RSTS/E

USER ENVIRONMENT TEST

MD-11-DBZMA-C

PACKAGE (UETP)

EP-DBZMA-C-DL

FICHE 1 OF 2

MAY 1978

digital

MADE IN USA

RSTS/E

USER ENVIRONMENT TEST

MD-11-DBZMA-C

PACKAGE (UETP)

EP-DBZMA-C-DL

FICHE 2 OF 2

MAY 1978

digital

MADE IN USA

IDENTIFICATION

Product Code: MAINDEC-11-DRZMA-C-D

Product Name: RSTS/E User Environment Test Package ("ETP)

Reference: Mast Appendix E
Release #V06B-06
(RSTS/E Monitor V06B)

Date: December 1976

Maintainer: Software Quality Management

PREFACE

The UETP (User Environmental Test Package) was designed for use in the Final Assembly and Test area of manufacturing in DEC. Support for special undocumented RSTS/E features in the UETP is neither expressed nor implied in this document. These special options are needed for performance evaluation in FA+T. Digital assumes no responsibility for the use or support of these features.

Table of Contents

Table of Contents (Con't)

```
E3350          SCRIPT-SCRPTR
F3400       ACCTST Device Exercisers
F3410          CPEXER
F3420          MTEXER
F3430          DTEXER
F3440          DKEXER
F3441          DMEXER
F3442          DFEXER
F3444          DSEXER
F3446          CREXER
F3447          PPEXER
F3448          PREXER
F3450          NEWRP
F3460          NEWRR
F3470          DXEXER
```

Table of Contents (Con't)

Table of Contents (Con't)
****

E0000 OVERVIEW

This test procedure is a system software exerciser routine based on the RSTS/E operating system. This procedure is applicable against those systems capable of operating RSTS/E (ref. E1100) and having completed the prerequisite actions defined in section E1000.

The user is led through the system bootstrap, system generation and simulated user environment system operation (ACCTST). Valid and erroneous system responses are defined at each stage of the procedure.

A system software exerciser checklist has been provided which will lead the user thru the above mentioned steps. A more detailed description of the RSTS/E UETP is given in sections E3000 thru E8400 of this document.

E1000 PRELIMINARY CHECKS AND TASKS

1. Appendix B (DEC-X11) has run without error.

2. Latest RSTS/E SYSGEN (DEC-11-ORSPA-E-MC9 for 9 track or -MC7 for 7 track) and LIBRARY (DEC-11-ORSLA-E-MA9 for 9 track or -MA7 for 7 track) media are available for Monitor V06B-02.

3. Insure that all devices ordered by the customer (Ref. construction req or key sheet) and not designated as field installed (manufacturing only) are physically connected to the system.

4. Insure that the customers distribution RK05 or RK06 has been backed-up (copied) using ROLLIN or an appropriate program (Ref. E6000).

5. For manufacturing use only:

   a. Remove the ACT daughter station and install the terminator.

   b. Insure that all hardware communications options are cabled and ready to run on-line with all turn-arounds removed.

   c. Insure that a general PM of the system is performed.

E1100 MINIMUM SYSTEM RESOURCES

SYSTEM HARDWARE

The hardware system building block can be a PDP-11/45 Central Processor Unit (CPU) with optional hardware floating point processor (FPP) and up to 32K words (K=1024) of MOS memory, a PDP-11/70 CPU with optional FPP, a PDP-11/40 CPU with Extended Instruction Set (EIS) and, optionally, Floating Instruction Set (FIS), or a PDP-11/34 CPU with a minimum of 48K words of memory. The OEM equivalent of these CPU types may also be substituted.

The RSTS/E system is capable of supporting a maximum of 128 terminals. A minimum of 48K words of memory for primary storage is required along with the KT11C/D Memory Management Unit which provides virtual memory expansion up to 124K words of memory. A PDP-11/70 system is capable of supporting up to 2048K words of memory.

In order to provide adequate secondary storage, RSTS/E must include at least two disks from the following types of disks: The RF11 Controller with up to 8 RS11 disks (platters) or an RJS03/RJS04 controller with up to 4 RS03/RS04 disks or an RC11 controller with up to 4 RS64 disk drives and the RK11 Controller with up to 8 RK05 (or RK03) DECpack cartridge disk drives, or at least one disk attached to the RP11-C Controller which can handle up to 8 RP02/RP03 Disk pack drives, or an RJP04 disk system with at least one RP04 disk pack drive or an RP06 disk system or an RK06 disk system. The optimal disk configuration includes a removable moving head disk which acts as the RSTS/E system disk, and an auxiliary fixed head disk which performs swapping operations. Additional disks are used to augment the capacity of public, private and swapping storage space.

The minimum disk configuration required to perform a SYSGEN using RK distribution consists of one RK05 or RK06 removable disk drive and one additional disk of the following type; RK05, RK05F, RK06, RP02/03, RP04/05/06, RS11 with at least 4 platters.

The BM792-YA Hardware Bootstrap Loader, the MR11-DB Bulk Storage Bootstrap Loader, the BM873-YA/YB Bootstrap Loader or the M9301-YA/YB/YC/YF Bootstrap/Diagnostic Loader are required on the system. Also, either the Programmable Real Time Clock KW11-P or the Line Frequency Clock KW11-L is necessary.

Additional devices such as the following can be used on the RSTS/E system. Terminals may be standard Teletypes, LA30 or LA36 DECwriters and VT50, VT52, VT05, or VT05B Alphanumeric Displays. The PC11 High Speed Reader/Punch or the PR11 High Speed Reader can be used to accelerate paper tape input/output operations. Up to eight LP11 and/or LS11 Line Printers can be configured in RSTS/E to make hard copy output more efficient. The CR11 punched card reader, CM11 marksense card reader or CD11 high speed punched card reader can be used to handle 80-column punched data cards or 40-column mark-sense cards.

Local terminals can be connected to the PDP-11 computer via the KL11, DL11A, DL11C, DL11WA, DL11B or LC11 line interface devices. Terminals on remote, dial-up lines can be connected via either the DC11 or DL11F remote line interface devices. Both local and remote lines can be connected through the DH11, DJ11, and DZ11 multiplexers, optionally with the DM11BB modem control multiplexer.

Industry Compatible Magnetic Tape using the TM11 controller and up to 8 TU10 drives or the TM02 controller and up to 8 TU16 drives may be configured. The TC11 Dectape controller and up to 8 TU56 dectape drives are also supported.

SYSTEM SOFTWARE

RSTS/E system software exists as PDP-11 assembly language code and as BASIC-PLUS language code. The assembly language code is tailored at system generation time according to the hardware configuration on which the system runs and to the software features which the system manager chooses to include in his RSTS/E system. Once the system is generated, this code is frozen and alterable only by use of patching or by generating new system code. The BASIC-PLUS language code exists in a system library of programs which are executed by the system itself of by individual users.

E1110 RSTS/E UETP HARDWARE USAGE TABLE

DEVICE	HARDWARE TESTED	COMMENTS
MEMORY	YES	48K minimum
RK05	YES	All units including 0 are exercised
RK06	YES	Same as RK05
RP02/RP03	YES	Same as RK05
RP04	YES	Same as RK05
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	Used only as UETP log output device (will be tested)
CARD READER	YES	Same as RK05
P.T. PUNCH	YES	Same as RK05
P.T. READER	YES	Same as RK05
DJ11	YES	Used with multiple keyboards and multi-level testing
DH11	YES	Same as DJ11
DZ11	YES	Same as DJ11

KL11,LC11,DL11A,DL11-B, DL11-C,DL11-D,DC11,DL11-E, DM11-BB	YES	Same as DJ11
DM11-A,DN11-A,DR11-A,C, PA611,DT03-FP,DX11, GT40,LPS11,KW11W,TA11, DQ11,DP11,DU11	NO	Not supported

E1120 DIFFERENCES BETWEEN RSTS/F UETP V06-05 AND V06-06
       (MAINDEC-11-DBZMA-B-D vs. MAINDEC-11-DRZMA-C-D)

This is a description of all the changes and enhancements placed into V06-05 to create V06-06. The following programs have been modified as indicated:

ACCTST

The ACCTST module determines if UETP is running on a V06A or V06B system and stores the result in the virtual array. This is necessary to allow programs to select the correct PEEK sequences. The ACCTST module asks seven new questions:

1. Paper tape reader (Y or N) - PREXER execution.
2. Paper tape punch (Y or N) - PPEXER execution.
3. Card reader (Y or N) - CREXER execution.
4. Keyboard exerciser (Y or N) - KBEXER execution.
5. Type number of RF11 platters? - DFEXER for V06B file structured RF11's.
6. Type number of RS03/RS04 drives? - DSEXER for V06B file structured RS03/RS04's.
7. Type number of RK06 drives? - RK06 exerciser.

This module has been partially cleaned up to meet the software coding standard for basic-plus.

The ACCTST module now determines if the running system has a minimal disk configuration, i.e., 2 RK05s or less. If this is true, it informs the user that more disk space is needed to run the UETP and some system library programs will have to be deleted. If the user elects to proceed ACCTST deletes the unneeded system library programs from account [1,2].

ACOTDL-ACOTST

The virtual array loading has been removed from here and placed in TDLFNS as a function common to all ACXTDL programs. The ACOTDL program now calls a TDLFNS routine to log out the pseudo keyboard before exiting to another module. This change had to be made due to a RSTS monitor change. The ACOTDL module does not print "SYSTEM RELIABILITY TESTS" unless it is actually going to run them (i.e., they are not run in a customer acceptance run). New script for the 6B error display program was added.

AC1TDL-AC1TST

The virtual array loading has been removed from here and placed in TDLFNS as a function common to all ACXTDL programs. The AC1TDL program now calls a TDLFNS routine to log out the pseudo keyboard before exiting to another module. This change had to be made due to a RSTS monitor change. The AC1TDL program now runs the appropriate number of DMEXER routines for RK06 drives. The appropriate number of DFEXER and DSEXER routines are run for the appropriate number of RF11 platters and RS03/04 drives.

AC2TDL-AC2TST

The virtual array loading has been removed from here and placed in TDLFNS as a function common to all ACXTDL programs. The AC2TDL program now calls a TDLFNS routine to log out the pseudo keyboard before exiting to another module. This change had to be made due to a RSTS monitor change. New script for the 6B error display program was added.

AC3TDL-AC3TST

The virtual array loading has been removed from here and placed in TDLFNS as a function common to all ACXTDL programs. The AC3TDL program now calls a TDLFNS routine to log out the pseudo keyboard before exiting to another module. This change had to be made due to a RSTS monitor change. During the user load test AC3TDL now runs 2 copies of FILMIN instead of one. New script for the 6B error display program was added.

AC4TDL-AC4TST

The virtual array loading has been removed from here and placed in TDLFNS as a function common to all ACXTDL programs. The AC4TDL program now calls a TDLFNS routine to log out the pseudo keyboard before exiting. This change has to be made due to a RSTS monitor change. New script for the 6B error display program was added.

AC5TDL-AC5TST

The virtual array loading has been removed from here and placed in TDLFNS as a function common to all ACXTDL programs. This is a new module to handle the UETP interactive tests. These include the paper tape reader test, papertape punch test, keyboard test, and the card reader test. The user can "chain" to AC0TST by answering "yes" to the CONTINUE WITH RELIABILITY TESTS question.

BUILDP

Program table increased to include AC5TST, DMEXER, DSEXEP, DFFXER, KREXER, ERRDPA, PPEXFR, and PREXER programs. The program now checks for a V06A or V06B systems and uses the correct PFFKs for each. The input media does not have to be magtape under account 8. The BUILDR program will now correctly lookup disk files and will accept accounts other than 8 for the input device.

CREXEP

This is a new program used in the interactive tests to test card readers.

CPUTST

This program was modified to avoid all use of matrix instructions.

CVTSCP

This program was changed to allocate files on SY: instead of SY0: in case on a small 2 RK05 system the first RK is full.

DFEXER

This is a new program added to exercise RSTS V06B file structured PF11 disks.

DMEXER

This is a new program added to exercise RK06 disk drives.

DSEXEP

This is a new program to exercise RSTS V06B file structured RS03/04 disk drives.

DSTATS

This program was modified to print device and unit number instead of

sequential disk number in system. Print-using was removed from this program.

ERRDPY, ERRDPA, ERRDP1 and ERRDP2

ERRDPY chains to ERRDPA which is the old ERRDPY. ERRDP2 has been modified to correctly obtain the DISK I/O STATISTICS.

FILMIN

This is a virtual file manipulation program. The file naming convention was changed to include the job number in the name. This allows running multiple copies of FILMIN.

MTEXER

This program now checks and determines if it should run at 1600 BPI and phase encoded PE bit or 800 BPI NRZI.

OPTPRM

This is a new program which mechanically exercises the RK06 disk drive. It reads from random sectors and random drives in non-file structured mode. The entire surface of the disk is not verified.

PKBSNF

This is an unofficial "new" program used only by Software Specialists in Manufacturing. Certain 11/70 system configurations make it impossible for one copy of the RSTS/E UETP to load the system down. More than one UETP requires more than one terminal device. PKBSNE allows a very knowledgeable user the ability to run a copy of the RSTS/E UETP from a pseudo keyboard in a detached state. It is very difficult to use and is a finger in the dike solution to this problem.

PPEXER

This is a new program used in the interactive tests to test paper tape punches.

PREXEP

This is a new program used in the interactive tests to test paper tape readers.

SCRPTR

This program is no longer a compressed version of SCRPTC. It was modified to handle 132 character wide terminals.

TDL and TDL1

These are the preprocessor programs for the Test Diagnostic Language (TDL) code. They are not distributed to the user. They were modified to reflect the new AC5TST, AC4TST, ACCTST and AC0TST configurations.

TDLFNS

A subroutine was added here to pick up the virtual array values stored by ACCTST. There was previously a copy of this routine in each ACXTDL. A check was added to determine the monitor type (V06B or V06A) and the corresponding PEEKs were changed. A subroutine was added to log out pseudo keyboards. This change was needed due to a monitor change which caused jobs to detach and hibernate.

The following programs and data files did not change from V06-05 to V06-06:

TAPSRT	TDLMOK	TDLOOK	IMMED.SCP (CTL FILE)
TAPSRU	TDLXOK	TDLSOK	FILES.SCP (CTL FILE)
TDLSRT	CPEXER	UDA	TTY.SCP (CTL FILE)
DXFXFR	CPFILE	VERIFY	VIRSTR.SCP (CTL FILE)
FLOPPY	NEWRM	DTEXER	CLUMSY.SCP (CTL FILE)
OPTPRP	DA	DKEXER	RANDAC.SCP (CTL FILE)
OPTPRK	JSTATS	DECMRG	CPU.SCP (CTL FILE)
SCRIPT	NEWRP	RANMAK	EDIT.SCP (CTL FILE)
			RELEAS.SEQ (FILE)
			RELEAS.SRT (FILE)
			MTSORT.ADT (FILE)

E2000 GETTING RSTS/E UETP ON THE AIR

This section describes the procedures to generate a RSTS/E V06B-02 system from the distribution media. The user performs the system generation procedure using the RSTS/E V06B-02 SYSGEN system generation monitor. Batch command files are used as the basis for the system generation process.

Following system generation the operator is led thru the RSTS/E initialization procedures which consist of refreshing the available disks and setting the system defaults. When this is completed the system library is loaded followed by the UETP library. The UETP control program ACCTST is then executed and the testing begins.

E2010 CONVENTIONS USED IN THIS DOCUMENT

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

```
TIME:  12:45
       -----
```

all responses are terminated by a carriage return (<CR>) unless otherwise indicated by having the line terminator enclosed in carots (i.e., <LF> for Line Feed).

E2100 BOOTSTRAP PROCEDURES

Mount the RSTS/E system generation media on unit 0 of the appropriate device. If the media is magtape or dectape insure that it is write locked. If the media is RK05 or RK06 insure that it is write protected.

Go to Section E2110, E2120, E2130, E2140, F2145, E2150, F2155 respectively if the bootstrap device is the RM873-YA, RM873-YB, MR11-DB, RM792-YB, M9301-YA or YB, M9301-YC, M9301-YF. If none of the above bootstrap devices are present, proceed to the section indicated for the media to be bootstrapped:

DEVICE	SECTION
TU10/TS03	E2160
TU16/TU45	E2163
RK05	E2165
RK06	E2167

For additional information on BOOT proecdures, refer to the System

Generation Manual, Chapter 2 and Appendix A.

E2110 RM#73-YA PROCEDURE

If the RM#73-YA Restart/Loader is on the system, perform the following steps.

Move the CPU Console ENABLE/HALT switch to its HALT position and back to its ENABLE position.

Set the CPU Switch Register to one of the following values depending on the system option from which bootstrapping is to be accomplished:

773000 for RF11 disk
773010 for RK11 disk cartridge
773100 for RP03 disk pack
773050 for TM11/TU10 magtape
773030 for TC11/TU56 DECtape

Depress the CPU LOAD ADRS switch.
Depress the CPU START switch.
Go to section F2180.

E2120 RM#73-YB PROCEDURE

If the RM#73-YB Restart/Loader is on the system, perform the following steps.

Move the CPU Console ENABLE/HALT switch to its HALT position and back to its ENABLE position.

Set the CPU Switch Register to one of the following values depending on the system option from which bootstrapping is to be accomplished:

773030 for RK11 disk cartridge
773136 for RF11 disk
773320 for RP04 disk pack
773350 for RP03 disk pack
773110 for TM11/TU10 magtape
773150 for TM02/TU16 magtape
773070 for TC11/TU56 DECtape

Depress the CPU LOAD ADRS switch.
Depress the CPU START switch.
Go to Section F2180.

E2130 MR11-DR PROCEDURE

If the MR11-DR Bulk Storage Loader is on the system, perform the following steps.

Move the CPU Console ENABLE/HALT switch to its HALT position and back to its ENABLE position.

Set the CPU Switch Register to one of the following values depending on the system option from which bootstrapping is to be accomplished:

773100 for RF11 disk
773110 for RK11 disk cartridge
773154 for RP03 disk pack
773136 for TM11/TU10 magtape
773120 for TC11/TU56 DECtape

Depress the CPU LOAD ADRS switch.
Depress the CPU START switch.
Go to Section E2180.

