  -- RUN $ERRDIS
LINE 1  6  --
LINE 1  6  -- Bye
LINE 1  6  --
LINE 1  6  --
LINE 1  7  --
LINE 1  7  --
LINE 1  7  --
LINE 1  7  --
LINE 1  7  --
LINE 1  7  --
LINE 1  7  -- Ready
LINE 1  7  --
LINE 1  7  --
LINE 1  8  -- ERRDIS V06B-03 RSTS V06B-02 Timesharing RJE
LINE 1  8  --
LINE 1  8  -- Input File <$ERRLOG.FIL>?
LINE 1  8  -- Output to <KB:ERRDIS.OUT>?
LINE 1  8  -- He[lp], Ba[d Blocks], Su[mmary] or Fu[ll] Report <summary>?
LINE 1  8  -- List Bad Blocks (Yes/No) <Yes>?
LINE 1  8  -- Zero Error File upon completion (Yes/No) <No>?
LINE 1  9  --
LINE 1  9  -- ERRDIS Summary Report taken on 27-May-77, 09:16AM
LINE 1  9  -- Input File: $ERRLOG.FIL         Output File: KB:ERRDIS.OUT
LINE 1  9  -- ^Z
LINE 1  9  --
LINE 1  11  --
LINE 1  11  -- Reported Date/Time Range:
LINE 1  11  --         26-May-77, 12:24:27 AM through 27-May-77, 08:32:10 AM
LINE 1  11  --
LINE 1  11  --       ERROR            TOTAL                 UNIT NUMBERS
LINE 1  11  -- CODE-DESCRIPTION      REC/LOG     0    1    2    3    4    5    6    7
LINE 1  11  --
LINE 1  11  -- T4 Trap to 4 (CPU)      4/4
LINE 1  16  -- PF PowerFail/Strtup     1/1
LINE 1  16  -- KT Memory Mgmt.         3/3
LINE 1  16  -- DK RK11/RK05            2/2            2
LINE 1  16  -- DB RH11/RP04-05-06     20/20      1         19
LINE 1  16  -- MM RH11/TM02/TU16      38/38     34    4
LINE 1  19  --
LINE 1  19  -- Total of 68 Errors Logged out of 68 Received
LINE 1  21  --
LINE 1  21  -- 20 out of 100 Blocks have been used in $ERRLOG.FIL
LINE 1  21  --
LINE 1  21  --
LINE 1  21  --
LINE 1  21  --
LINE 1  21  --
LINE 1  21  -- List of Possible Bad Blocks
LINE 1  21  -- ---------------------------
LINE 1  26  --
LINE 1  26  -- None Found
LINE 1  26  --
LINE 1  26  --
LINE 1  26  --
LINE 1  26  -- Input File <$ERRLOG.FIL>?
LINE 1  26  -- Ready
LINE 1  26  --
LINE 1  26  !! (END OF RUN)
LINE 1  26  -- ^C
LINE 1  26  -- BYE/F
LINE 1  36  --
LINE 1  36  -- Ready
```

2.8 TERMINATING UETP

When UETP has completed a pass the following message is printed on the console:

```
**************************************************
**************************************************
**************************************************

********* ACCTST PASS COMPLETE   15-OCT-76   15:25

**************************************************
**************************************************
**************************************************
```

If you did not request indefinite repetition of the test (that is, you answered question 17 N), UETP stops and the RSTS/E monitor prints Ready. If UETP is running continuously, or you must abort the run, the proper procedure is as follows:

1. Type ^C to abort UETP. The RSTS/E monitor will print Ready.

2. Follow the instructions in RSTS/E System Manager's Guide.

   a. Type "END" to close all files.

   b. Run the SHUTUP program from the console terminal to stop the system. Shutting down the system automatically stops any exercisers.

      NOTE

      Use the UTILTY KILL command as an alternate procedure to stop the jobs (See UTILTY KILL in the RSTS/E System Manager's Guide).

   c. Restart the system and log into the system under the UETP account.

Now, run ERRDIS to preserve a copy of the error log.