E2140 RM792-YB PROCEDURE

If the RM792-YB Hardware Loader is on the system, perform the following steps.

Move the CPU Console ENABLE/HALT switch to its HALT position and back to its ENABLE position.

Set the CPU Switch Register to 773100.

Depress the CPU LOAD ADRS switch.

Set the CPU Switch Register to one of the following values depending on the system option from which bootstrapping is to be accomplished:

777462 for RF11 disk
777406 for RK11 disk cartridge
776716 for RP03 disk pack
777344 for TC11/TU56 DECtape

Depress the CPU START switch.
Go to Section E2180.

E2145 M9301-YA AND M9301-YB PROCEDURE

If the M9301-YA or M9301-YB bootstrap is on the system, perform the following steps.

While holding down the CTRL switch press the BOOT switch on the front of the computer or load address 773000 and depress CPU START. The bootstrap prints the contents of registers R0, R4, the Stack Pointer and the program counter. It then prints the @ character.

Type the device code and unit number followed by the RETURN key. The following are the device codes:

MT	TU10/TS03 Magtape
MM	TU16/TU45 Magtape
DK	RK05 Disk Cartridge
DP	RP02/03 Disk Pack
DB	RP04/05/06 Disk Pack
DS	RS03/04 Fixed Head Disk
DT	TU56 Dectape
DX	RX11 Floppy Disk

Go to section E2180

E2150 M9301-YC PROCEDURE

If the M9301-YC Bootstrap/Diagnostic loader is on the system perform the following steps.

Move the CPU console ENABLE/HALT switch to its HALT position and back to its ENABLE position.

Set the CPU switch register to 17765000.

Depress the CPU LOAD ADRS switch.

Set the CPU switch register to one of the following values depending on the system option from which bootstrapping is to be accomplished (unit 0 only):

00000010 for TM11/TU10 magtape
00000020 for TC11/TU56 DECtape
00000030 for RK11 disk cartridge
00000040 for RP03 disk pack
00000070 for RP04 disk pack
00000060 for TM02/TU16 magtape
00000100 for RS03/04 fixed head disk

Depress the CPU START switch
Go to Section E2180.

E2155 M9301-YF PROCEDURE

If the M9301-YF bootstrap is on the system, perform the following steps.

While holding down the CTRL switch, press BOOT switch on the front of the computer.

The bootstrap points the contents of registers R0, R4, the Stack Pointer and the program counter. It then prints the @ character.

Type the device code and unit number followed by the RETURN key. The following are the device codes:

MT	TU10/TS03 Magtape
MM	TU16/TU45 Magtape
DK	RK05 Disk cartridge
DM	RK06 Disk cartridge
DP	RP02/03 Disk pack
DB	RP04/05/06 Disk pack
DS	RS03/04 Fixed head disk
DT	TU56 Dectape

Go to section F2180.

E2160 BOOTSTRAPPING TM11/TU10 MAGTAPE WITHOUT MR11-DB OR BM873 LOADERS

To bootstrap a TM11/TU10 magtape when the system has no hardware bootstrap loader, the user must manually enter a load routine into memory using the CPU console Switch Register and the DEP switch.

To load the routine, perform the following steps.

Move the CPU Console ENABLE/HALT switch to its HALT position and back to its ENABLE position.

Set the CPU Switch Register to 010000.

Depress the CPU LOAD ADRS switch.

Load the following contents into memory using the Switch Register and DEP switch.

Address	Contents
010000	012700
010002	172524
010004	005310
010006	012740
010010	060011
010012	105710
010014	100376
010016	005710
010020	100767
010022	012710
010024	060003
010026	105710
010030	100376
010032	005710
010034	100777
010036	005007

Set the Console Switch Register to 010000.
Depress the CPU LOAD ADRS switch.
Depress the CPU START switch.
Go to Section E2180.

If the system reads the tape but halts at address 010034, the device generated a magtape error. The user can try another drive. If the system appears to take no action and halts, verify the accuracy of the routine by using the CPU Console EXAM switch. Use the Switch Register and the DEP switch to correct any erroneous contents. Rewind the tape to its load point before executing the routine again. If no recovery is successful, it will be necessary to have the hardware checked.

E2163 BOOTSTRAPPING TM02/TU16/TU45 MAGTAPE WITHOUT A HARDWARE BOOTSTRAP LOADER

To bootstrap a TU16/TU45 magtape when the system has no hardware bootstrap loader, the user must manually enter a load routine into memory using the CPU console Switch Register and the DEP switch.

To load the routine, perform the following steps.

Move the CPU Console ENABLE/HALT switch to its HALT position and back to its ENABLE position.

Set the CPU Switch Register to 010000.

Depress the CPU LOAD ADRS switch.

Load the following contents into memory using the Switch Register and DEP switch.

Address	Contents
010000	012700
010002	172440
010004	012760
010006	001300
010010	000032
010012	012760
010014	177777
010016	000006
010020	012710
010022	000031
010024	005760
010026	000012
010030	100375
010032	012760
010034	177400
010036	000002
010040	012710
010042	000071
010044	105710
010046	100376
010050	005710
010052	100777
010054	005007

Set the Console Switch Register to 010000.
Depress the CPU LOAD ADRS switch.
Depress the CPU START switch.
Go to Section E2180.

If the system reads the tape but halts at address 010052, the device generated a magtape error. The user can try another drive. If the

system appears to take no action and halts, verify the accuracy of the routine by using the CPU Console EXAM switch. Use the Switch Register and the DEP switch to correct any erroneous contents. Rewind the tape to its load point before executing the routine again. If no recovery is successful, it will be necessary to have the hardware checked.

E2165 BOOTSTRAPPING RK11/RK05 DECPACK WITHOUT A HARDWARE BOOTSTRAP LOADER

To bootstrap an RK11/RK05 Decpack cartridge when the system has no hardware bootstrap loader, the user must manually enter a load routine into memory using the CPU Console Switch Register and the DEP switch.

To load the routine perform the following steps.

Move the CPU Console ENABLE/HALT switch to its HALT position and back to its ENABLE position.

Set the CPU Switch Register to 010000.

Depress the CPU LOAD ADRS switch.

Load the following contents into memory using the Switch Register and the DEP Switch.

Address	Contents
010000	012737
010002	000005
010004	177404
010006	000001

Set Console Switch Register to 10000,
Depress the CPU LOAD ADDRESS,
Depress the CPU Start Switch,
Wait 5 seconds and then depress HALT
Set the Console Switch Register to 000000,
Depress the CPU LOAD ADRS switch,
Depress the CPU START switch,
Go to section F2180.

If the system appears to take no action and halts, verify the accuracy of the routine by using the CPU Console EXAM switch. Use the Switch Register and the DEP switch to correct any erroneous contents. If no recovery is successful, it will be necessary to have the hardware checked.

E2167 BOOTSTRAPPING RK611/RK06 DECPACK WITHOUT
A HARDWARE BOOTSTRAP LOADER

To bootstrap an RK611/RK06 DEC pack cartridge when the system has no hardware boostrap loader, the user must manually enter a load routine into memory using the CPU Console Switch Register and the DEP switch.

To load the routine perform the following steps.

Move the CPU Console HALT/ENABLE switch to its HALT position and back to ENABLE.

Set the CPU Switch Register to 010000.

Depress the CPU Load Address Switch.

Load the following contents into memory using the Switch Register and the DEP Switch.

Address	Contents
010000	012737
010002	177400
010004	177442
010006	012737
010010	000000
010012	177444
010014	012737
010016	000000
010020	177446
010022	012737
010024	000000
010026	177460
010030	012737
010032	000021
010034	177440
010036	013701
010040	177440
010042	105701
010044	100374
010046	000000
010050	177440
010052	000003

Set the Console Switch Register to 010000.
Depress the CPU LOAD ADRS switch.
Depress the CPU START switch.
The computer will halt.
Set the Console Switch Register to 000000.
Depress the CPU LOAD ADRS switch.
Depress the CPU START switch.
Go to section E2180.

If the system appears to take no action and halts, verify the accuracy of the routine by using the CPU console EXAM switch. Use the Switch Register and the DEP switch to correct any erroneous contents. If no recovery is successful, it will be necessary to have the hardware checked.

E2170 SUMMARY OF HARDWARE BOOTSTRAP ADDRESSES

Device to BOOTSTRAP		Bootstrap Type		
	BM873-YA	BM873-YB	MR11-DB	BM792-YB(1)
RF11 DISK	773000	773136	773100	777462
RS03/04	-	-	-	-
RK11 DISK CARTRIDGE	773010	773030	773110	777406
RP03 DISK PACK	773100	773350	773154	776716
RP04, RP05 OR RP06 DISK PACK	-	773320	-	- -
TM11/TU10 and TS03 magtape	773050	773110	773136	(2)
TM02/TU16 and TU45 magtape	-	773150	-	-
TC11/TU56 DECtape	773030	773070	773120	777344
RK611/RK06 DISK CARTRIDGE	-	-	-	-
RX01/RX01 floppy disk	-	-	-	-

(1) For the BM792-YB loader, set the address 773100 in the Switch Register, depress the LOAD ADRS switch, set the value from the table in the Switch Register, and press the START switch. (2) To bootstrap a magtape, use the loading routine described in Section E2160. (3) For the M9301-YC Loader, set the address 17765000 in the switch register, depress the LOAD ADRS switch, set the value from the table in the switch register, and press the START switch.

	M9301-YA	M9301-YB	M9301-YC(3)	M9301-YF
RF11 DISK	-	-	-	-
RS03/04	DS	DS	00000100	DS
RK11 DISK CARDTRIDGE	DK	DK	00000030	DK
RP03 DISK PACK	DP	DP	00000040	DP
RP04, RP05 OR RP06 DISK PACK	-	DB	00000070	DB
TM11/TU10 AND TS03 MAGTAPE	MT	MT	00000010	MT
TM02/TU16 AND TU45 MAGTAPE	-	MM	00000060	MM
TC11/TU56 DECTAPE	DT	DT	00000020	DT
RK611/RK06 DISK CARTRIDGE	-	-	-	DM
RX01/RX01 FLOPPY DISK	DX	DX	-	DX

E2180 LOADING THE SYSTEM GENERATION MONITOR

When the distribution media has been boot-strapped the console terminal will print:

```
     RSTS V06B (dev)          (Dev is the name of the device that
                               was just booted).

     OPTION:
```

Proceed to section E2181.

E2181 TARGET SYSTEM DISK INITIALIZATION (DSKINT)

The SYSGEN monitor must be placed on disk to run a system generation. This disk will be the same disk as the final RSTS monitor and BASIC-PLUS run time system will reside. The following creates the minimal RSTS/E file structure on the target disk and pattern checks the disk to allocate bad blocks as un-useable.

For further information on the DSKINT Option refer to section 3.2 of the SYSTEM GENERATION MANUAL.

Mount a scratch disk pack (write enabled) in an available disk drive. Initialize the disk using the following guide...

Summary of DSKINT Option

Question	Possible Response
OPTION:	DSKINT
DD-MMM-YY?	DD-MMM-YY
HH:MM?	HH:MM
DISK?	DP, DB, DM, DK (specify the mnemonic name of the disk that you selected to be the target disk where the RSTS that you generate will reside).
UNIT?	0 (asked for multi-unit controllers only)

```
PACK ID?                    SYSPAK
                            ------

PACK CLUSTER SIZE?          4 (2 for RK, RF or RS disk 8 for RP06 disk)
                            --

SATT,SYS BASE?              <LF>
                            ----

MFD PASSWORD?               SYSMFD
                            ------

MFD CLUSTER SIZE?           4 (2 for RK, RF or RS disk 8 for RP06 disk)
                            --

PUB, PRI, or SYS?           SYS
                            ---

LIBRARY PASSWORD?           SYSLIB
                            ------

LIBRARY CLUSTER SIZE?       4 (2 for RK, RF or RS disk 8 for RP06 disk)
                            --

DATE LAST MODIFIED?         Y
                            --

FORMAT?                     N
                            --

PATTERNS?                   1 (use 8 patterns in F,A,  & T,)
                            --

PROCEED (Y or N)?           Y
                            --

PATTERN #1
```

NOTE

The system prints out all the bad blocks uncovered during a pattern test. The following is a sample printout for an RP04 pack with bad sectors

```
SECTOR   CLUSTER   RPDS   RPER   RPCS   RPWC   RPMA   RPCA   RPDA

 68973    34486   160001 000150 100206 000000 002000 000530 010445
```

If bad blocks are detected on an RK, RS or RF disk, this disk should be replaced and DSKINT run again on the new disk.

Proceed to Section E2182.

E2182 TRANSFERRING THE SYSGEN MONITOR FROM DISTRIBUTION MEDIA TO DISK

This procedure copies the SYSGEN monitor and RT11 runtime system from the distribution media to the target system disk.

Question	Possible Response
OPTION:	COPY
DD-MMM-YY	<LF> (If not correct enter correct date)
HH:MM	<LF> (if not correct enter correct time)
TO WHICH DISK?	DK, DM, DP or DB (Specify the disk that you initialized as the target system disk in section E2181)
UNIT?	0-2 (specify the unit number on which the target disk is physically mounted. Note that this question is asked for multi-unit controllers only).

NOTE

As a result of the copy option the target disk will be automatically bootstrapped. At this point the INIT.SYS program is running from the target disk.

RSTS V06B (new dev) (New dev indicates the disk just bootstrapped, i.e., the target disk)

OPTION:

Proceed to section E2185.

E2185 CHECKING THE HARDWARE CONFIGURATION

The configuration can be checked in INIT by the HARDWARE option.

The hardware configuration should be verified at this time. In response to the OPTION query type:

```
OPTION:HA LI
       -- --
```

The INIT program will now list the hardware devices, addresses and vectors. The list should be checked against the equipment on the system before proceeding. After checking equipment list proceed to section F2186. If there are any discrepencies, contact a software specialist.

F2184 ESTABLISH FILES FOR THE SYSGEN SYSTEM

Use the REFRESH option to establish the necessary files. The table below contains questions and appropriate responses for this use of REFRESH. Refer to section 3.7 of the System Generation Manual for further information on REFRESH. In response to the OPTION: query type REFRESH.

OPTION:	REFRESH
DD-MMM-YY?	<LF> (If not correct enter correct date)
HH:MM?	<LF> (If not correct enter correct time)
DISK?	DC,DF,DS,DK,DM,DP, OR DB (Current system disk)
UNIT?	Current System disk unit #
CLEAN?	Y
DISK IS BEING CLEANED - WAIT...	
REFRESH SUBOPTION?	CHANGE
SWAP,SYS CHANGES?	Y
SIZE?	224

```
BASE?                       <LF>
                            ----

SWAP0.SYS CHANGES?          <LF>
                            ----

SWAP1.SYS CHANGES?          <LF>
                            ----

SWAP3.SYS CHANGES?          <LF>
                            ----

OVP.SYS CHANGES?            <LF>
                            ----

ERR.SYS CHANGES?            <LF>
                            ----

BUFF.SYS CHANGES?           <LF> (If DECtape on system answer YES)
                            ----

SIZE?                       40
                            --
                                 (NOT ASKED IF NO DECtape on system)
BASE?                       <LF>
                            ----

     CRASH.SYS CHANGES?     Y
                            -

     SIZE?                  40
                            --

     BASE?                  <LF>
                            ----

     OTHER FILES?           <LF>
                            ----

     REFRESH SUBOPTION?     LIST
                            ----

     Operator should note that all files are listed as OK (If no
     DECtape on system, BUFF.SYS will not have an OK status - this is
     alright).

     REFRSH SUBOPTION?      <LF>
                            ----

System will now print OPTION:  query.  Proceed to E2187.


E2187  INSTALLING THE SYSGEN MONITOR
       ----------------------------
```

In response to the OPTION query type:

```
OPTION:INSTALL
       -------
```

In response to the SIL query answer SYSGEN for the sysgen monitor.

```
  SIL ? SYSGEN
        ------
```

Proceed to section F2188 when system prints the OPTION: query.

E2188 ESTABLISHING SYSGEN MONITOR DEFAULTS

The monitor defaults must be set. In response to the OPTION query type:

```
     OPTION:              DEFAULT
                          -------

     NO DEFAULTS ARE CURRENTLY SET
     YOU CURRENTLY HAVE JOBMAX = 2, SWAPMAX = 8K

     JOB MAX OR SWAP MAX
     CHANGES?             Y
                          -

     NEW JOB MAX?         <LF>
                          ----

     NEW SWAP MAX?        28
                          --

     NOTE:  SWAPMAX should be 26 if only 48K of memory on system.

     YOU CURRENTLY HAVE JOBMAX = 2, SWAPMAX = 28K
     JOBMAX OR SWAPMAX CHANGES? <LF>
                                ----

     RUN-TIME SYSTEM?     RT11
                          ----

     ERROR MESSAGE FILE? ERR
                          ---

     INSTALLATION NAME?  SYSGEN SYSTEM (In FACT, use DFC 8)
                         -------------
```

The INIT program now prints a memory allocation table at this point. It should be checked to make sure all memory is present and of the correct type.

```
     TABLE SUBOPTION?     <LF>
                          ----

     YOUR CURRENTLY HAVE CRASH DUMP DISABLED.

     CRASH DUMP?          Y
                          -

     MAGTAPE LABEL DEFAULT (NONE)?  DOS
                                    ---
```

The system will now print the OPTION: query again. Proceed to section E2189.

E2180 ESTABLISHING DEVICE UNIT CHARACTERISTICS FOR THE SYSGEN MONITOR

Device characteristics are set using the SET option of the INIT program. In response to the OPTION query type SET.

```
OPTION:             SET
                    ---
SET SUBOPTION?      LIST
                    ----
DEVICE?             <LF>
                    ----
```

Init will now print a device list for all units that are supported by the currently installed SIL.

```
SET SUBOPTION?      LP
                    --
TYPE?               LP or LV or LS
                    --------------
UNIT?               Only asked if more than 1
                    -------------------------
WIDTH?              80, 120, or 132
                    ---------------
LOWER CASE?         Y or N
                    ------
SET SUBOPTION?      <LF>
                    ----
OPTION:
```

E2190 STARTING THE SYSGEN SYSTEM

Use the START option to start the SYSGEN system. Invoke the option by typing the LINE FEED key in response to the OPTION query. Refer to section 3.10 of the System Generation Manual for information on start-up messages and errors.

If all has gone correctly, the system will type:

```
OPTION:  <LF>
         ----

YOU CURRENTLY HAVE:  JOB MAX=2, SWAP MAX=20K,
YOU CURRENTLY HAVE CRASH DUMP ENABLED.

DATE  DD-MMM-YY <LF> if OK
              ----
TIME  HH:MM? <LF> if OK
             ----

N DEVICES DISABLED
```

NOTE: ANY DEVICES THAT ARE CONFIGURED INTO THE SYSGEN monitor but are not physically on the system will be disabled.

? Can't find file or account

.

E2200 STARTING THE SYSTEM GENERATION

Once the system generation monitor (SYSGEN.SIL) is running and the "Can't find file or account" message is printed, the console terminal is logged in under account [1,2] and the default run-time system is RT11 as indicated by the "." for a prompt.

If the distribution media is magtape then physically mount the tape labelled (DEC-11-ORSPA-E-MC9 for 9 track or -MC7 for 7 track) (Sysgen tape) on an available drive, WRITE PROTECTED.

If the distribution media is RK05 or RK06 then:

1. Physically mount the system generation disk (DEC-11-ORSPA-E-HC for RK05 or -BC for RK06) in an available drive, WRITE PROTECTED. (unless you have already mounted it previously).

2. Logically mount the disk by typing the following command...

```
.MOUNT Dev:ORSPAE/RO
```

Where 'Dev' is the disk and unit number of the drive that you physically mounted during step 1.

To run the batch file that initiates the system generation dialogue type the following command, where 'Dev' is replaced by the name and unit number of the drive in which the distribution media is mounted.

```
.R Dev:CREATE.SAV
```

Create enables logins, transfers some files to the system disk, and then starts the sysgen dialogue. The following is an example of the first portion that CREATE outputs to the console terminal. (For this example the sysgen tape is on MM0:).

```
^C
HELLO 1/2
Password:
1 other user is logged in under this account
```

```
.ASSIGN MM0:
.ASSIGN MM0:IN

.R IN:PIP.SAV
*SY:=*.*,*=IN:$PIP.SAV,$LOGOUT.SAV,$UTILTY.SAV
*SY:=*.*,*=IN:$MACRO.SAV,$CREF.SAV,$LINK.SAV,$SILUS.SAV,$HOOK.SAV/N
*SY:=*.*,*=IN:$SYSGEN.SAV,$SYSBAT.SAV,$FRP.STB/N
*C

.DISMOUNT MM0:
.DEASSIGN

.R LOGOUT
Confirm: Y
Saved all disk files; 272 blocks in use
Job 2 User 1,2 logged off KB1 at DD-MMM-YY HH:MM
1 other user still logged in under this account
System PSTS V06B-02 SYSGEN SYSTEM
Run time was 0.3 seconds
Elapsed time was 1 minute
Good afternoon



**DD-MMM-YY
```

Beginning of PSTS/E system generation.

Proceed to section F2300.

E2300 CONFIGURATION QUESTIONS

After the batch command file starts, the system generation program SYSGEN runs and enters a dialogue with the user. The dialogue is a series of approximately 60 hardware and software configuration questions.

In case a question is unclear, the user need type only the RETURN key in response to a short form query and SYSGEN prints the long form of that particular question.

If the user wants to restart system generation, he can do so by using the checkpoint facility described in Section E2320.

Implications of the configuration questions are given in Section E2330, and the operator is supposed to follow the check sheet he has been provided with. If a particular response is not indicated a default response of line feed is desired.

After the user answers the configuration questions, a second batch command file must be started to build the new monitor. For information on this part of the procedure, consult Section F2350.

If the user is uncertain of any reply to the configuration questions, Section E2310 should be read carefully.

E2310 AUTOMATIC ANSWERS

The automatic answer capability allows SYSGEN to physically check the hardware configuration of the computer on which it is running. SYSGEN actually addresses each device to determine its existence and the number and types of units.

SYSGEN denotes the meaning of the automatic answer by printing it enclosed in special characters as shown below.

1	The answer 1 is correct for the current hardware.
0	The answer assumes something concerning the system; the user should verify the answer.
??	The answer cannot be determined; an answer must be entered.

For example, the program can accurately determine whether the computer has either FIS or FPP and thus prints either **Y** or **NO** as the correct answer. However, for certain devices such as DECtape, Magtape, and RK disk, SYSGEN can only verify the presence of the controller and must assume the existence of 1 drive. The answer for such devices is either **0** or **0** as the assumed answer for the number of drives based upon the absence or presence of the controller. The user should verify SYSGEN'S assumption and enter the correct answer. Example:

TU10'S: **0** 2<CR>

The following format indicates valid responses to automatic answers.

LINE FEED key	Use the automatic answer as the response.
x and RETURN key	Use the value, typed as x, as the response.
RETURN key	Reprint query or print long form of the query.

If SYSGEN prints an answer and it appears to be incorrect, the hardware is possibly connected incorrectly. For example, terminal interface jumpers can be cut improperly.

If an error is suspected by an incorrect automatic answer, go back and check the hardware list option for correct addresses and vectors.

Care must be taken if an automatic answer is overridden for a floating address device. Devices such as the DJ11, DH11, DQ11, and DU11 have so called floating addresses which vary depending upon the presence or absence of other floating addresses devices. Before printing the automatic answer for such a device, SYSGEN recomputes the correct floating addresses based on responses to previous questions. For example, DJ11 devices are assigned addresses on the UNIBUS before DH11 devices. If 1 is the automatic answer for the DJ11 question but is overridden by typing 2, SYSGEN recomputes the floating address of and looks for the presence of a DH11 at an address based on 2 DJ11 multiplexers. This procedure is correct only if the jumpers for any DH11 are cut correctly for a system with 2 DJ11 multiplexers. Similarly, if the same automatic answer is overridden by entering 0, SYSGEN recomputes the floating address of and looks for the presence of DH11 devices at an address based on no DJ11 devices. Automatic answers SYSGEN computes for DH11, DQ11, and DU11 devices are incorrect if the user overrides the automatic answer of 1 DJ11 and the hardware is correctly configured for 1 DJ11 device. For more information on floating address assignments, see Sections E5100 and E5200.

E2320 CHECKPOINTS

There are two places where a sysgen can be restarted by resuming that part of the process which was interrupted.

The first point immediately precedes the configuration dialogue. The second point immediately precedes the execution of the second batch command file SYSGEN.CTL. To restart the dialogue type CONTROL/C and then R SYSGEN. To restart the batch file SYSGEN.CTL type CONTROL/C and R SYSBAT.

For example, to abort SYSGEN and terminate the batch stream, type the CONTROL/C combination. Assume SYSGEN prints a configuration question and the user types the CONTROL/C combination.

```
AC FREQ? ^C
         --
.
```

The monitor echoes the CONTROL/C combination and prints the dot. The user then types R SYSGEN to restart the dialog.

For example, to abort the batch processor portion of the System Generation procedure, type the CONTROL/C combination; assume SYSBAT prints a line and the user types the CONTROL/C combination.

```
^C
--
.
```

the monitor echoes the CONTROL/C combination and prints the dot. The user then types R SYSBAT to restart the batch processor.

E2325 SYSGEN ANSWERS FOR UETP

The following questions must be answered as shown in order for the UETP control programs to function properly.

FORM? S/Q ---		- You must answer first with S or L followed by a /Q to indicate that job and disk statistics are to be retained by the RSTS/E monitor.
PSEUDO KEYBOARDS ? 10 --		- At least ten pseudo keyboards are required. If multi-level processing, more than one copy of ACCTST, is desired, then # of copies times 10 e.g., 3 copies of ACCTST would require 30 for a type-in. See Section E2800 for multi-level requirements.
MAXIMUM JOBS ? 15 --		- At least 15 jobs are required. If multi-level processing, more than one copy of ACCTST, is desired, then type number of copies of ACCTST*(10)+5. See section E2800 for multi-level requirements.
SMALL BUFFERS ? 150 ---		- A minimum of 10 times the maximum number of jobs. The type-in should not exceed 400.
FUNCTIONS ? Y -		- Extended mathematical functions are necessary.
MATH PRECISION ? 2 -		- 2 word math will cause the CPU to work harder.
PRINT USING ? Y -		- PRINT USING command is necessary (only for expanded job mix, statistics).

All other questions should be answered according to the particular hardware configuration being used Following the system software exerciser checklist that has been provided.

E2330 POSSIBLE SYSGEN QUESTIONS AND ANSWERS

If the operator is familiar with the SYSGEN questions he should proceed below. If not he should first consult section E2325 to determine the answers fixed for the UETP test. Otherwise he should follow the system software exerciser checklist that has been provided.

The operator will be asked certain questions according to the system configuration. This section deals with all possible questions and answers in short form, for long form consult Section E7000. All the questions will be asked with automatic answers, see Section E2310 if unfamiliar.

During this dialogue, if the operator wants to restart, there are two checkpoints available, consult Section E2320.

Question	Possible Responses	Comments
FORM?	S,L/Q	Answer S for short form or L for long form. The /Q will generate the statistics code in the monitor for use by the UETP.
SAME SYSTEM?	YES, NO	(FA+T always YES)
DISTRIBUTION MEDIUM?	DT,MT,MM,DK,DM,SY	Answer SY if RSTS/E is already on the system disk.
OUTPUT MEDIUM?	SY,DF,DK,DM,DP,DS,MT,MM	Target system disk. Answer SY since SYSGEN is running on target disk.
PACK ID?	Pack ID for disk	Asked only if output medium is DF, DK, DP, DS, DR, or DM.
DELETE FILES?	YES, NO	ESC returns to Output medium.
LP FOR SYSGEN?	YES, NO	Used for printing load maps.
GENERATE MONITOR?	YES,NO	If you answer NO, the next question

		BASIC-PLUS RTS NAME.
MONITOR NAME?	1 to 6 alphanumeric	The RSTS/E monitor has the extension SIL.
ASSEMBLY LISTINGS?	YES, NO	Asked only if answer to LP for SYSTEM? was YES.
GENERATE BASIC-PLUS?	YES, NO	If you answer NO, the next question is clock? ESC returns to GENERATE MONITOR.
BASIC-PLUS RTS NAME?	1 to 6 alphanumeric characters	BASIC-PLUS run-time system has the extension RTS.

Now you must specify the hardware configuration on which this RSTS/E system will run.

CLOCK?	L, P, C	If SYSGEN finds both the KW11-L and KW11-P clocks, its automatic answer is C.
AC FREQ?	50,60	
KW11-P INTERRUPT RATE?	50,100,150...10000	Asked only if the answer to clock? is C.
KL11,LC11,DL11A,DL11B's?	1 to 16	Include the console terminal in your count.
DL11C, DL11D's?	0 to 31	Do not include the console terminal in your count.
DC11's?	0 to 32	
DC11E's	0 to 31	After this question, SYSGEN returns to KL11, LC11, DL11A, DL11B's? If you have configured more than 31 DL11C, DL11D, & DL11E

		interfaces.
DJ11's	0 to 16	If you answer 0, the next question is DH11's?
DJ11 UNIT ** LINES ENABLED?	0 TO 16	SYSGEN repeats this question for each DJ11 unit configured. ESC returns to DJ11's?
DH11's?	0 to 16	If you answer 0, the next question is DZ11's?
DH11 UNIT ** LINES ENABLED?	0 TO 16	SYSGEN repeats this question for each DH11 unit configured.
DATA SET SUPPORT FOR DH11's?	YES, NO	ESC returns to DH11's?
DZ11's	0 to 16	If you answer 0, the next question PSEUDO KEYBOARDS? ESC returns to DH11 UNIT *** LINE ENABLED?
DZ11 UNIT ** LINES ENABLED?	0 TO 16	SYSGEN repeats this question for each DZ11 unit configured.
DATASET SUPPORT FOR DZ11's?	YES, NO	ESC returns to DZ11's?
PSEUDO KEYBOARDS?	0 TO 63	(10 * number of copies of ACCTST) If you have configured more than 128 terminals (including the console terminal and pseudo keyboards), SYSGEN returns to the question KL11, LC11, DL11A, DL11B's?
2741 SUPPORT?	YES, NO	Asked only if you

		configured one or more DL11D, DL11E, DC11, DH11, or DZ11 interfaces. If you answer NO, the next question is MULTI-TTY SERVICE?
SINGLE LINE 2741 SUPPORT?	YES, NO	Asked only if you configured DL11D, DL11E, or DC11 interfaces.
2741 SUPPORT ON DH's?	YES, NO	Asked only if you configured one or more DH11's. ESC returns to 2741 support?
2741 SUPPORT ON DZ's?	YES, NO	Asked only if you configured one or more DZ11's. ESC returns to 2741 support?
2741 CODE(S)?	CORR,EBCD,SBCD,C360	ESC returns to 2741 support?
MULTI-TTY SERVICE?	YES, NO	ESC returns to 2741 support if the system has 2741-compatible interfaces, or to PSEUDO KEYBOARDS? If it does not,
ECHO CONTROL?	YES, NO	(yes allows terminals to be opened in block mode).
RC/RS64's?	YES, NO	
RF/RS11's?	YES, NO	
RS03/RS04's?	0 to 8	
RP02/RP03's?	0 to 8	If the controller exists, the automatic answer is 1. If the answer is 0 or 1, SYSGEN skips

		the next question.
OVERLAPPED SEEK?	YES, NO	
RP04/RP05/RP06's?	0 to 8	If you answer 0 or 1 SYSGEN skips the next question.
OVERLAPPED SEEK?	YES, NO	
RK05's?	0 to 8	If the controller exists, the automatic answer is 1. If the answer is 0 or 1, SYSGEN skips the next question
OVERLAPPED SEEK?	YES, NO	If the controller has any RK05-F units attached then answer NO to this question.
RK06's?	0 to 8	If the controller exists, the automatic answer is 1. If the answer is 0 or 1, SYSGEN skips the next question.
OVERLAPPED SEEK?	YES, NO	
TU16's	0 to 8	
TU10's?	0 to 8	
DECTAPES?	0 to 8	
PRINTERS?	0 to 8	
RX01's?	0 to 8	
CR11/CM11 CARD READER?	YES, NO	
CD11 CARD READER?	YES, NO	
CARD DECODE?	029,026,1401,ANSI	Skipped if you answered NO to the two previous prompts. ESC returns to CR11/CM11 CARD READER

P.T. READER?	YES, NO	
P.T. PUNCH?	YES, NO	
DECNET NETWORK SUPPORT?	YES, NO	If you answer NO, SYSGEN skips the next two questions.
DMC11's?	1 to 8	
DECNET/E DISTRIBUTION MEDIUM?	SY,DT,MT,MM,DK,DM	ESC returns to DECNET NETWORK SUPPORT?
2780 SUPPORT	YES,NO	If you answer NO, SYSGEN skips the next two questions.
2780 INTERFACE?	DP, DU	If both are on the system, the automatic answer is DU.
2780 DISTRIBUTION MEDIUM?	SY,DT,MT,MM,DK,DM	ESC returns to 2780 support?
MAXIMUM JOBS?	1 to 63	(number of copies of ACCTST*(10)+5)
SMALL BUFFERS?	30 to 999	The automatic answer is 9 times the configured job maximum plus 80. Do not exceed 400.
SYSTEM WIDE LOGICALS?	0 to 30	At least one is needed if RK05F's are present
DELAY FACTOR?	1 to 300	
FIP BUFFERING?	YES, NO	
RESIDENT DISK HANDLING?	YES, NO	
RESIDENT SEND/RECEIVE?	YES, NO	
RESIDENT SYS CALLS?	YES, NO	
RES. LOGIN/ATTACH/ATTRIBUTE?	YES, NO	
RESIDENT CATALOG LOOKUP?	YES, NO	

The following questions deal with the BASIC-PLUS run time system.

FPP?	YES, NO	If you answer YES, SYSGEN skips the next question.
FIS?	YES, NO	
MATH PRECISION?	2, 4	
FUNCTIONS?	YES, NO	
TIME FORMAT?	AM, 24-HOUR	
ALPHABETIC MONTH?	YES, NO	
PRINT USING?	YES, NO	
MATRICES?	YES, NO	
STRING ARITHMETIC?	YES, NO	

The system generation dialog is finished. If you have any special requirements which require editing the Generated File CONFIG.MAC, System Configuration File, or SYSGEN.CTL, Batch Control File, you may do it now. When ready type "R SYSBAT".

E2350 RUNNING THE BATCH PROCESSOR

To start the batch process to generate the monitor and basic plus type R SYSBAT. At this point BATCH will continue to print messages on the console. All instructions should be followed.