```
RUN $ERRDIS

ERRDIS V06B-03 RSTS V06B-02 Timesharing
Input File <$ERRLOG.FIL>? RET

Output To <KB:ERRDIS.OUT>? RET

He[lp], Ba[d Blocks], Su[mmary] or Fu[ll] Report <SUMMARY>? RET

List Bad Blocks (Yes/No) <YES>? RET

Zero Error File upon completion (Yes/No) <NO>? RET

ERRDIS Summary Report taken on 15-Oct-76, 12:19 PM
```

(example continued on next page)

```
Input File:  $ERRLOG.FIL        Output File:  KB:ERRDIS.OUT
Reported Date/Time Range:
        14-Oct-76, 03:03:01 PM through 15-Oct-76, 10:56:07 PM

       ERROR              TOTAL                    UNIT NUMBERS
CODE-DESCRIPTION         REC/LOG   0   1   2   3   4   5   6   7

PF Powerfail/Strtup        1/1
DB RH11/RP04-05-06         1/1                         1
MM RH11/TM02/TU16          5/5     3   2

Total of 7 Errors Logged out of 7 Received

1 out of 100 Blocks have been used in $ERRLOG.FIL


List of Possible Bad Blocks


None Found


Input File <$ERRLOG.FIL>? RET


Output File <KB:ERRDIS.OUT>? RET


He[lp], Ba[d Blocks], Su[mmary] or Fu[ll] Report <SUMMARY>? FULL


Specific Error Type <ALL>? RET


Starting Date <First Error>? RET


Ending Date <Last Error>? RET


Zero Error File upon completion (Yes/No) <No>? RET
```

A full report will be printed on the keyboard.

```
Input File <$ERRLOG.FIL>?  ^Z


Ready
```

If the UETP log files were directed to the disk, proceed as follows to preserve the log files.