An example of the BATCH message follows:

```
.R SYSBAT
 --------
SYSGEN batch processing has started.
If any problems develop during the batch
process it may be aborted by typing
"Control/C". To restart type "R SYSBAT".




^C
HELLO 1/2
Password:
1 other users are logged in under this account




.ASSIGN  SY:  ORSPAE

.R PIP.SAV
*TRL.OBJ,TTDVR.OBJ/D
?CAN'T FIND FILE OR ACCOUNT - FILE TRL    .OBJ - CONTINUING
?CAN'T FIND FILE OR ACCOUNT - FILE TTDVR  .OBJ - CONTINUING
*TRL.LST,TTDVR.LST/D
?CAN'T FIND FILE OR ACCOUNT - FILE TRL    .LST - CONTINUING
?CAN'T FIND FILE OR ACCOUNT - FILE TTDVR  .LST - CONTINUING
*RSTS.SAV,EMT.SAV,FIP.SAV,OVR.SAV,NSP.SAV,RJ2780.SAV/D
?CAN'T FIND FILE OR ACCOUNT - FILE RSTS   .SAV - CONTINUING
?CAN'T FIND FILE OR ACCOUNT - FILE EMT    .SAV - CONTINUING
?CAN'T FIND FILE OR ACCOUNT - FILE FIP    .SAV - CONTINUING
?CAN'T FIND FILE OR ACCOUNT - FILE OVR    .SAV - CONTINUING
?CAN'T FIND FILE OR ACCOUNT - FILE NSP    .SAV - CONTINUING
?CAN'T FIND FILE OR ACCOUNT - FILE RJ2780.SAV - CONTINUING
*RSTS.MAP,EMT.MAP,FIP.MAP,OVR.MAP,NSP.MAC,RJ2780.MAP/D
?CAN'T FIND FILE OR ACCOUNT - FILE RSTS   .MAP - CONTINUING
?CAN'T FIND FILE OR ACCOUNT - FILE EMT    .MAP - CONTINUING
?CAN'T FIND FILE OR ACCOUNT - FILE FIP    .MAP - CONTINUING
?CAN'T FIND FILE OR ACCOUNT - FILE OVR    .MAP - CONTINUING
?CAN'T FIND FILE OR ACCOUNT - FILE NSP    .MAP - CONTINUING
?CAN'T FILE FILE OR ACCOUNT - FILE RJ2780.MAP - CONTINUING
*RSTS.STB,EMT.STB,FIP.STB,OVR.STB,NSP.STB,RJ2780.STB/D
?CAN'T FIND FILE OR ACCOUNT - FILE RSTS   .STB - CONTINUING
?CAN'T FIND FILE OR ACCOUNT - FILE EMT    .STB - CONTINUING
```

```
?CAN'T FIND FILE OR ACCOUNT - FILE FIP    .STB - CONTINUING
?CAN'T FIND FILE OR ACCOUNT - FILE OVR    .STB - CONTINUING
?CAN'T FIND FILE OR ACCOUNT - FILE NSP    .STB - CONTINUING
?CAN'T FIND FILE OR ACCOUNT - FILE RJ2780.STB - CONTINUING
*^C

.R MACRO.SAV
*TRL,TRL/C=ORSPAE:COMMON,KERNEL,DK:CONFIG,ORSPAE;CHECK,TRL

ERRORS DETECTED: 0
FREE CORE: 9275. WORDS

*^C

.R MACRO.SAV
*TTDVR,TTDVR/C=ORSPAE:COMMON,KERNEL,DK:CONFIG,ORSPAE;CHECK,KBDEF,T
TDVR
ERRORS DETECTED: 0
FREE CORE: 8230. WORDS

*^C

.R LINK.SAV
*RSTS/Z,RSTS/A,RSTS=TRL,$ERR,STB/X/B:0/U:$1000/I/C
*ORSPAE:RSTS/C
*TTDVR
ROUND SECTION:
? MORBUF
LIBRARY SEARCH:
? BUF
? CR029
? DKSEEK
? DASFEK
?



*^C

.R LINK.SAV
*FMT/Z,FMT/A,FMT=ORSPAE:FMT,DK:RSTS,STB/X/B:$117000/U:$1000/C
*ORSPAE:RSTS
ROUND SECTION:
? FMTPAT



*^C

.R LINK.SAV
*FIP/Z,FIP/A,FIP=ORSPAE:FIP,DK:RSTS,STB/X/B:$117000/U:$1000/I/C
*ORSPAE:RSTS
ROUND SECTION:
? FIPPAT
LIBRARY SEARCH:
```

```
? OPY
?



*^C

.R LINK.SAV
*OVR/Z,OVR/A,OVR=ORSPAE:OVR,DK:FIP,STB,ORSPAE:RSTS/X/B:#1000


*^C

.R SILUS.SAV
*SY0:[0,1]RSTS.SIL=RSTS,EMT/M,FIP/M/C
*OVR/M/C
*ORSPAE:ODT,DEFALT
*^C

.R PIP.SAV
*BASIC.SAV,BASIC.STB/D
?CAN'T FIND FILE OR ACCOUNT - FILE BASIC .SAV - CONTINUING
?CAN'T FIND FILE OR ACCOUNT - FILE BASIC .STB - CONTINUING
*^C

.R LINK.SAV
*BASIC/Z,BASIC/TBRTS/A,BASIC=ORSPAE:RTS,DK:#ERR,STB/X/H:#177776/U:
#4000/C
*ORSPAE:MA2F/C
*ORSPAE:XL2F/C
*ORSPAE:XT2F/C
*ORSPAE:IO/C
*ORSPAE:PU/C
*ORSPAE:SN/C
*ORSPAE:TI/C
*ORSPAE:DA/C
*ORSPAE:VE
ROUND SECTION:
? PA


*^C

.R SILUS.SAV
*BASIC.RTS=BASIC
*^C

.R PIP.SAV
*SY0:[0,1]*.*/MODE:16.=BASIC/RTS
*^C

.R LOGOUT
Confirm: Y
Saved all disk files: 4624 blocks in use
Job 2 User 1,2 logged off KB2 at DD-MMM-YY HH:MM PM
```

```
1 other user still logged in under this account
System RSTS V06B-02 ACCEPTNCE TEST
Run time was 7 minutes, 32.1 seconds
Elapsed time was 10 minutes, 33 seconds
Good afternoon




BATCH JOB COMPLETED.
```

When the processing of RSTS and BASIC-PLUS are complete the batch processor will print BATCH JOB COMPLETED. At this point the RSTS.SIL and the BASIC RTS have been generated on the output media. To load the target system the system must be shut down and the new SIL must be installed and booted. The following steps should be taken.

```
.R UTILTY
 --------

*NO LOGINS
 ---------
*ZERO SY:[1,2] (This deletes unnecessary files created during SYSGEN)
 -------------
*SHUTUP
 ------

RSTS/E V06B-02 ACCEPTNCE TEST

OPTION:
```

Proceed to section E2360.

E2360 INSTALLING THE RSTS/E SIL

At this point the new monitor should be installed. To do this type

```
OPTION: INSTALL
        -------
   SIL: RSTS
        ----

OPTION:
```

Proceed to section E2500.

E2500 BUILDING THE SYSTEM AND OTHER DISKS

E2510 CURRENT PATCHES FOR RSTS/E V06B-02

The following patches are required as of the date of this manual.

RSTS/E V06B-02
RSTS/E INIT.SYS Program
INIT.SYS Program Patches

DC11'S AT PR5 ARE DISABLED - MANDATORY INIT.SYS PROGRAM PATCH

PROBLEM:

When the INIT.SYS program is loaded (booted), it scans all devices on the system for existence. All existing devices are forced to interrupt to determine their vector and interrupt priority. Any device that interrupts at a priority higher than expected is automatically disabled. The DC11 single line, modem control interface normally interrupts at PR5, but the INIT.SYS tables indicate that its maximum priority is PR4. This error causes all DC11's to be disabled.

SOLUTION:

This required patch corrects the problem by changing the INIT.SYS tables to indicate a maximum priority of PR5 for DC11's.

PROCEDURE:

1. This is a required patch to the INIT.SYS Program. Since patching the distribution medium is not recommended, the patch must be installed every time the COPY option is used to copy INIT.SYS from the distribution medium. Any copy of the <u>patched</u> INIT.SYS will propagate the fix.

2. When the COPY option has completed moving all files from the distribution medium, it automatically bootstraps the output disk. The PATCH option should be then used as shown below.

```
Option: PATCH
File to patch? INIT.SYS
Base address? 71472
Offset address? 403
 Base   Offset  Old       New?
071472  000403     000  ? 40
071472  000404     263  ? ^C            (CTRL/C to exit)

Option:
```

RSTS/E V06B-02
RSTS/E INIT.SYS Program
INIT.SYS Program Patches

DZ11'S ARE DISABLED - MANDATORY INIT.SYS PROGRAM PATCH

PROBLEM:

When the INIT.SYS program is loaded (booted), it scans all devices on the system for existence. All existing devices are forced to interrupt to determine their vector and interrupt priority. The routine that forces the DZ11 multiplexor to interrupt does not properly enable interrupts, and, therefore, the DZ11 never interrupts. This error causes all DZ11's to be disabled.

SOLUTION:

This required patch corrects the problem by correctly enabling interrupts on the DZ11 multiplexor.

PROCEDURE:

1. This is a required patch to the INIT.SYS Program. Since patching the distribution medium is not recommended, the patch must be installed every time the COPY option is used to copy INIT.SYS from the distribution medium. Any copy of the <u>patched</u> INIT.SYS will propagate the fix.

2. When the COPY option has completed moving all files from the distribution medium, it automatically bootstraps the output disk. The PATCH option should be then used as shown below.

```
Option: PATCH
File to patch? INIT.SYS
Base address? 25470
Offset address? 4024
 Base   Offset  Old       New?
025470  004024  020000  ? 40040
025470  004026  004736  ? ^C            (CTRL/C to exit)

Option:
```

RSTS/E V06B-02
RSTS/E INIT.SYS Program
INIT.SYS Program Patches

CRASH DUMP/AUTO-RESTART FAILS - MANDATORY INIT.SYS PROGRAM PATCH

PROBLEM:

The crash dump code fails to properly write out a CRASH.SYS file if the system disk is an RP04/05/06. In addition, the auto-restart code fails in attempting to reboot the INIT.SYS program after a power failure if the system disk is an RK06 or RP04/05/06.

SOLUTION:

This required patch corrects these problems by altering the crash dump/auto-restart disk driver code.

PROCEDURE:

1. This is a required patch to the INIT.SYS Program. Since patching the distribution medium is not recommended, the patch must be installed every time the COPY option is used to copy INIT.SYS from the distribution medium. Any copy of the patched INIT.SYS will propagate the fix.

2. When the COPY option has completed moving all files from the distribution medium, it automatically bootstraps the output disk. The PATCH option should be then used as shown below.

```
Option: PATCH
File to patch? INIT.SYS
Base address? 40:143750
Offset address? 7256
 Base   Offset  Old       New?
143750  007256  010125  ? 12704
143750  007260  012704  ? 157014
143750  007262  157016  ? 12425
143750  007264  012425  ? ^Z            (CTRL/Z for new offset)
Offset address? 7372
 Base   Offset  Old       New?
143750  007372  000207  ? 137
143750  007374  000207  ? 104226
143750  007376  004767  ? ^Z            (CTRL/Z for new offset)
Offset address? ^Z                     (CTRL/Z for new base)
```

```
RSTS/E V06B-02
RSTS/E INIT.SYS Program
INIT.SYS Program Patches


Base address? 104226
Offset address? 0
 Base    Offset  Old        New?
104226  000000  000000  ? 12700
104226  000002  000000  ? 23
104226  000004  000000  ? 12701
104226  000006  000000  ? 156
104226  000010  000000  ? 22737
104226  000012  000000  ? 41104
104226  000014  000000  ? 157022
104226  000016  000000  ? 1410
104226  000020  000000  ? 12700
104226  000022  000000  ? 3
104226  000024  000000  ? 12701
104226  000026  000000  ? 74
104226  000030  000000  ? 22737
104226  000032  000000  ? 46504
104226  000034  000000  ? 157022
104226  000036  000000  ? 1006
104226  000040  000000  ? 12725
104226  000042  000000  ? 1
104226  000044  000000  ? 13715
104226  000046  000000  ? 116306
104226  000050  000000  ? 60125
104226  000052  000000  ? 10025
104226  000054  000000  ? 207
104226  000056  000000  ? ^C              (CTRL/C to exit)

Option:
```

RSTS/E V06B-02
RSTS/E INIT.SYS Program
INIT.SYS Program Patches

REFRESH OPTION FAILS TO DELETE FILES - MANDATORY INIT.SYS PROGRAM PATCH

PROBLEM:

If the FILE suboption of the REFRESH option creates a file or changes a file's size and then attempts to delete another file without returning to the REFRESH SUBOPTION question, the deletion does not succeed.

SOLUTION:

This required patch corrects the problem by changing the initialization code.

PROCEDURE:

1. This is a required patch to the INIT.SYS Program. Since patching the distribution medium is not recommended, the patch must be installed every time the COPY option is used to copy INIT.SYS from the distribution medium. Any copy of the patched INIT.SYS will propagate the fix.

2. When the COPY option has completed moving all files from the distribution medium, it automatically bootstraps the output disk. The PATCH option should be then used as shown below.

```
Option: PATCH
File to patch? INIT.SYS
Base address? 134:117002
Offset address? 1622
 Base   Offset  Old       New?
117002  001622  105767  ? 4737
117002  001624  005704  ? 124136
117002  001626  001012  ? 240
117002  001630  004537  ? ^C            (CTRL/C to exit)

Option:
```

RSTS/E V06B-02
RSTS/E Executive
Monitor Patches

MESSAGE SEND BUFFER RELEASE - MANDATORY MONITOR PATCH

PROBLEM:

Under certain circumstances the message send code will fail to return a 16-word buffer to the system.

SOLUTION:

The following required monitor patch fixes the problem.

PROCEDURE:

1. This is a required patch to the RSTS/E V06B-02 executive. It must be installed in all target monitor SILs.

2. Bootstrap your system disk and use the PATCH option as shown below.

```
Option: PATCH
File to patch?  <lf>     (LINE FEED for installed monitor SIL)
Module name? EMT
Base address? CAL
Offset address? 2362
 Base    Offset  Old        New?
??????  002362  004537   ? 10004
??????  002364  ??????   ? 137
??????  002366  000763   ? FMTPAT
??????  002370  012737   ? ^Z              (CTRL/Z for new offset)
Offset address? ^Z                        (CTRL/Z for new base)
Base address? FMTPAT
Offset address? 0
 Base    Offset  Old        New?
120000  000000  000000   ? 4
120000  000002  000000   ? 40
120000  000004  000000   ? 4537
120000  000006  000000   ? REGRES
120000  000010  000000   ? 137
120000  000012  000000   ? CAL+2336
120000  000014  000000   ? ^C              (CTRL/C to exit)

Option:
```

RSTS/E V06B-02
RSTS/E Executive
Monitor Patches

NETWORK MESSAGE SEND FAILURE - MANDATORY MONITOR PATCH

PROBLEM:

The code which determines whether a message send is a local send or a network message send fails to detect network message sends.

SOLUTION:

The following required monitor patch fixes the problem.

PROCEDURE:

1. This is a required patch to the RSTS/E V06B-02 executive. It must be installed in all target monitor SILs.

2. Bootstrap your system disk and use the PATCH option as shown below.

```
Option: PATCH
File to patch?  <lf>     (LINE FEED for installed monitor SIL)
Module name? EMT
Base address? F8S
Offset address? 4056
 Base    Offset  Old        New?
??????  004056  003770   ? 2370
??????  004060  004737   ? ^C              (CTRL/C to exit)

Option:
```

RSTS/E V06B-02
RSTS/E Executive
Terminal Service Patches

JOB SET-UP FAILURE IF NO SMALL BUFFERS - MANDATORY TERMINAL SERVICE PATCH

PROBLEM:

When the job set-up code in the terminal service cannot create a job because there are no available small buffers, it should print a message (?No logins) on the terminal. In one special set of circumstances, the job set-up code will dispatch to the wrong address and cause a system crash.

SOLUTION:

This required terminal service patch corrects the problem.

PROCEDURE:

1. This is an required patch to the RSTS/E V06B-02 executive. It must be installed in all target monitor SIL's.

2. Answer the initial 'Offset address?' question with 42 (octal) if "echo control" was not included in the terminal service at system generation time. Otherwise answer with 56 (octal).

3. Bootstrap your system disk and use the PATCH option as shown below.

```
Option: PATCH
File to patch? <lf>      (LINE FEED for installed monitor SIL)
Module name? RSTS
Base address? (^MODCLK<>0)*12+ENDKEY
Offset address? 42                          (if no "echo control")
        -or-                                (   --see step 2--   )
Offset address? 56                          (if "echo control"   )
 Base   Offset  Old       New?
??????  0000??  004767  ? 4737
??????  0000??  ??????  ? PATCH
??????  0000??  012637  ? ^Z                (CTRL/Z for new offset)
Offset address? ^Z                          (CTRL/Z for new base)
Base address? PATCH
Offset address? 0
 Base   Offset  Old       New?
??????  000000  000000  ? 4737
??????  000002  000000  ? NEWJOB
??????  000004  000000  ? 207
```

M5

```
??????  000006  000000  ? °C              (CTRL/C to exit)
Option:
```

RSTS/E V06B-02
RSTS/F Executive
Terminal Service Patches

CONTROL/B IN ECHO CONTROL MODE - MANDATORY TERMINAL SERVICE PATCH

PROBLEM:

If a terminal has incoming escape sequences enabled and is being used in "echo control" mode, the character CONTROL/B is not ignored.

SOLUTION:

This required terminal service patch corrects the problem.

PROCEDURE:

1. This is an required patch to the RSTS/E V06B-02 executive. It must be installed in all target monitor SIL's.

2. Bootstrap your system disk and use the PATCH option as shown below.

```
Option: PATCH
File to patch? <lf>      (LINE FEED for installed monitor SIL)
Module name? RSTS
Base address? ..CTY.
Offset address? 114
  Base    Offset  Old        New?
??????  000114  032761   ? 4737
??????  000116  000040   ? PATCH+6
??????  000120  000010   ? 240
??????  000122  001003   ? ^Z              (CTRL/Z for new offset)
Offset address? ^Z                        (CTRL/Z for new base)
Base address? PATCH
Offset address? 6
  Base    Offset  Old        New?
??????  000006  000000   ? 32761
??????  000010  000000   ? 40
??????  000012  000000   ? 10
??????  000014  000000   ? 1003
??????  000016  000000   ? 132761
??????  000020  000000   ? 10
??????  000022  000000   ? 32
??????  000024  000000   ? 207
??????  000026  000000   ? ^C              (CTRL/C to exit)

Option:
```

RSTS/E V06B-02
RSTS/E Executive
Device Driver Patches

RK06 OVERLAPPED SEEK DRIVER TIMEOUTS - MANDATORY DEVICE DRIVER PATCH

PROBLEM:

The RK611/RK06 overlapped seek disk device driver fails to reenable controler interrupts in certain cases. This can cause the logging of timeout errors on one or more drives.

SOLUTION:

The following required device driver patch fixes the problem.

PROCEDURE:

1. This is an required patch to the RSTS/E V06B-02 executive. It must be installed in any target monitor SIL which is configured for RK06 disks using the overlapped seek driver.

2. Bootstrap your system disk and use the PATCH option as shown below.

```
Option: PATCH
File to patch? <lf>      (LINE FEED for installed monitor SIL)
Module name? RSTS
Base address? DMSEEK
Offset address? 234
 Base   Offset  Old       New?
??????  000234  001414  ? 240
??????  000236  012764  ? ^Z            (CTRL/Z for new offset)
Offset address? 366
 Base   Offset  Old       New?
??????  000366  000005  ? 105
??????  000370  004737  ? ^C            (CTRL/C to exit)

Option:
```

RSTS/E V06B-02
RSTS/E Executive
Device Driver Patches

RK06 DRIVER ERROR LOGGING - MANDATORY DEVICE DRIVER PATCH

PROBLEM:

The RK611/RK06 disk device driver (both non-overlapped seek and overlapped seek) fails to log all possible device error information in certain cases.

SOLUTION:

The following required device driver patch fixes the problem.

PROCEDURE:

1. This is an required patch to the RSTS/E V06B-02 executive. It must be installed in any target monitor SIL which is configured for RK06 disks.

2. Bootstrap your system disk and use the PATCH option as shown below.

```
Option: PATCH
File to patch? <if>      (LINF FEFD for installed monitor SIL)
Module name? RSTS
Base address? DMDSK
Offset address? 44
 Base   Offset  Old       New?
??????  000044  016703  ? 4737
??????  000046  ??????  ? PATCH+26
??????  000050  005002  ? ^Z              (CTRL/Z for new offset)
Offset address? ^Z                       (CTRL/Z for new base)
Base address? PATCH
Offset address? 26
 Base   Offset  Old       New?
??????  000026  000000  ? 13703
??????  000030  000000  ? CSR,DM
??????  000032  000000  ? 12713
??????  000034  000000  ? 100000
??????  000036  000000  ? 207
??????  000040  000000  ? ^C              (CTRL/C to exit)

Option:
```

RSTS/E V06B-02
RSTS/E BASIC-PLUS
BASIC-PLUS Patches

COMPILE WITH EXPLICIT EXTENSION - MANDATORY BASIC-PLUS PATCH

PROBLEM:

The BASIC-PLUS COMPILE command defaults the compiled file's extension to .BAC. If an explicit extension is typed to the COMPILE command this default should be overridden but it is not.

SOLUTION:

The following required BASIC-PLUS patch fixes the problem.

PROCEDURE:

1. This is a required patch to the BASIC-PLUS Run-Time System. It must be installed in all BASIC-PLUS Run-Time Systems.

2. Bootstrap your system disk and use the PATCH option as shown below. Answer the 'File to patch?' question with the name of the BASIC-PLUS Run-Time System to be patched.

```
Option: PATCH
File to patch? BASIC.RTS          (BASIC-PLUS Run-Time System name)
Base address? ED
Offset address? 4060
 Base    Offset  Old        New?
??????  004060  012764   ? 4737
??????  004062  000036   ? PA
??????  004064  000004   ? 1003
??????  004066  013764   ? ^Z              (CTRL/Z for new offset)
Offset address? ^Z                        (CTRL/Z for new base)
Base address? PA
Offset address? 0
 Base    Offset  Old        New?
??????  000000  000000   ? 12764
??????  000002  000000   ? 36
??????  000004  000000   ? 4
??????  000006  000000   ? 5764
??????  000010  000000   ? 14
??????  000012  000000   ? 207
??????  000014  ??????   ? ^C              (CTRL/C to exit)

Option:
```

Go to section F2530.

E2520 PATCHING THE RSTS/E SYSTEM--PATCH OPTION

The PATCH option of the RSTS/E Initialization Code provides a convenient means for altering the RSTS/E system code as bugs are found and corrections are published.

You can patch any file in account [0,1]. This account includes the initialization code (INIT.SYS) and any SIL. Patching makes permanent changes to the code on disk.

Patches take many different forms. Some are in-place patches to one or more words in one or more modules. Others require patch space in the affected areas. The RSTS/E monitor, initialization code, and run-time system always include patch space. You can patch the overlay code by using either free space in overlay segments or monitor patch space. Sometimes, patches affect fixed addresses and are straight forward; usually, however, the exact octal address of a patch varies from system to system. Published patches describe the procedures required to make the alterations correctly.

Use the PATCH option to perform patching operations. To invoke PATCH, type PATCH or PA. PATCH replies by asking for the name of the file to patch. Next, PATCH requests a MODULE NAME if the file is a SIL. Finally, for any file, it requests a BASE ADDRESS and an OFFSET ADDRESS.

The file to patch may or may not be a SIL. Since the initialization code cannot immediately distinguish a SIL from other files, you must make the distinction when you specify the filename. To distinguish a file that is not a SIL, append /N to its name.

The module name designates the SIL module to be patched. The base address defines the actual locations to be patched. For example, if you are patching the PRINT USING section of BASIC-PLUS, you can find its base address in the BASIC load map and enter that as the base address. The offset address is the first location to be patched relative to the specified base. For example, a PRINT USING patch may begin at an offset of 100 octal bytes from the beginning of PRINT USING.

Responses to the BASE ADDRESS? and OFFSET ADDRESS? queries can be valid octal numbers or expressions. Valid octal numbers are 0 to 177777, and leading zeros are optional. An expression is two octal numbers separated by a plus (+) or minus (-) sign, as in 173012+1026 and 43451-2077. When patching a SIL, you can substitute a global symbol name for an octal number anywhere. The load map for the module being patched contains the octal addresses and global symbol names for that module. A global symbol name must be one to six alphanumeric characters and must be defined in the symbol table for the current module. To refer to a global symbol in another module of the current SIL, type the symbol name followed by a slash and the name of the module in which the symbol is defined. For example, LOGIN/BASIC and DISK/ERR refer to the symbols LOGIN and DISK in the modules BASIC and ERR, respectively.

After you specify the base and offset addresses, PATCH opens the specified locations, prints the old contents, and accepts input. The table below summarizes the possible input.

Possible Input to the PATCH Option

Input		Meaning
octal number octal expression global symbol	RETURN	Enter the number, expression, or symbol as the new contents of the current location.
LINE FEED		Advance to the next location without altering the contents of the current location.
CTRL/Z		Return to the previous

question.

CTRL/C	Finish all patching and return to the OPTION? query.

PATCH makes the specified changes immediately after you type <CR>. Therefore, if you make an error, you must patch the location again to correct your mistake. To check that an entire patch is correct, use CTRL/Z to return to the MODULE NAME or BASE ADDRESS query. Type the same base and offset, then type the LINE FEED key to examine all the patched locations. If the old contents listed for any location do not match the published patch, restore the location to its original contents and try again to install the patch. The published patch may contain an error.

If you patch the initialization code, you must rebootstrap the system. Rebootstrapping loads the changed version of the initialization code into memory. To perform the bootstrap procedure, use the BOOT option, which is described in Section E2100.

The examples in the next section illustrate the use of PATCH.

E2521 PATCHING THE RSTS/F MONITOR--EXAMPLE

The following is an example of the use of the PATCH option to alter the RSTS/F monitor. Remember that when the patch is made, the monitor resides on disk but not in memory. The initialization code is in memory and the PATCH option allows you to change the disk image of the monitor before it is loaded into memory for normal timesharing.

```
OPTION: PATCH
MODULE NAME? RSTS              EXAMPLE ONLY--NOT A REAL PATCH
BASE ADDRESS? [NAME]
OFFSET ADDRESS? 120

MODULE   BASE   OFFSET   OLD      NEW?
------   ----   ------   ---      ----

RSTS     [NAME] 000120   104760   ? 004737
RSTS     [NAME] 000122   103364   ? [PATCH]+20 (OCTAL ADDITION)
RSTS     [NAME] 000124   005062   ? 102637
RSTS     [NAME] 000126   012762   ? [JBSTAT]-2 (OCTAL SUBTRACTION)
RSTS     [NAME] 000130   004737   ? <LF>  (NO CHANGE)
RSTS     [NAME] 000132   104726   ? 104730
RSTS     [NAME] 000134   010423   ? ^C    (CONTROL/C EXIT)

OPTION: PATCH
MODULE NAME? RSTS                      (FROM RSTS LOAD MAP)
BASE ADDRESS? [PATCH]
OFFSET ADDRESS?  20

MODULE   BASE   OFFSET   OLD      NEW?
```

```
------   ----   ------   ---      ----

RSTS     [PATCH] 000020  000000  ? 010203
RSTS     [PATCH] 000022  000000  ? 011104
RSTS     [PATCH] 000024  000000  ? 000207
RSTS     [PATCH] 000026  000000  ? ^C     (CONTROL/C EXIT)

OPTION:
```

All numbers printed by the PATCH option and all numeric responses are octal numbers. In the example, the notation [NAME] is used to indicate an address which must be found in a load map or a quantity which must be computed. PATCH will not perform any arithmetic. Hence, expression of the form [NAME]+20 must be manually calculated using 2's complement arithmetic. If you are not familiar with the octal representation of binary numbers or with 2's complement arithmetic, consult your Software Support Representative. As PATCH opens successive locations, it prints the current or old contents of the location and then waits for new data to be entered as an octal word. A carriage return <CR> is used to enter the new data. PATCH then sequences to the next location. A line feed <LF> with no data causes PATCH to sequence to the next location without altering the current location. PATCH continues to open successive locations until the CTRL/C combination is typed. CTRL/C returns to the initialization code OPTION query.

Note that changes are made immediately upon typing the carriage return key. If an error is made it will be necessary to reenter the PATCH option to correct the mistake. The printing of the old contents of a location provides one check on proper placement of a patch. If the old contents of any location shown in a published patch is not identical to that printed by the PATCH option, all locations should be restored to their old contents. This may indicate an error in using the load maps or an error in the published patch itself. Finally, a complete patch may be double checked by reentering the PATCH option and using the line feed key to examine successive locations.

E2522 PATCHING THE INITIALIZATION CODE--EXAMPLE

Patches to the Initialization Code are usually simpler since INIT is the same for all systems. There is usually no need to refer to a load map unless the value of a global parameter is needed for the patch. Recall once again that patches are made to the CIL on disk and not in core. This is an important distinction when patching INIT since the in-core copy (which is running) is not changed by the PATCH option. It is necessary to use the BOOT option (described fully in Section E2523) to load the altered INIT code into memory. The example below illustrates the procedure for making an INIT patch.

```
OPTION: PATCH          (EXAMPLE ONLY--NOT A REAL PATCH)
MODULE NAME? INIT
BASE ADDRESS? 67472
OFFSET ADDRESS? 4724

MODULE   BASE   OFFSET   OLD      NEW?
------   ----   ------   ---      ----

INIT     067472 004724   100200 ? 104200      (SINGLE WORD PATCH)
INIT     067472 004726   xxxxxx ? <LF>        (OLD CONTENTS VARIABLE)
INIT     067472 004730   005776 ? <LF>        (PRINTED FOR VERIFICA-
                                                TION ONLY)
INIT     067472 004732   001000 ? ^C          (CONTROL/C EXIT)
```

E2523 BOOT OPTION IF PATCH OPTION USED

```
OPTION: BOOT                  (Boot required to load altered)
        ----                  (Init code into memory)

BOOT DEVICE? <LF>             (Line feed boots the system disk)
             ----

RSTS V06B-02 ACCEPTNCE TEST

OPTION:
```

E2530 DISK INITIALIZATION--DSKINT OPTION

The operator is required to initialize all the disks. All the disk devices present and their numbers can be obtained from configuration questions and answers.

**The disk on which the system is now running has been initialized and should not be done again as this would destroy the system and RSTS sils.

E2532 PUBLIC DISKS--DSKINT

Operator: 1. Perform DSKINT for the remaining disks, all the public disks (RP, RB, RK, RM, RF or RS) then go to section F2540.

2. If there are any bad blocks uncovered during Pattern Test for public RF, RS or RK disks, then that disk should be fixed or replaced before proceeding further.

3. In FA+T environments type "PACK" for "PACK ID?", "MFD" for "MFD PASSWORD?", and "8" for "PATTERNS?".

DSKINT FOR RP, RB, RM, RF, RK AND RS DISKS.

```
     OPTION:  DSKINT
              ------
     DD-MMM-YY?  DD-MMM-YY          (Type DD-MMM-YY)
                 ---------
     HH:MM?  HH:MM                   (TYPE HH:MM)
             -----

     DISK?  DK or DP or DB or DS or DM or DF
            --    --    --    --    --    --
                                    (Name of the disk)
     UNIT?  1                       (Unit #)
            -
     PACK ID?  PACK
               ----
     PACK CLUSTER SIZE?  2          (4 for RP and RB disks, 8 for RP06
                         -           disks)
     SATT.SYS BASE?  <LF>
                     ----
     MFD PASSWORD?  MFD
                    ---
     MFD CLUSTER SIZE?  2           (4 for RP and RB disks, 8 for RP06
                        -           disks)
     PUB, PRI, SYS?  PUB            (Public disk)
                     ---
     CREATE LIBRARY ACCOUNT?  NO
                              --
     DATE LAST MODIFIED?  Y
                          --
     FORMAT?  NO                    (not asked for RS or RF disk)
              --
     PATTERNS?  2                   (1 for RP/RB disks, 8 for RS and RF
                -                    disks, 8 patterns required for all
                                     disks in FA + T.)
```

```
PROCFED (Y OR N)?  Y
                   -

STARTING FORMAT PASS
END FORMAT PASS

PATTERN #2
PATTERN #1

OPTION:
```

E2540 BUILDING THE SYSTEM FILES-REFRESH OPTION

By referencing the table below, determine whether or not system requires multi-level testing:

AMOUNT OF MEMORY	# OF COPIES OF UFTP	MULTI-LEVEL TEST
Less than 128	1	No
128K - 256K	2	Yes
256K - 512K	3	Yes
512K - 2000K	4	Yes

NOTE

Multi-level testing requires either an RP type disk or an RK06 type disk, (I.e., RK05's do not support multi-level testing.)

If multi-level testing is reqiurd go to section E2800, otherwise:

If system has no RF11's or RS03/04's, then go to section E2545.

If system has one RF11 or 1 RS03/04, then go to section E2541.

If system has two RF11's or 2 RS03/04's, then go to section E2542.

If system has three RF11's or 3 RS03/04's, then go to section E2543.

E2541 USING REFRESH-MOVING HEAD DISK WITH ONE RF11 OR RS03/04

```
OPTION:  REFRESH              (Type REFRESH)
         -------
DD-MMM-YY?  DD-MMM-YY         (<LF> IF OK)
            ---------          ----
HH:MM?  HH:MM                 (<LF> IF OK)
        -----                  ----
DISK?  DP, DB, DM OR DK       (CURRENT SYSTEM DISK)
       ----------------
UNIT?  0
       -
CLEAN?  YES
        ---
```

```
DISK IS BEING CLEANED - WAIT...
REFRESH SUBOPTION?  BADS
                    ----
BADS?  LIST
       ----
THERE ARE NO BAD BLOCKS        (This message or actual BAD
                                blocks will be printed)
BADS?  <LF>
REFRESH SUBOPTION?  CHANGE
                    ------
SWAP.SYS CHANGES?  YES
                   ---
DELETE?  <LF>
         ----
SIZE?  384 For RF11/RS03, 128 for RS04
       ---                ---
BASE?  See Table 1 below
SWAP0.SYS CHANGES?  <LF>
                    ----
SWAP1.SYS CHANGES?  <LF>
                    ----
SWAP3.SYS CHANGES?  <LF>
                    ----
OVR.SYS CHANGES?  <LF>
                  ----
ERR.SYS CHANGES?  <LF>
                  ----
BUFF.SYS CHANGES?  <LF>        (If DECtape on system answer YES)
                   ----        Size? = # dectape drives X3
BASE?  SEE TABLE 1 BELOW       (Asked only if dectape present)
CRASH.SYS CHANGES?  YES
                    ---
SIZE?  80
       --
BASE?  SEE TABLE 1 BELOW
OTHER FILES?  <LF>
              ----
REFRESH SUBOPTION?  LIST
                    ----
     (File status table is printed, operator should
      note that the status of all files are "OK".)

REFRESH SUBOPTION?  <LF>
                    ----
```

TABLE 1: Chose the appropriate base for the system disk from table below:

SYSTEM DISK	BASE TYPE-IN
RK05	2400
RK06	14000
RP02	20000

```
RP03              40000
RP04              80000
RP05              80000
RP06             160000


OPTION:  REFRESH
         -------
DD-MMM-YY?  DD-MMM-YY          (<LF> IF OK)
            ---------           ----
HH:MM?  HH:MM                  (<LF> IF OK)
        -----                   ----
DISK?  DF or DS
       --------
UNIT?  0
       -
CLEAN?  YES
        ---
REFRESH SUBOPTION?  BADS
                    ----
BADS?  LIST
       ----

THERE ARE NO BAD BLOCKS        (This message or actual BAD blocks
                                 will be printed)
BADS?  <LF>
       ----
REFRESH SUBOPTION?  CHANGE
                    ------
SWAP.SYS CHANGES?  <LF>
                   ----
SWAP0.SYS CHANGES?  YES
                    ---
SIZE?  600 for RS03, or RF11, 1600 for RS04
       ---                     ----
BASE?  <LF>
       ----
SWAP1.SYS CHANGES?  <LF>
                    ----
SWAP3.SYS CHANGES?  <LF>
                    ----
OVR.SYS CHANGES?  <LF>
                  ----
ERR.SYS CHANGES?  <LF>
                  ----
BUFF.SYS CHANGES?  <LF>
                   ----
CRASH.SYS CHANGES?  <LF>
                    ----
OTHER FILES?  <LF>
              ----
REFRESH SUBOPTION?  LIST
                    ----

     (File status table is printed, operator should
```

note that the status of all files are "OK".)

```
REFRESH SUBOPTION?  <LF>
                    ----
OPTION:  Proceed to section E2550.
```