```
RUN $PIP                    - run the PIP system program


PIP V06B-03 RSTS V06B-02 Timesharing


#xx:=AC0LOG.LOG             - where xx is the output device


#xx:=AC1LOG.LOG             - where xx is the output device
```

```
#xx:=AC2LOG.LOG           - where xx is the output device

#xx:=AC3LOG.LOG           - where xx is the output device

#xx:=AC5LOG.LOG           - where xx is the output device
```

The UETP run is now complete. The system may be shut down by following the procedures described in the RSTS/E System Manager's Guide.

CHAPTER 3

UETP COMPONENTS

3.1 UETP COMPONENT DESCRIPTIONS

The UETP package uses the RSTS/E pseudo keyboard mechanism to control and monitor test programs. At the highest level (real attached keyboard), a set of control programs is executed one at a time. Each of the control programs causes certain test programs to be executed, either directly or through a script program.

Pseudo keyboards are used for communication between the control program and the script program. They are also used by the script program to control the test programs. Because at least 10 pseudo keyboards should be configured on every system that is to run UETP, a control program can drive the script program which in turn can control up to eight test programs. One extra pseudo keyboard is then available for use by the Batch facility.

Test programs consist of Device Exercisers (section 3.4), Interactive Tests (section 3.5), Supplementary programs (section 3.6), and User Simulation Programs (section 3.7).

In addition to all programs in this section, UETP uses the ERRDIS program contained in the System Library Account [1,2]. ERRDIS is described in the RSTS/E System Manager's Guide.

3.2 UETP CONTROL PROGRAMS

3.2.1 UETP

This program is run by the user to initiate a test run. UETP determines which hardware is to be tested and the way the tests are to be managed. It is described in section 2.5.

The major internal tasks are as follows:

1. If this is the initial run, interrogate the user and set up the virtual file PERDTA.DTA.

2. Set up the logs for later use by UETP programs.

3. Chain to AC5TST if paper tape, keyboard and/or card reader tests have been selected.

4. Chain to AC0TST for "normal" run.

3.2.2 AC0TST

This is the second UETP control program. Its function is to run the CPU and RP03/RP04 exerciser tests. When AC0TST finishes, it chains to AC1TST.

The job procedures for AC0TST are as follows:

1. Pick up the peripheral data from the virtual file PERDTA.DTA.

2. Set up to log the subjobs under the current user account.

3. If expanded job statistics have been selected, print the statistics and reset the STATS.DTA statistical matrix.

4. Print the error log and zero it ONLY if the line printer is the error log output device.

5. Run three CPEXER test loops. Each loop consists of one, four or eight copies of CPEXER with each copy running 15 minutes.

6. If RP03's or RP04/05/06's are present, run NEWRP or NEWRB on Drive 0 for two iterations.

7. If RP03's or RP04/05/06's are present, run eight copies NEWRP or NEWRB on all drives with a round-robin selection for two iterations.

8. If expanded job statistics have been selected, print them.

9. Chain to AC1TST.

3.2.3 AC1TST

This is the third UETP control program. It selectively exercises the RX01 disk drives, RK05 disk drives and TU56 DECtape drives, and then exercises multiple RP03/RP04, RK05, RF11, RS03/04, RX01, and TU56 drives. When AC1TST finishes, it chains to AC2TST.

The job procedures for AC1TST are as follows:

1. Get the peripheral data from the virtual file PERDTA.DTA.

2. Set up to log the subjobs under the current user account.

3. If RX01 disks are present, run DXEXER on drive 0 for two iterations, then run DXEXER on all drives for three iterations.

4. If RK05 disks are present, run DKEXER on Drive 0 for four iterations.

5. If RF or RS disks are present, run DFEXER or DSEXER on each drive for 4 iterations.

6. If RK05 disks are present, run eight copies of DKEXER on all drives with a round-robin selection for four iterations. Allow no drive to have more than four copies selected. Drive 0 has a maximum of two copies.

7. If DECtape is present, run DTEXER on TU56, Drive 0, then run DTEXER on all TU56 drives.

8. If the appropriate devices are present, run the RP03/RP04/RP05/RP06, RK05, RX01, and TU56 exercisers on each drive, up to a maximum of two exercisers per device.

9. If RK06 devices are present, run eight copies of DMEXER on RK06 drive 0.

10. If expanded job statistics have been selected, print them.

11. Chain to AC2TST.

3.2.4 AC2TST

This is the fourth UETP control program. Its function is to test the mechanical and data reliability of magtape and to exercise all devices simultaneously. WHEN AC2TST finishes, it chains to AC3TST.

The job procedures for AC2TST are as follows:

1. Pick up peripheral data from the virtual file PERDTA.DTA.

2. Set up to log the subjobs under the current user account.

3. Print the error log.

4. If magtape is present, run MTEXER on all drives for 40 iterations, with two feet of tape. This is the mechanical reliability test.

5. If magtape is present, run MTEXER on Drive 0 for 500 feet of tape, then run all drives for two iterations with 500 feet of tape. This is the data reliability test.

6. Run the "ALL DEVICES TOGETHER" job step. The algorithm for device selection is as follows:

   a. Allow testing of at least one unit of each device on the system.

   b. The units are selected using the following priority: DP/DB, RK, RX, MT, DT.

   c. If eight units cannot be selected then the remaining units will comprise DP/DB, RK and CP jobs in that order.