E2542 USING REFRESH=MOVING HEAD DISK WITH TWO RF11's or RS03/04's

```
OPTION:  REFRESH                (Type REFRESH)
         -------
DD-MMM-YY?  DD-MMM-YY           (<LF> IF OK)
            ---------            ----
HH:MM?  HH:MM                   (<LF> IF OK)
        -----                    ----
DISK?  DP, DR, DM OR DK         (CURRENT SYSTEM DISK)
       ---------------
UNIT?  0
       -
CLEAN?  YES
        ---
DISK IS BEING CLEANED - WAIT...
REFRESH SUBOPTION?  BADS
                    ----
BADS?  LIST
       ----
THERE ARE NO BAD BLOCKS         (This message or actual BAD
                                 blocks will be printed)
BADS?  <LF>
       ----
REFRESH SUBOPTION?  CHANGE
                    ------
SWAP.SYS CHANGES?  YES
                   ---
DELETE?  <LF>
         ----
SIZE?  128 For RF11/RS03, 128 for RS04
       ---                ---
BASE?  See Table 1 below
SWAP0.SYS CHANGES?  <LF>
                    ----
SWAP1.SYS CHANGES?  <LF>
                    ----
SWAP3.SYS CHANGES?  <LF>
                    ----
OVR.SYS CHANGES?  <LF>
                  ----
ERR.SYS CHANGES?  <LF>
                  ----
BUFF.SYS CHANGES?  <LF>         (If DECtape on system answer YES)
                   ----          Size? = # dectape drives X3
BASE?  SEE TABLE 1 BELOW        (Asked only if dectape present)
CRASH.SYS CHANGES?  YES
```

```
                        ---
     SIZE?  80
            --
     BASE?  SEE TABLE 1 BELOW
     OTHER FILES?  <LF>
                   ----
     REFRESH SUBOPTION?  LIST
                         ----
          (File status table is printed, operator should
           note that the status of all files are "OK".)

     REFRESH SUBOPTION?  <LF>
                         ----
```

TABLE 1: Chose the appropriate base for the system disk from table below:

SYSTEM DISK	BASE TYPE-IN
RK05	2400
RK06	14000
RP02	20000
RP03	40000
RP04	80000
RP05	80000
RP06	160000

```
     OPTION:  REFRESH
              -------
     DD-MMM-YY?  DD-MMM-YY          (<LF> IF OK)
                 ---------           ----
     HH:MM?  HH:MM                  (<LF> IF OK)
             ------                  ----
     DISK?  DF or DS
            --------
     UNIT?  0
            -
     CLEAN?  YES
             ---
     REFRESH SUBOPTION?  BADS
                         ----
     BADS?  LIST
            ----
     THERE ARE NO BAD BLOCKS?       (This message or actual BAD
                                     blocks will be printed)
     BADS?  <LF>
            ----
     REFRESH SUBOPTION?  CHANGE
                         ------
     SWAP.SYS CHANGES?  <LF>
                        ----
     SWAP0.SYS CHANGES?  <LF>
```

```
                        ----
SWAP1.SYS CHANGES?  YES
                        ---
SIZE?  600 for RS03, or RF11, 1600 for RS04
       ---                      ----
BASE?  <LF>
       ----
SWAP3.SYS CHANGES?  <LF>
                    ----
OVR.SYS CHANGES?  <LF>
                  ----
ERR.SYS CHANGES?  <LF>
                  ----
BUFF.SYS CHANGES?  <LF>
                   ----
CRASH.SYS CHANGES?  <LF>
                    ----
OTHER FILES?  <LF>
              ----
REFRESH SUBOPTION?  LIST
                    ----

     (File status table is printed, operator should
      note that the status of all files are "OK".)

REFRESH SUBOPTION?  <LF>
                    ----
OPTION:  Proceed to section E2550
```

E2543 USING REFRESH-MOVING HEAD DISK WITH THREE RF11'S OR RS03/04'S

```
OPTION:  REFRESH                (Type REFRESH)
         -------
DD-MMM-YY?  DD-MMM-YY           (<LF> IF OK)
            ---------            ----
HH:MM?  HH:MM                   (<LF> IF OK)
        ----                     ----
DISK?  DP, DB, DM OR DK         (CURRENT SYSTEM DISK)
       ----------------
UNIT?  0
       -
CLEAN?  YES
        ---
DISK IS BEING CLEANED - WAIT...
REFRESH SUBOPTION?  BADS
                    ----
BADS?  LIST
       ----
THERE ARE NO BAD BLOCKS         (This message or actual BAD
                                 blocks will be printed)
BADS?  <LF>
REFRESH SUBOPTION?  CHANGE
```

```
                    ------
SWAP.SYS CHANGES?  YES
                    ---
DELETE?  <LF>
         ----
SIZE?  128 For RF11/RS03, 128 for RS04
       ---               ---
BASE?  See Table 1 below
SWAP0.SYS CHANGES?  <LF>
                    ----
SWAP1.SYS CHANGES?  <LF>
                    ----
SWAP3.SYS CHANGES?  <LF>
                    ----
OVR.SYS CHANGES?  <LF>
                  ----
FRP.SYS CHANGES?  <LF>
                  ----
BUFF.SYS CHANGES?  <LF>        (If DECtape on system answer YES)
                   ----         Size? = # dectape drives X3
BASE?  SEE TABLE 1 BELOW       (Asked only if dectape present)
CRASH.SYS CHANGES?  YES
                    ---
SIZE?  80
       --
BASE?  SEE TABLE 1 BELOW
OTHER FILES?  <LF>
              ----
REFRESH SUBOPTION?  LIST
                    ----
     (File status table is printed, operator should
      note that the status of all files are "OK".)

REFRESH SUBOPTION?  <LF>
                    ----
```

TABLE 1: Chose the appropriate base for the system disk from table below:

SYSTEM DISK	BASE TYPE-IN
RK05	2400
RK06	14000
RP02	20000
RP03	40000
RP04	80000
RP05	80000
RP06	160000

```
OPTION:  REFRESH
         -------
DD-MMM-YY?  DD-MMM-YY          (<LF> IF OK)
```

```
          ---------          ----
HH:MM?  HH:MM                (<LF> IF OK)
        ------               ----
DISK?  DF or DS
       --------
UNIT?  0
       -
CLEAN?  YES
        ---
REFRESH SUBOPTION?  BADS
                    ----
BADS?  LIST
       ----
THERE ARE NO BAD BLOCKS?      (This message or actual BAD
                               blocks will be printed)
BADS?  <LF>
       ----
REFRESH SUBOPTION?  CHANGE
                    ------
SWAP.SYS CHANGES?  <LF>
                   ----
SWAP0.SYS CHANGES?  <LF>
                    ----
SIZE?  600 for RS03, or RF11, 1600 for RS04
       ---                    ----
BASE?  <LF>
       ----
SWAP1.SYS CHANGES?  <LF>
                    ----
SWAP2.SYS CHANGES?  <LF>
                    ----
DVR.SYS CHANGES?  <LF>
                  ----
ERR.SYS CHANGES?  <LF>
                  ----
BUFF.SYS CHANGES?  <LF>
                   ----
CRASH.SYS CHANGES?  <LF>
                    ----
OTHER FILES?  <LF>
              ----
REFRESH SUBOPTION?  LIST
                    ----

     (File status table is printed, operator should
      note that the status of all files are "OK".)

REFRESH SUBOPTION?  <LF>
                    ----
OPTION:  REFRESH
         -------
DD-MMM-YY?  DD-MMM-YY         (<LF> IF OK)
            --------           ----
HH:MM?  HH:MM                 (<LF> IF OK)
        -----                  ----
```

```
DISK?  DF or DS
       --------
UNIT?  1
       -
CLEAN?  YES
        ---
REFRESH SUBOPTION?  BADS
                    ----
BADS?  LIST
       ----
THERE ARE NO BAD BLOCKS        (This message or actual BAD
                                blocks will be printed)
BADS?  <LF>
       ----
REFRESH SUBOPTION?  CHANGE
                    ------
SWAP.SYS CHANGES?  <LF>
                   ----
SWAP0.SYS CHANGES?  <LF>
                    ----
SWAP1.SYS CHANGES?  YES
                    ---
SIZE?  600 for RS03, or RF11, 1600 for RS04
       ---                    ----
BASE?  <LF>
       ----
SWAP3.SYS CHANGES?  <LF>
                    ----
DVP.SYS CHANGES?  <LF>
                  ---
ERR.SYS CHANGES?  <LF>
                  ----
BUFF.SYS CHANGES?  <LF>
                   ----
CRASH.SYS CHANGES?  <LF>
                    ----
OTHER FILES?  <LF>
              ----
REFRESH SUBOPTION?  LIST
                    ----

     (File status table is printed, operator should
      note that the status of all files are "OK".)

REFRESH SUBOPTION?  <LF>
                    ----
OPTION:  REFRESH
         -------
DD-MMM-YY?  DD-MMM-YY          (<LF> IF OK)
            --------            ----
HH:MM?  HH:MM                  (<LF> IF OK)
        -----                   ----
DISK?  DF or DS
       --------
UNIT?  2
```

K7

```
CLEAN?  YES
        ---
REFRESH SUBOPTION?  BADS
                    ----
BADS?  LIST
       ----
THERE ARE NO BAD BLOCKS        (This message or actual BAD
                                blocks will be printed)
BADS?  <LF>
       ----
REFRESH SUBOPTION?  CHANGE
                    ------
SWAP.SYS CHANGES?  <LF>
                   ----
SWAP0.SYS CHANGES?  <LF>
                    ----
SWAP1.SYS CHANGES?  <LF>
                    ----
SWAP3.SYS CHANGES?  YES
                    ---
SIZE?  600 for RS03, or RF11, 1600 for RS04
       ---                    ----
BASE?  <LF>
       ----
OVR.SYS CHANGES?  <LF>
                  ---
FPR.SYS CHANGES?  <LF>
                  ----
BUFF.SYS CHANGES?  <LF>
                   ----
CRASH.SYS CHANGES?  <LF>
                    ----
OTHER FILES?  <LF>
              ----
REFRESH SUBOPTION?  LIST
                    ----

     (File status table is printed, operator should
      note that the status of all files are "OK".)

REFRESH SUBOPTION?  <LF>
                    ----
OPTION:  Proceed to section E2550.


E2545  USING REFRESH MOVING HEAD DISK ONLY
       ----------------------------------

OPTION:  REFRESH                (Type REFRESH)
         -------
DD-MMM-YY?  DD-MMM-YY           (<LF> IF OK)
            ---------            ----
HH:MM?  HH:MM                   (<LF> IF OK)
```

```
        ----                          ----
DISK?  DP, DR, DM OR DK         (CURRENT SYSTEM DISK)
       ----------------
UNIT?  0
       -
CLEAN?  YES
        ---
DISK IS BEING CLEANED - WAIT...
REFRESH SUBOPTION?  BADS
                    ----
BADS?  LIST
       ----
THERE ARE NO BAD BLOCKS         (This message or actual BAD
                                 blocks will be printed)
BADS?  <LF>
       ----
REFRESH SUBOPTION?  CHANGE
                    ------
SWAP.SYS CHANGES?  YES
                   ---
DELETE?  <LF>
         ----
SIZE?  960
       ---
BASE?  See Table 1 below
SWAP0.SYS CHANGES?  <LF>
                    ----
SWAP1.SYS CHANGES?  <LF>
                    ----
SWAP3.SYS CHANGES?  <LF>
                    ----
OVR.SYS CHANGES?  <LF>
                  ----
FRR.SYS CHANGES?  <LF>
                  ----
BUFF.SYS CHANGES?  <LF>         (If DECtape on system answer YES)
                   ----          Size? = # dectape drives X3
BASE?  SEE TABLE 1 BELOW        (Asked only if dectape present)
CRASH.SYS CHANGES?  YES
                    ---
SIZE?  80
       --
BASE?  SEE TABLE 1 BELOW
OTHER FILES?  <LF>
              ----
REFRESH SUBOPTION?  LIST
                    ----
     (File status table is printed, operator should
      note that the status of all files are "OK".)

REFRESH SUBOPTION?  <LF>
                    ----

TABLE 1:  Chose the appropriate base for the system disk  from  table
below:
```

SYSTEM DISK	BASE TYPE-IN
RK05	2400
RK06	14000
RP02	20000
RP03	40000
RP04	80000
RP05	80000
RP06	160000

OPTION: Proceed to section E2550.

```
E2550  ESTABLISHING DEFAULT START UP CONDITIONS--DEFAULT OPTION
       ********************************************************

      OPTION:  DEFAULT
               -------
      NO DEFAULTS ARE CURRENTLY SET.

      YOU CURRENTLY HAVE: JOB MAX = 15, SWAP MAX = 8K

      JOB MAX OR SWAP MAX CHANGES?  YES
                                    ---
      NEW JOB MAX?  <LF>  (should be same as SYSGEN entry)
                    ----
      NEW SWAP MAX?  16
                     --

      YOU CURRENTLY HAVE: JOB MAX = 15, SWAP MAX = 16K

      JOB MAX OR SWAP MAX CHANGES?  <LF>
                                    ----

      RUN-TIME SYSTEM ? BASIC
                        -----

      ERROR MESSAGE FILE ? ERR
                           ---

      INSTALLATION NAME ? ACCEPTNCE TEST (In FA*T use DEC #)
                          --------------

Current memory allocation table is printed.

      MEMORY ALLOCATION BREAKDOWN:

      0000000 - 0117777 ( 20K) ; EXEC
      0120000 - 0207777 ( 14K) ; BASIC
      0210000 - 0437777 ( 38K) ; USER
      0440000 - END            ; NXM

      TABLE SUBOPTION?  PARITY        (Identify parity memory on
                        ------         system.  See Section E4240
                                       for complete information)
      PARITY REGISTER BREAKDOWN:
      ALL MEMORY IS 11/70 PARITY MEMORY

      TABLE SUBOPTION?  <LF>
                        ----
      YOU CURRENTLY HAVE  CRASH DUMP DISABLED.
      CRASH DUMP?  YES
                   ---
```

```
     MAGTAPE LABELLING DEFAULT (NONE): DOS
                                       ---
     OPTION:

OPERATOR, PROCEED TO SECTION E2555.


E2555  ESTABLISHING DEVICE CHARACTERISTICS - SET OPTION
       ------------------------------------------------

     OPTION: SET
             ---
     SET SUBOPTION ? LIST
                     ----
     DEVICE?  <LF>
              ----

This will list all devices and characteristics that are supported
by the currently installed SIL.

     SET SUBOPTION ? LP (If system does not have a line
                     -- printer type <LF> to this query and
                        proceed to Section E2560).

     UNIT?  0 (only asked if more than 1)
            --
     TYPE ? LP, LV or IS
            --  --    --
     WIDTH?  80, 120 or 132
             --  ---    ---
     LOWER CASE ? Y or N
                  -    -

     SET SUBOPTION ? <LF>
                     ----

     OPTION:


E2560  START SYSTEM DISK--START OPTION
       -------------------------------

     OPTION:  <LF>
              ----
     YOU CURRENTLY HAVE: JOB MAX = 15, SWAP MAX = 16K

     YOU CURRENTLY HAVE CRASH DUMP ENABLED

     DD-MMM-YY?  DD-MMM-YY        (Type current date or <LF> if OK)
                 ---------
     HH:MM?  HH:MM                (Type current time or <LF> if OK)
             -----
```

```
?Can't find file or account (These messages indicate
?Program lost-sorry          that RSTS/F is running
                             correctly)

Ready
```

At this point set the console switch register to 777777.

Proceed to Section E2570 to build the System Library.

E2570 BUILDING THE SYSTEM LIBRARY

Operator: Go to Section E2571, E2572, E2573 respectively if the system library is distributed on magtape, DECtape or RK cartridge disk.

E2571 USING MAGTAPE

Mount the RSTS/E SYSTEM LIBRARY #1 (DEC-11-OPS1A-E-MA9 for 9 track or -MA7 for 7 track) magtape reel on unit 0 with the write-enable ring removed.

Ensure that the FILE PROT indicator comes on.

Set the ON-LINE/OFF-LINE switch to ON-LINE and ensure that the RDY and LD PNT indicators are lit.

Go to Section F2580 and during the build dialogue replace XX by MT0 if TU10 or TS03 magtape or MM0 if TU16 or TU45 magtape.

E2572 USING DECTAPE

Mount the RSTS/E SYSTEM LIBRARY (DEC-11-ORSCA-E-UA1,2,3,13,14) DECtape reel on DECtape unit 0.

Set the REMOTE/OFF/LOCAL switch to REMOTE and the WRITE ENABLE/WRITE LOCK switch to WRITE LOCK on DECtape unit 0.

Go to Section E2580 and during the build dialogue replace XX by DT0.

E2573 USING RK DISK

Physically mount the library disk pack (DEC-11-ORSLAE-HA1 for RK05 or -RA for RK06) on an available drive. Insure that the RDY light is on and the WT PROT light is on. Type the following command to logically mount the disk...

```
MOUNT XXN:ORSLAE/RO <CR>
```

(where XX is DM for RK06, DK for RK05 and N is the unit number of the drive where the pack is physically mounted).

After the system types 'READY', go to section E2580. During the build dialogue, replace XX with the value that you used here for XXN.

E2500 BUILD DIALOGUE - SYSTEM LIBRARY

Proceed with the system library build as follows:

```
RUN XXN:BUILD8            (where XX=MT or MM or RP or DT or DK)
------------
                          (as mentioned in previous section)

BUILD   V06B-03 RSTS V06B-02 ACCEPTNCE TEST
SYSTEM BUILD? YES
           ---
TARGET SYSTEM DEVICE <SY0:>? <CR>
                             ----
SOURCE INPUT DEVICE? XXN:
                     ----
LIBRARY OUTPUT DEVICE <SY:>? <CR>
                             ----
LIBRARY ACCOUNT <[1,2]>? <CR>
                         ----
ASSIGN [1,2]
```

Normal library building proceeds without manual intervention. Many messages are printed as the library programs are compiled (ref E8300). When BUILD is finished the following messages are printed:

```
BUILD COMPLETE
Ready
```

The RSTS/E system is now ready for use. Proceed to section E2600 to load the UETP library account.

E2600 PREPARING THE UETP USER AREA

After RSTS/E is up and running, with all previously mentioned areas completed and checked, we may proceed to ready the system for the UETP control programs.

E2610 MOUNTING ALL PUBLIC DISKS

The operator must mount all the public disks on the system using the UTILTY program's MOUNT, CLEAN and UNLOCK commands. They are used as follows, where dev is DPn, DBn, DMn, DF, DSn, or DKn, where n is the unit # and ID is the PACK ID created during the DSKINT OPTION (Ref. Section F2532).

```
RUN $UTILTY
------------

UTILTY V06B-03 RSTS V06B-02 ACCEPTNCE TEST
(enter here if just completing E2620 with UETP on RK media)

$MOUNT dev:PACK          -Mount disk using this command
 --------------
$CLEAN dev:              -Clean the disk
 -----------
$UNLOCK dev:             -Unlock the disk
 -----------
```

Repeat the above MOUNT, CLEAN and UNLOCK commands for each public disk that was initialized with the DSKINT OPTION. When all disks have been mounted, proceed as follows:

```
$Z                       -Type Z when done
 --
```

Proceed to section E2620 to build the UETP Library.

F2620 BUILDING THE UETP ACCOUNT

Mount the supplied RSTS/E system library Test media on the appropriate device. Consult section F2800 for multi-level testing considerations. Proceed as follows:

```
RUN XXN:$BUILDR
---------------

RSTS/E UETP LIBRARY BUILDER V06-05
BUILD UETP ONTO WHICH SY: 1, ACCOUNT<44>?  <CR>   -defaults to 44
                                           ----
(this must be done for each copy of ACCTST that is run)
FROM WHICH DEVICE <MT0:>?  XXN
                           ---
LIBRARY ACCOUNT FOR DEVICE IS <(1,2)>?  <CR>      -defaults to 6
                                        ----
```

Where XXN is: MT0 for TU10 or TS03 magtape
MM0 for TU16 or TU45 magtape
DKY for RK05 disk
DMY for RK06 disk
DT0 for DECtape

Where Y is the available RK05, RK06 device e.g. DK1, DK2, DK3, DM1, DM2, DM3 etc.

The UETP library will now be built. Many messages will be printed on the console showing the actual build procedures (ref F8400). When BUILDR is finished the following message will be printed.

ALL UETP FILES FOR ACCOUNT [1,44] ARE LOADED (See NOTE 2 if RK media was used)

NOTE: 1. BUILDR must be run once for each of the RSTS/E system library DECtape containing UETP modules.

2. The UETP media must be dismounted and scratch media mounted

```
RUN $UTILTY <CR>
----------------
UTILTY V06B-03 RSTS V06B-02 ACCEPTNCE TEST
$DISMOUNT DKY:<CR>
 ----------------
$(now reference E2610 to mount the pack "left out"
previously, after physicaly removing UETP media and placing
previously removed pack back in drive)

$EXIT
 ----
Ready
```

F2621 PREPARING SYSTEM FOR UETP RUN

If system has no RF11's or RS03/04's, proceed to section F2625.

If system has one RF11 or one RS03/04, proceed to section E2622.

If system has two RF11's or two RS03/04's, proceed to section F2623.

If system has three RF11's or three RS03/04's, proceed to section E2624.

E2622 ONE RF11 OR ONE RS03/04

```
RUN $ UTILTY
-------------
UTILTY V06B-03 RSTS V06B-02 ACCEPTNCE TEST
* ADD SWAPFILE 0 DEV:  SWAP0.SYS
  ------------------------------
      WHERE DEV:  * DF0 FOR RF11
                    DS0 FOR RS03/04

*EXIT
 ----
Ready
```

Proceed to section E2625.

E2623 TWO RF11'S OR TWO RS03/04'S

```
RUN $ UTILTY
-----------
UTILTY V06B-03 RSTS V060-02 ACCEPTNCE TEST
* ADD SWAPFILE 0 DEV:  SWAP0.SYS
  ------------------------------
      WHERE DEV: * DF0 FOR RF11
                   DS0 FOR RS03/04

* ADD SWAPFILE 1 DEV:  SWAP1.SYS
  ------------------------------
      WHERE DEV: * DF1 FOR RF11
                   DS1 FOR RS03/04
*EXIT
 ----
Ready
```

Proceed to section E2625.

E2624 THREE RF11'S OR THREE RS03/04'S

```
RUN $ UTILTY
$
UTILTY V06B-03 RSTS V060-02 ACCEPTNCE TEST
* ADD SWAPFILE 0 DEV: SWAP0.SYS
*
        WHERE DEV: = DF0 FOR RF11
                     DS0 FOR RS03/04

* ADD SWAPFILE 1 DEV: SWAP1.SYS
*
        WHERE DEV: = DF1 FOR RF11
                     DS1 FOR RS03/04

* ADD SWAPFILE 3 DEV: SWAP3.SYS
*
        WHERE DEV: = DF2 FOR RF11
                     DS2 FOR RS03/04

*EXIT
$
Ready
```

Proceed to section E2625.

E2625 ENABLING LOGINS & START ERRINT

```
RUN $ UTILTY
$
UTILTY V06B-03 RSTS V06B-02 ACCEPTNCE TEST
* SET LOGINS 63
*
* LOGINS
*
If the system has RK05F's and an RK05J as the system disk then
type
* ADD LOGICAL DKn:DK1
*
Where n:  the second unit number of the last
          available RK05F disk drive
  Otherwise
* EXIT
$
Ready

RUN $ ERRINT
$
ERRINT V06B-03 RSTS V06B-02 ACCEPTNCE TEST
ERRLOG FILE IS 0% FULL
```

```
CHANGE SIZE TO <100>? <CR>
                     ----
UTILIZE CRASH FILE OUTPUT (YES/NO) <NO>? <CR>
                                       ----
DETACHING.....

HELLO 1/44;UETP
---------------
Ready
RUN $ ERRDIS
------------
ERRDIS V06B-03 RSTS V06B-02 ACCEPTNCE TEST

Input File <$ERRLOG.FIL>? <CR>
                          ----
Output to <KB:ERRDIS.OUT>? <CR>
                           ----
He[lp], Ba[d Blocks], Su[mmary] or Fu[ll] Report <Summary> <CR>
                                                           ----
List Bad Blocks (Yes/No) <Yes>?  <CR>
                                 ----
Zero Error File upon completion (Yes/No) <No>?  YES
                                                ---

ERRDIS Summary Report taken on 01-Dec-76, 09:46 AM
Input File: $ERRLOG.FIL   Output File: KB:ERRDIS.OUT

Reported Date/Time Range:
     30-Nov-76, 09:42:49 AM through 01-Dec-76, 12:37:01 AM

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

     PF PowerFail/Strtup            1/1

Total of 1 Errors Logged out of 1 Received 1 out of  100  Blocks  have
been used in $ERRLOG.FIL

List of Possible Bad Blocks
---------------------------
None Found

Input File <$ERRLOG.FIL> ^Z
                         -- Ready
```


K8

E2630 LOGGING INTO THE UETP ACCOUNT

```
HELLO  1,44                  - Login to system P,PN
PASSWORD:  UETP              - Type password (will not echo)
PSTS V06B-02 ACCEPTNCE TEST - PSTS prints message

READY
```

E2640 LOADING SCRATCH MEDIA

Now the user must check all devices for scratch (work) media (DECtapes and magtapes). All scratch media must be WRITE ENABLED.

Proceed to section E2700 to start the ACCTST control program.

E2650 MANUAL EXECUTION OF EXERCISERS

If less than 9 pseudo key boards are available, then it will be necessary to run each exerciser from the console terminal. Select the appropriate exerciser from section E3400 and execute for each device.

EXAMPLE:

```
RUN DKEXER
TEST WHICH RK DRIVE? n (unit number)
HOW MANY DK ITERATIONS? NN (1-99)
```

Repeat procedure for all devices.

E2700 UETP CONTROL PROGRAM--ACCTST

At this point you are ready to run the UETP control program called ACCTST which is a series of six BASIC-PLUS programs (ACCTST,AC1TST,AC2TST,AC3TST,AC4TST,AC5TST) which will run the reliability tests, the interactive mode tests, and the user simulation tests. These will all be run with no operator intervention.

At this point answer the questions regarding the test system configuration. If you desire not to use the line printer for log printouts then type "N" and six disks files (ACCLOG.LOG, AC1LOG.LOG, AC2LOG.LOG, AC3LOG.LOG, AC4LOG.LOG, AC5LOG.LOG) will be created on SY: and will contain the information that is normally output to the line printer. If the old log files are still to be preserved, the new log information will be appended to the old log files. If the operator desires the expanded job mix statistical output, it will be output to the log file at the end of each control program. If the operator desires the control programs to be continuous running, then answer yes to the continuous running test question. Consult section E2800 for multi-level testing consideration.

```
RUN ACCTST
**********
********** ACCTST V06-07
********** RSTS V06B-02 ACCEPTNCE TEST
********** RSTS/E UETP SYSTEM TEST PACKAGE

********** 23-SEP-76   20:51

********** PAPER TAPE READER TEST (Y OR N)?  N
********** PAPER TAPE PUNCH TEST (Y OR N)?  N
********** CARD READER TEST (Y OR N)?  N
********** KEYBOARD EXERCISER TEST (Y OR N)?  N
********** CUSTOMER ACCEPTANCE PROCEDURE (Y OR N)?  N
********** MULTI-LEVEL TESTING (Y OR N)?  N
********** TYPE NUMBER OF MAGTAPE DRIVES ON SYSTEM?  2
********** TYPE NUMBER OF DECTAPE DRIVES ON SYSTEM?  0
********** NO DECTAPE WORK
********** TYPE NUMBER OF RP03/RP02 DRIVES ON SYSTEM?  0
********** NO RP03 WORK
********** TYPE NUMBER OF RP04/RP05/RP06 DRIVES ON SYSTEM?  1
********** TYPE NUMBER OF RK03/05 DRIVES ON SYSTEM?  0
********** NO RK03/05 WORK
********** TYPE NUMBER OF RK06 DRIVES ON SYSTEM? 0
********** NO RK06 WORK
********** TYPE NUMBER OF RF11 PLATTERS ON SYSTEM? 0
********** NO RF11 WORK
********** TYPE NUMBER OF RS03/04 DRIVES ON SYSTEM? 1
********** TYPE NUMBER OF RX01 DRIVES ON SYSTEM?  0
*********  NO RX01 WORK
********** RUN AND ERROR LOGS TO LINE PRINTER (Y OR N)?  N
********** PRESERVE OLD RUN AND ERROR LOG FILES (Y OR N)?  N
********** EXPANDED JOB MIX STATISTICS (Y OR N)?  Y
```

```
********* CONTINUOUS RUNNING TEST (Y OR N)?  Y
********* OPTION PPE-TEST RUN (Y OR N)?  N
```

IF Interactive Test were requested (i.e., keyboard test, PPFXFR, etc.), attend system and answer any queries that are prompted on the console terminal.

The last query being:

```
CONTINUE WITH RELIABILITY TEST? YES
                               ---
```

Answer yes to the above query then The entire test will now run unattended. At this time insure that all drives are loaded with the proper scratch media.

If errors occur, on console printout, ref E2710 to "dump" actual errors and then ref E2720 to check for allowable media error criteria.

When you desire to stop ACCTST you may proceed to section E2710.

NOTE

Paper Tape, Card Reader and Keyboard facilities have been added (V06B RSTS/E release).

NOTE

If the UETP questions are answered such that the UFTP believes the disk structure consists of two or less RK05s the UETP will print an additional message indicating that parts of the system library will need to be deleted before the UETP can run. If yes is answered to proceed, the UETP will list those programs in account [1,2] that it deleted. The system library will have to be rebuilt after such a run.

On the console keyboard will be printed the current job statistics and all RSTS/E system errors detected by the individual test programs (See Section F4300). A normal job report printout is illustrated below.

```
********* SYSTEM RELIABILITY TESTS          14-OCT-76     21:39
********* CPU EXERCISER TESTS               21:39
21:39       RUN SCRIPT
21:39       STARTING SCRIPT V06-03
21:39       @LP:<Z2CTL.TMP
21:39       LINE 1 -- $CPEXER V06-03 RUNNING      21:39
21:39
21:54       LINE 1 -- $CPEXER FINISHED            21:54
21:54
21:54       FROM SCRIPT -- 1 IS FINISHED
21:54       FROM SCRIPT -- ALL CHANNELS ARE FINISHED
21:54
21:54
21:54       READY
21:54
21:54       RUN SCRIPT
21:54       STARTING SCRIPT V06-03
21:54       @LP:<Z3CTL.TMP
21:54       LINE 2 -- $CPEXER V06-03 RUNNING      21:54
21:54
21:54       LINE 1 -- $CPEXER V06-03 RUNNING      21:54
21:54
22:09       LINE 1 -- $CPEXER FINISHED            22:09
22:09
22:09       FROM SCRIPT -- 1 IS FINISHED
22:09       LINE 2 -- $CPEXER FINISHED                  22:09
22:09
22:09       FROM SCRIPT -- 2 IS FINISHED
22:09       FROM SCRIPT -- ALL CHANNELS ARE FINISHED
22:09
22:09
22:09       READY
22:09
```

Run and Error log information is sent to the line printer, if a yes is answered to the Run and Error log question, and contains information concerning the pseudo keyboard jobs. The third field contains the number of seconds since this job step was started.

```
LINE 1   0  -- HELLO
LINE 2   0  -- HELLO
LINE 2   0  -- 1/44
LINE 1   0  --
LINE 1   0  -- RSTS V06B-02 ACCEPTNCE TEST JOB 4  KB2  14-OCT-76 21:54
LINE 1   0  -- #1/44
LINE 2   0  --
LINE 2   1  -- PASSWORD:
LINE 2   1  -- 3 OTHER USER(S) ARE LOGGED IN UNDER THIS ACCOUNT
LINE 2   1  --
LINE 2   1  --
LINE 2   1  -- READY
LINE 2   1  --
LINE 2   1  -- RUN CPEXER
LINE 1   1  -- PASSWORD:
LINE 1   1  -- 2 OTHER USER(S) ARE LOGGED IN UNDER THIS ACCOUNT
LINE 1   1  --
LINE 1   1  --
LINE 1   1  -- READY
LINE 1   1  --
LINE 1   1  -- RUN CPEXER
LINE 1   1  -- 15
LINE 2   1  -- $CPEXER V06-03 RUNNING        21:54
LINE 2   2  --  # OF MINUTES? 15
LINE 2   3  --
LINE 1   3  -- $CPEXER V06-03 RUNNING        21:54
LINE 1   3  --  # OF MINUTES?
LINE 1   906  -- 0 ERRORS DETECTED - 13 PASSES COMPLETED
LINE 1   906  --
LINE 1   906  -- $CPEXER FINISHED              22:09
LINE 1   906  --
LINE 1   906  -- READY
LINE 1   906  --
LINE 1   906  !! (END OF RUN)
LINE 1   906  -- ^C
LINE 1   906  -- BYE/F
LINE 1   906  --
LINE 1   906  -- READY
LINE 1   906  --
LINE 2   907  --  0 ERRORS DETECTED - 13 PASSES COMPLETED
LINE 2   907  --
LINE 2   907  -- $CPEXER FINISHED              22:09
LINE 2   907  --
LINE 2   907  -- READY
LINE 2   907  --
LINE 2   907  !! (END OF RUN)
LINE 2   907  -- ^C
LINE 2   907  -- BYE/F
```

The log file also contains the output of the error display program. The ACCTST control programs automatically dump this report 3 times during the test run. This log is identical to the log output by the standard RSTS/E ERRDIS program with the addition of a disk I/O summary. Examples are shown below.

```
LINE 1   0  -- HELLO
LINE 1   1  -- 1/44
LINE 1   1  --
LINE 1   1  -- RSTS V06B-02 ACCEPTNCE TEST  JOB 5  KB2  14-OCT-76 06:06
LINE 1   2  -- #
LINE 1   2  -- PASSWORD:
LINE 1   2  -- 2 OTHER USER(S) ARE LOGGED IN UNDER THIS ACCOUNT
LINE 1   2  --
LINE 1   2  --
LINE 1   2  --
LINE 1   2  -- READY
LINE 1   2  --
LINE 1   2  -- RUN $ERRDIS
LINE 1   2  --
LINE 1   2  --
LINE 1   3  --
LINE 1   3  -- ERRDIS V06B-03 RSTS V06B-02 ACCEPTNCE TEST
LINE 1   3  --
LINE 1   3  --
LINE 1   3  --
LINE 1   A  -- Input File <$ERRLOG.FIL>?
LINE 1   A  -- Output to <KB:ERRDIS.OUT>?
LINE 1   A  -- Help, Summary or Full Report <SUMMARY>?
LINE 1   A  -- Zero Error File upon completion <NO>?
LINE 1   A  --
LINE 1   A  -- ERRDIS Summary Report taken on 14-Oct-76, 07:50 AM
LINE 1   A  -- Input File:  $ERRLOG.FIL          Output File:  KB:  ERRDIS.OUT
LINE 1   A  -- Reported Date/Time Range:
LINE 1   A  --        14-Oct-76, 07:48:19 AM through 14-Oct-76, 07:49:03 AM
LINE 1   A  --
LINE 1   A  --       ERROR             TOTAL                    UNIT NUMBERS
LINE 1   A  -- CODE-DESCRIPTION       REC/LOG    0    1    2    3    4    5    6    7
LINE 1   A  --
LINE 1   A  -- MM RH11/TM02/TU16       4/4            4
LINE 1   A  --
LINE 1   A  -- Total of 4 Errors Logged out of 4 Received
LINE 1   A  --
LINE 1   A  -- 2 out of 100 Blocks have been used in $ERRLOG.FIL
LINE 1   A  --
LINE 1   A  --
LINE 1   A  --  Z
LINE 1   A  --
LINE 1   A  --
LINE 1   A  --
LINE 1   A  --
LINE 1   A  --
LINE 1   A  --
LINE 1   A  --
LINE 1   A  -- Input File <$ERRLOG.FIL>?
LINE 1   A  --
```

```
LINE 1   16  --
LINE 1   16  --
LINE 1   16  --
LINE 1   16  -- Ready
LINE 1   16  --
LINE 1   16  -- BYE
LINE 1   16  -- Confirm: Y
LINE 1   16  -- Saved all disk files: 1228 blocks in use
LINE 1   16  -- Job 5 User 1,44 logged off KB2 at 14-Oct-76 07:56 AM
LINE 1   16  -- 2 other users still logged in under this account
LINE 1   16  -- System RSTS V06B-02 ACCEPTANCE TEST
LINE 1   16  -- Run time was 34.1 seconds
LINE 1   16  -- Elapsed time was 7 minutes, 6 seconds
LINE 1   16  -- Good morning
LINE 1   16  --
LINE 1   16  --
```