   d. If the system disk is the RK05, do not permit more than three jobs on unit 0.

7. If expanded job statistics have been selected, print them.

8. Chain to AC3TST.

3.2.5 AC3TST

This is the fifth UETP control program. Its function is to simulate a timesharing system by executing a series of scripts which simulate users interacting with RSTS/E. A series of user application packages is also run which will also simulate the user environment. If you selected continuous test in the initial dialogue, AC3TST chains to AC0TST when it finishes. If you did not select continuous test, UETP terminates.

The job procedures for AC3TST are as follows:

1. Pick up the peripheral data from the virtual file PERDTA.DTA.
2. Set up to log the subjobs under the current user account.
3. Execute RANMAK. This program creates an inventory file.
4. Execute CVTSCP to convert the raw scripts to the current account.
5. Execute the following scripts: VIRSTR, CPU, EDIT, CLUMSY, RANDAC, TTY, FILES, and IMMED.
6. Execute the following programs that create files to be used later: DA, CRFILE, and UDA.
7. If magtape is present, execute TAPSRT - magtape sort program.
8. If floppy disks are present, execute FLOPPY - sort/merge program.
9. If DECtape is present, execute DECMRG - DECtape sort/merge program.
10. Execute CPUTST, FILMIN and TDLSRT.
11. Execute VERIFY to verify the data.
12. Delete the created data files.
13. Print the error log and, if selected, the expanded job statistics.
14. If continuous running was selected, chain to AC0TST; otherwise kill the PERDTA.DTA and STATS.DTA files and terminate.

3.2.6 AC4TST

This program is used by Digital Equipment Corporation Manufacturing only.

3.2.7 AC5TST

This is the sixth UETP control program. Its function is to run the tests on card readers, paper tape equipment and keyboards. The job procedures are as follows for AC5TST:

1. Pick up the peripheral data from the virtual file PERDTA.DTA.
2. Run the keyboard exerciser test if selected.
3. Set up to log the subjobs under the current user account.
4. If paper tape punch is selected, execute PPEXER.
5. If paper tape reader is selected, execute PREXER.
6. If a card reader is selected, execute CREXER.
7. Chain to AC0TST if selected; otherwise, terminate UETP.

3.3 SCRIPT PROGRAMS - SCRIPT AND SCRPTB

The SCRIPT program reads one or more script files and compiles them into intermediate code. SCRPTB then executes the code, which sends simulated TTY input to the test programs being executed and accepts the keyboard output from the test program lines. If so directed, SCRPTB records this simulated TTY traffic in a log file. The SCRPTB program runs as one job on a timesharing system, supplying inputs for up to eight other jobs.

3.4 UETP DEVICE EXERCISERS

The UETP system includes programs to exercise all RSTS/E hardware devices.

3.4.1 CPEXER

The central processor exerciser is designed to put a heavy load on the central processor. CPEXER runs compute bound for short bursts and then sleeps for five seconds so that average processor loading is not sufficient to appreciably affect the performance of the system. CPEXER was specifically designed to test the FPP option, but also serves to verify general CPU integrity and correct PDP 11/40 EIS operation. There is also a test of the PEEK function for kernel addresses 0 thru 25000.

The tests performed by CPEXER are described briefly below. In most of the tests, results are compared to known correct values. Two "grind" tests (heavily compute-bound) are also included to verify consistent results of duplicate calculations. Finally, tests are for verifying FPP divide by zero trap, integer divide by zero trap, integer conversion error, and EIS operation. These tests have detected hardware failures in the past.

CPEXER test descriptions:

1. SIN(X) - Uses SIN(X) extended function.
2. SIN(X) - Uses polynomial approximation to SINE function.
3. LOG(X) - Uses LOG(X) extended function.

4. EXP(X) - Uses EXP(X) extended function.

5. SQR(X) - Uses SQR(X) extended function.

6. SQR(X) - Uses Newton-Raphson method to determine square root.

7. LOG(EXP(X)) - Grind test.

8. ATN(TAN(X)) - Grind test.

9. A=1.0/0.0 - Verify FPP divide by zero trap.

10. A%=1%/0% - Verify integer divide by zero trap.

11. A%=60000. - Verify integer conversion error.

12. INT(40.6621*100+0.5)/100==
40.66 - Verify EIS operation.

3.4.2 MTEXER

The magtape exerciser checks normal operation of the TM11 magtape control or RH11/TM02 controller and up to eight 7-track or 9-track TU10, TS03, TU16, or TU45 drives. MTEXER allows the operator to select the drive to be tested, the length of tape to be written, and the number of iterations to be performed. On each iteration, MTEXER zeroes the tape, opens a file, and writes data until the specified length of tape has been used. MTEXER then rewinds the tape, opens the file for input, and reads and verifies the data. If errors are detected, MTEXER prints a count of the number of bytes found to be incorrect before proceeding to the next iteration.

The data pattern used is a worst case NRZ pattern for 9-track drives. This pattern is not worst case for 7-track recording. The pattern is loaded into a 512-byte buffer and X PUT's are used to write the tape. The variable X is equal to the repetition number so that X identical records are written on repetition X. The pattern buffer is then changed and the process continues until the required length of tape has been written. Since the number of PUT's increases, tape speed increases on each successive iteration. Furthermore, the pattern base varies with X so that the contents of the pattern buffer also varies on successive iterations.

3.4.3 DTEXER

The DECtape exerciser tests the normal operation of the TC11 DECtape control and up to eight TU56 DECtape drives. DTEXER begins by opening a file on the drive being tested and fills the file with floating point numbers. Out of a possible 578 tape blocks, 420 blocks are written.

Numbers written on the tape are read and checked, keeping a count of incorrect values. If at the end of the test the error count is not zero, DTEXER prints the error count.

3.4.4 DKEXER

This disk exerciser tests the normal operation of the RK11 disk controller and up to eight RK03/RK05 disk cartridge drives. It is possible to run several copies of DKEXER in order to test several drives simultaneously or to put a heavier load on any single drive.

DKEXER begins by asking several questions to determine the drive number and number of test iterations to be performed. After this dialogue, the exerciser opens and extends a file to the pre-determined size. A pattern buffer is then loaded with one of four patterns (all zeroes, all ones, 125252, and 52525) and the file is written. Each block is then read and compared. This procedure is repeated for each pattern. Upon completion of all iterations for a drive, a status report is printed.

3.4.5 DMEXER

This program is similar to DKEXER except that it is for RK06's.

3.4.6 DFEXER

This program is similar to DKEXER except that it is for file structured RF11 disks.

3.4.7 DSEXER

This program is similar to DKEXER except that it is for file structured RS03/04 disks.

3.4.8 CREXER

This program tests the normal operation of the CD11 or CR11 card reader using a fixed card deck (MAINDEC-89-D181-C labeled Alpha Card Deck).

3.4.9 PPEXER

This program tests the normal operation of the paper tape punch.

3.4.10 PREXER

This program tests the normal operation of the paper tape reader by reading the tape punched by PPEXER.

3.4.11 NEWRP

This disk exerciser tests the normal operation of the RP11C disk controller and up to eight RP02 or RP03 disk pack drives. NEWRP tests maximum read/write data transfers on the selected RP02 or RP03 drive.

NEWRP uses very little processor time but causes considerable activity on the disk pack under test and on the UNIBUS. This is accomplished by PUTing and GETing 1000 records on a temporary disk file. Then, starting with the first record in the file, each record is rewritten. NEWRP performs these operations until all iterations are completed. NEWRP then closes and kills the temporary file.

NEWRP never checks actual disk data because the main object of the exerciser is to cause maximum data transfers and maximum UNIBUS switching.

3.4.12 NEWRB

This disk exerciser tests the normal operation of the RH11 disk controller and up to eight RP04, RP05, or RP06 disk pack drives. NEWRB tests maximum read/write data transfers on the selected RP04, RP05, or RP06 drive.

NEWRB uses very little processor time but causes considerable activity on the disk pack under test and the UNIBUS. This is accomplished by PUTing and GETing 1500 records on a temporary disk file. Then, each record is rewritten starting with the first record in the file.

NEWRB performs these operations until all iterations are completed. NEWRB then closes and kills the temporary file.

NEWRB never checks actual disk data because the main object of the exerciser is to cause maximum data transfers and maximum UNIBUS switching.

3.4.13 DXEXER

This disk exerciser tests normal operation of the RX11 disk controller and up to eight RX01 Floppy disk drives. A file consisting of floating point numbers is written on the drive under test then read back and checked. Out of a possible 494 blocks, 420 are written and checked. The program also displays the number of words read and written for error rate decisions.

3.5 UETP INTERACTIVE TESTS

Scripts are written to simulate the following interactive tasks. The scripts may be run in any combination, with any number of copies of each.

3.5.1 RANDAC.SCP

This script simulates random access to disk records. After a file is built, simulated users update random records using GET/PUT statements. Parameters to the script include the size of the file and the logical record size.

3.5.2 VIRSTR.SCP

This script builds a virtual array. A program then extracts and prints strings from the array according to requests from a simulated user. Strings are selected randomly by the script.

3.5.3 EDIT.SCP

This script simulates a user editing a program and compiling it. The simulated user types in a simple BASIC-PLUS program, runs it, gets an error, edits in a correction, saves the new version, compiles and reruns the program. The file is then deleted.

3.5.4 CLUMSY.SCP

This script simulates a rather clumsy programmer who types in a program, getting many error messages. Each error is corrected, and execution is attempted after typing in each change.

3.5.5 CPU.SCP

This exerciser exercises the MAT commands. A program is typed in by the user which builds a matrix which is inverted and printed. Several matrices are inverted, including singular and almost singular examples.

3.5.6 FILES.SCP

This script exercises the file system by creating and copying files within the current account. Files are created, appended, and deleted.

3.5.7 TTY.SCP

A program that generates heavy terminal output is run.

3.5.8 IMMED.SCP

This script performs simple calculations in direct execution mode. User executes commands to print results of simple calculations such as SQRT and LOG. All BASIC-PLUS functions are exercised.

3.6 UETP SUPPLEMENTARY PROGRAMS

The UETP system requires several support programs for its successful operation. These are described in this section.

3.6.1 CVTSCP

This program edits the interactive scripts to run under the UETP account.

3.6.2 DSTATS (requires The Monitor Statistics Option)

DSTATS generates an expanded breakdown of the monitor disk statistics.

3.6.3 JSTATS (requires The Monitor Statistics Option)

Using the virtual file "STATS.DTA" as a base, JSTATS generates an expanded breakdown of the monitor kept job statistics. Statistical categories are:

USER RUN-TIME
NULL TIME
FIP
SYSTEM TICKS
EXEC TIME
CACHE HIT

3.6.4 ERRDP2 (requires The Monitor Statistics Option)

This program is a copy of the DSTATS program modified to allow certain output lines to appear on the user keyboard.

3.6.5 RANMAK

RANMAK is a random access file creator program. File INVENT.ORY is created for use by the interactive test script RANDAC.SCP.

3.6.6 CRFILE

This program creates a disk file consisting of a specified number of records consisting of letters and numbers interspaced with random separators.

3.6.7 DA, UDA

These two programs create files that are used by the TDLSRT sort programs.

3.6.8 VERIFY

This program checks the output files of the TDLSRT sort programs to make sure they are in the proper order. VERIFY checks ten key files and returns check status to the terminal device.

3.7 USER SIMULATION PROGRAMS

These programs simulate a commercial application including a variety of sorting procedures. The programs sort files on disk, magtape, DECtape, and floppy disk (depending on what is available) and execute a CPU-bound "background" program.

3.7.1 Disk Sort - (TDLMQK, TDLOQK, TDLXQK, TDLSQK, and TDLSRT)

The disk sort run starts with the creation of three files using the supplied BASIC-PLUS programs (DA and UDA). The next procedure is to run the sort program with an indirect command file, called RELEAS.SRT. The sort program then proceeds to run through its specific modules with 27 program calls being executed. At the end of the sort checkout the program VERIFY is executed.

3.7.2 Magtape Sort - (TAPSRT and TAPSRU)

The magtape sort programs sort a given input file which is created by the BASIC-PLUS program CRFILE. The tape sort works on a string basis. It orders (sorts) these strings into fewer and longer strings until the file is one sorted string. After the file is sorted, the output file is checked, word for word, with a known output file MTSORT.ADT.

3.7.3 Floppy Disk Sort/Merge - (FLOPPY)

This program starts by checking the device configuration and setting up a file for each unit. This job merges one file into another and checks the results. The program also displays the number of words read and written for error rate determinations.

3.7.4 Disk File Manipulation - (FILMIN)

This program can create many disk requests by defining virtual arrays and using these arrays as input and output to a file creation and verification section. The file manipulation includes many disk requests for data and many "opens" and "closes" for maximum file handling checkout. The program transposes five 40-by-40 matrices back and forth and checks the results after each pass. These matrices are virtual arrays.

3.7.5 Disk/Tape Merge - (DECMRG)

This program starts by checking the device configuration file and using its contents for the job set up. This job merges one file into another. The objective is to check the capability of the UNIBUS logic to switch from device to device.

3.7.6 The Compute Bound Program - (CPUTST)

This program requests the number of wall clock minutes desired for run time (number of minutes desired). The program checks for the expiration of this requested time at strategic points during a normal run sequence (8 checkpoints).

CPUTST clears and/or sets three 2-dimensional arrays to specified values. Matrix A is filled with random numbers using the BASIC-PLUS random number function. Matrix A is then transposed and the results placed in Matrix B. The transposition is checked by the equation/statement A(J,I) = B(I,J) with I and J being varied. Any errors are reported. If no error, the program goes back to the matrix clear instructions. When the time expires, the program prints the CPU seconds expended for this job.

INDEX

RSTS/E
User Environment
Test Package
AA-5704A-TC

READER'S COMMENTS

NOTE: This form is for document comments only. DIGITAL will use comments submitted on this form at the company's discretion. Problems with software should be reported on a Software Performance Report (SPR) form. If you require a written reply and are eligible to receive one under SPR service, submit your comments on an SPR form.

Did you find errors in this manual? If so, specify by page.

____
____
____
____
____
____

Did you find this manual understandable, usable, and well-organized? Please make suggestions for improvement.

____
____
____
____
____
____

Is there sufficient documentation on associated system programs required for use of the software described in this manual? If not, what material is missing and where should it be placed?

____
____
____
____
____
____

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
- □ Non-programmer interested in computer concepts and capabilities

Name ______________________ Date ______________

Organization ______________________________________

Street ____

City ______________ State ______________ Zip Code or Country ______________

Please cut along this line.

Fold Here

Do Not Tear - Fold Here and Staple

FIRST CLASS
PERMIT NO. 33
MAYNARD, MASS.

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES

Postage will be paid by:

digital

Software Documentation
146 Main Street ML5-5/E39
Maynard, Massachusetts 01754