E2710 TERMINATING ACCTST

When ACCTST has completed a pass the following message will be printed on the console:

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

********* ACCTST PASS COMPLETE   DD-MMM-YY   HH:MM

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

If at least one of these messages appears on the console you must proceed as follows:

```
^C                          - TYPE CNTRL/C
^^
READY                       - RSTS/E prints READY


RUN $ERRDIS
ERRDIS  V06B-03 RSTS V06B-02 ACCEPTNCE TEST
INPUT FILE <$ERRLOG.FIL>? <CR>
                          ----
OUTPUT TO <KB:ERRDIS.OUT>? <CR>
                          ----
HE(LP), BA(D BLOCKS), SU(MMARY) OR FU(LL) REPORT <SUMMARY>? <CR>
                                                          ----
LIST BAD BLOCKS (YES/NO) <YES>? <CR>
                          ----
ZERO ERROR FILE UPON COMPLETION (YES/NO) <NO>? <CR>
                                          ----

ERRDIS SUMMARY REPORT TAKEN ON DD-MMM-YY, HH:MM PM
INPUT FILE: $ERRLOG.FIL          OUTPUT FILE: KB:ERRDIS.OUT
REPORTED DATE/TIME RANGE:
        DD-MMM-YY, HH:MM:SS PM THROUGH DD-MMM-YY, HH:MM:SS 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 200 BLOCKS HAVE BEEN USED IN $ERRLOG.FIL
```

```
LIST OF POSSIBLE BAD BLOCKS
***************************

NONE FOUND


INPUT FILE <$ERRLOG.FIL>? <CR>
                          ****
OUTPUT FILE <K9:ERRDIS.OUT>? <CR>
                             ****
HE[LP], BA[D BLOCKS], SU[MMARY] OR FU[LL] REPORT <SUMMARY>? FULL
                                                    ****
SPECIFIC ERROR TYPE <ALL>? <CR>
                           ****
STARTING DATE <FIRST ERROR>? <CR>
                             ****
ENDING DATE <LAST ERROR>? <CR>
                          ****
ZERO ERROR FILE UPON COMPLETION <NO>? <CR>
                                      ****
Full report will be printed on keyboard in detail.
INPUT FILE <$ERRLOG.FIL>? Z
                          **

     Ready
```

If the ACCTST log files were directed to the disk, proceed as follows. If not then go to Section E2900 to shutdown RSTS/E.

```
     RUN $PIP                  - run the PIP system program
     ********
     PIP V06B-03 RSTS V06B-02 ACCEPTNCE TEST

     xx:<yyy>LOG.LOG           - where xx is the output device and
     ***************           - where yyy is AC0,AC1,AC2,AC3,AC4 and
                                 AC5
```

Repeat the above command for each of the Log files.

Now proceed to Section E2900 to shut down RSTS/E. Consult section E2720 and the log file printouts to determine if the results are acceptable. In FA+T environments consult with Software Specialist for further testing and/or signoff.

E2720 ACCEPTABLE DATA RELIABILITY CRITERIA

Where the statistics and acceptable levels are supplied, recoverable error rates should be checked against the acceptable levels.

The following devices can be checked with the tables supplied here:

RK06
RK05
RP04/05/06
RX01

Words transferred and soft error information are supplied by RSTS UETP for comparison with the tables. Hard errors (device is aborted) or media errors (repetitive errors at the same media location) should not be included in the soft error count.

RK05 ACCEPTABLE RECOVERABLE DATA ERRORS

	NUMBER OF WORDS TRANSFERRED/DRIVE		ALLOWABLE SOFT ERRORS
	RK11-D	RK11-E	
Greater than	62,500,000	55,600,000	1
	250,000,000	222,300,000	2
	437,500,000	389,000,000	3
	625,000,000	555,700,000	4
	812,500,000	722,400,000	5
	1,000,000,000	889,100,000	6
	1,187,500,000	1,055,800,000	7

For each error beyond eight on the RK11-D, add 187,500,000 to the number of words required to be transferred for the drive to be acceptable.

For each error beyond eight on the RK11-E, add 166,700,000 to the number of words required to be transferred for the drive to be acceptable.

RP04 ACCEPTABLE RECOVERABLE DATA ERRORS

NUMBER OF WORDS READ/DRIVE		ALLOWABLE SOFT ERRORS
Greater than	62,500,000	1
	125,000,000	2
	187,500,000	3
	250,000,000	4
	312,500,000	5
	375,000,000	6
	437,500,000	7
	500,000,000	8

For each error beyond eight, add 62,500,000 to the number of words read for the drive to be acceptable.

RP04 ACCEPTABLE SEEK ERROR RATE

NUMBER OF SEEKS/DRIVE		ALLOWABLE SEEK ERRORS
Greater than	1,000,000	1
	2,000,000	2
	3,000,000	3
	4,000,000	4

For each seek error beyond four, add 1,000,000 to the number of seeks required for the drive to be acceptable.

RX01 ACCEPTABLE RECOVERABLE DATA ERRORS

NUMBER OF WORDS READ/DRIVE		ALLOWABLE SOFT ERRORS
Up to	62,500,000	1
	125,000,000	2
	187,500,000	3

250,000,000	4

RK06 ACCEPTABLE RECOVERABLE DATA ERRORS

	NUMBER OF WORDS TRANSFERRED/DRIVE	ALLOWABLE SOFT ERRORS PER DRIVE
Greater than:	62,500,000	1
	125,000,000	2
	187,000,000	3

	NUMBER OF SEEKS/DRIVE	ALLOWABLE SEEK ERRORS/DRIVES
Greater than:	1,000,000	1
	2,000,000	2
	3,000,000	3

E2800 MULTI-LEVEL TESTING CONSIDERATIONS

If no RF11's or RS03/04 are present, go to section E2810.

If one RF11 or RS03/04 are present, go to section E2820.

If two RF11's or RS03/04's are present, go to section E2830.

If three RF11's or RS03/04's are present, go to section E2840.

E2810 REFRESH-MULTIPLE COPIES USING MOVING HEAD DISK ONLY

```
OPTION:  REFRESH                (Type REFRESH)
         -------
DD-MMM-YY?  DD-MMM-YY           (<LF> IF OK)
            ---------            ----
HH:MM?   HH:MM                  (<LF> IF OK)
         -----                   ----
DISK?  DP, DB, DM OR DK         (CURRENT SYSTEM DISK)
       ----------------
UNIT?  0
       -
CLEAN?  YES
        ---
DISK IS BEING CLEANED - WAIT...

REFRESH SUBOPTION?  BADS
                    ----
BADS?  LIST
       ----
THERE ARE NO BAD BLOCKS         (This message or actual BAD
                                 blocks will be printed)
BADS?  <LF>
       ----
REFRESH SUBOPTION?  CHANGE
                    ------
SWAP.SYS CHANGES?  YES
                   ---
SIZE?  CHOOSE APPROPRIATE VALUE FROM TABLE

COPIES      SIZE
------      ----

  2         1600
  3         2240
  4         2880

BASE?  SEE TABLE 1 BELOW

SWAP0.SYS CHANGES?  <LF>
                    ----
```

```
SWAP1.SYS CHANGES?  <LF>
                    ----
SWAP3.SYS CHANGES?  <LF>
                    ----
OVR.SYS CHANGES?  <LF>
                  ----
FRP.SYS CHANGES?  <LF>
                  ----
BUFF.SYS CHANGES?  <LF>        (If DECtape on system answer YES)
                   ----         Size? = # dectape drives X3
BASE?  SEE TABLE 1 BELOW       (Asked only if dectape present)
CRASH.SYS CHANGES?  YES
                    ---
SIZE?  80
       --
BASE?  SEE TABLE 1 BELOW
OTHER FILES?  <LF>
              ----
REFRESH SUBOPTION?  LIST
                    ----

     (File status table is printed, operator should
      note that the status of all files are "OK".)

REFRESH SUBOPTION?  <LF>
                    ----
     TABLE 1:  Choose the appropriate base for the system disk from
     table below:

     DISK              BASE
     ----              ----
     RK06             14000
     RP02             20000
     RP03             40000
     RP04             80000
     RP05             80000
     RP06            160000

     OPTION:  Proceed to section E2550.


E2820  REFRESH-MULTIPLE COPIES USING MOVING HEAD DISK
       ---------------------------------------------
       AND ONE RF11 OR RS03/04
       -----------------------

     OPTION:  REFRESH              (Type REFRESH)
              -------
     DD-MMM-YY?  DD-MMM-YY         (<LF> IF OK)
                 ---------          ----
     HH:MM?  HH:MM                 (<LF> IF OK)
             -----                  ----
     DISK?  DP, DB, DM OR DK       (CURRENT SYSTEM DISK)
            ----------------
```

```
UNIT?  0
       -
CLEAN?  YES
        ---
DISK IS BEING CLEANED - WAIT...

REFRESH SUBOPTION?  BADS
                    ----
BADS?  LIST
       ----
THERE ARE NO BAD BLOCKS        (This message or actual BAD
                                blocks will be printed)
BADS?  <LF>
       ----
REFRESH SUBOPTION?  CHANGE
                    ------
SWAP.SYS CHANGES?  YES
                   ---
DELETE?  <LF>
         ----
SIZE?  CHOOSE APPROPRIATE VALUE FROM TABLE

COPIES      RF11/RS03      RS04
------      ---------      ----

  2            1025         128
  3            1664         640
  4            2304        1280

SWAP0.SYS CHANGES?  <LF>
                    ----
SWAP1.SYS CHANGES?  <LF>
                    ----
SWAP3.SYS CHANGES?  <LF>
                    ----
OVR.SYS CHANGES?  <LF>
                  ----
ERR.SYS CHANGES?  <LF>
                  ----
BUFF.SYS CHANGES?  <LF>        (If DECtape on system answer YES)
                   ----         Size? = # dectape drives X3
BASE?  SEE TABLE 1 BELOW       (Asked only if dectape present)
CRASH.SYS CHANGES?  YES
                    ---
SIZE?  80
       --
BASE?  SEE TABLE 1 BELOW
OTHER FILES?  <LF>
              ----
REFRESH SUBOPTION?  LIST
                    ----

     (File status table is printed, operator should
      note that the status of all files are "OK".)
```

```
REFRESH SUBOPTION? <LF>
                   ----
TABLE 1: Choose the appropriate base for the system disk from
table below:

DISK              BASE
----              ----
RK06             14000
RP02             20000
RP03             40000
RP04             80000
RP05             80000
RP06            160000

OPTION: REFRESH
        -------
DD-MMM-YY? DD-MMM-YY          (<LF> IF OK)
           ---------           ----
HH:MM? HH:MM                  (<LF> IF OK)
       ------                  ----
DISK? DF or DS
      --------
UNIT? 0
      -
CLEAN? YES
       ---
REFRESH SUBOPTION? BADS
                   ----
BADS? LIST
      ----
THERE ARE NO BAD BLOCKS?      (This message or actual BAD
                               blocks will be printed)
BADS? <LF>
      ----
REFRESH SUBOPTION? CHANGE
                   ------
SWAP.SYS CHANGES? <LF>
                  ----
SWAP0.SYS CHANGES? YES
                   ---
SIZE? 600 for RS03, or RF11, 1600 for RS04
      ---                    ----
BASE? <LF>
      ----
SWAP1.SYS CHANGES? <LF>
                   ----
SWAP3.SYS CHANGES? <LF>
                   ----
OVR.SYS CHANGES? <LF>
                 ----
ERR.SYS CHANGES? <LF>
                 ----
BUF.SYS CHANGES? <LF>
                 ----
```

```
     CRASH,SYS CHANGES?  <LF>
                         ----
     OTHER FILES?  <LF>
                   ----
     REFRESH SUBOPTION?  LIST
                         ----

          (File status table is printed, operator should
           note that the status of all files are "OK".)

     REFRESH SUBOPTION?  <LF>
                         ----
     OPTION:  PROCEED TO SECTION E2550


E2830  REFRESH-MULTIPLE COPIES USING MOVING HEAD DISK
       ----------------------------------------------
       WITH TWO RF11'S OR RS03/04'S
       ----------------------------

     OPTION:  REFRESH                (Type REFRESH)
              -------
     DD-MMM-YY?  DD-MMM-YY           (<LF> IF OK)
                 ---------            ----
     HH:MM?  HH:MM                   (<LF> IF OK)
             -----                    ----
     DISK?  DP, DB, DM OR DK         (CURRENT SYSTEM DISK)
            ---------------
     UNIT?  0
            -
     CLEAN?  YES
             ---
     DISK IS BEING CLEANED - WAIT...

     REFRESH SUBOPTION?  BADS
                         ----
     BADS?  LIST
            ----
     THERE ARE NO BAD BLOCKS         (This message or actual BAD
                                      blocks will be printed)
     BADS?  <LF>
            ----
     REFRESH SUBOPTION?  CHANGE
                         ------
     SWAP,SYS CHANGES?  YES
                        ---
     DELETE?  <LF>
              ----
     SIZE?  CHOOSE APPROPRIATE VALUE FROM TABLE

     COPIES      RF11/RS03      RS04
     ------      ---------      ----

       2            1025         128
```

```
  3            1664          640
  4            2304         1280

BASE?  SEE TABLE 1 BELOW

SWAP0.SYS CHANGES?  <LF>
                    ----
SWAP1.SYS CHANGES?  <LF>
                    ----
SWAP3.SYS CHANGES?  <LF>
                    ----
OVR.SYS CHANGES?  <LF>
                  ----
ERR.SYS CHANGES?  <LF>
                  ----
BUFF.SYS CHANGES?  <LF>          (If DECtape on system answer YES)
                   ----           Size? = # dectape drives X3
BASE?  SEE TABLE 1 BELOW         (Asked only if dectape present)
CRASH.SYS CHANGES?  YES
                    ---
SIZE?  80
       --
BASE?  SEE TABLE 1 BELOW
OTHER FILES?  <LF>
              ----
REFRESH SUBOPTION?  LIST
                    ----

     (File status table is printed, operator should
      note that the status of all files are "OK".)

REFRESH SUBOPTION?  <LF>
                    ----
TABLE 1:  Choose the appropriate base for the system disk from
table below:

DISK              BASE
----              ----
RK06             14000
RP02             20000
RP03             40000
RP04             80000
RP05             80000
RP06            160000

OPTION:  REFRESH
         -------
DD-MMM-YY?  DD-MMM-YY          (<LF> IF OK)
            ---------           ----
HH:MM?  HH:MM                  (<LF> IF OK)
        ------                  ----
DISK?  DF or DS
       --------
UNIT?  0
       -
```

```
CLEAN?  YES
        ---
REFRESH SUBOPTION?  BADS
                    ----
BADS?  LIST
       ----
THERE ARE NO BAD BLOCKS?        (This message or actual BAD
                                 blocks will be printed)
BADS?  <LF>
       ----
REFRESH SUBOPTION?  CHANGE
                    ------
SWAP.SYS CHANGES?  <LF>
                   ----
SWAP0.SYS CHANGES?  YES
                    ---
SIZE?  600 for RS03, or RF11, 1600 for RS04
       ---                    ----
BASE?  <LF>
       ----
SWAP1.SYS CHANGES?  <LF>
                    ----
SWAP3.SYS CHANGES?  <LF>
                    ----
OVR.SYS CHANGES?  <LF>
                  ----
ERR.SYS CHANGES?  <LF>
                  ----
BUF.SYS CHANGES?  <LF>
                  ----
CRASH.SYS CHANGES?  <LF>
                    ----
OTHER FILES?  <LF>
              ----
REFRESH SUBOPTION?  BADS
                    ----

     (File status table is printed, operator should
      note that the status of all files are "OK".)

REFRESH SUBOPTION?  <LF>
                    ----
OPTION:  REFRESH
         -------
DD-MMM-YY?  DD-MMM-YY          (<LF> IF OK)
            --------            ----
HH:MM?  HH:MM                  (<LF> IF OK)
        -----                   ----
DISK?  DF or DR
       --------
UNIT?  1
       -
CLEAN?  YES
        ---
REFRESH SUBOPTION?  LIST
```

```
                    ----
BADS?  LIST
       ----
THERE ARE NO BAD BLOCKS        (This message or actual BAD
                                blocks will be printed)
BADS?  <LF>
       ----
REFRESH SUBOPTION?  CHANGE
                    ------
SWAP.SYS CHANGES?  <LF>
                   ----
SWAP0.SYS CHANGES?  <LF>
                    ----
SWAP1.SYS CHANGES?  <LF>
                    ----
SIZE?  600 for RS03, or RF11, 1600 for RS04
       ---                    ----
BASE?  <LF>
       ----
SWAP3.SYS CHANGES?  <LF>
                    ----
OVR.SYS CHANGES?  <LF>
                  ---
ERR.SYS CHANGES?  <LF>
                  ----
BUF.SYS CHANGES?  <LF>
                  ----
CRASH.SYS CHANGES?  <LF>
                    ----
OTHER FILES?  <LF>
              ----
REFRESH SUBOPTION?  LIST
                    ----

     (File status table is printed, operator should
      note that the status of all files are "OK".)

REFRESH SUBOPTION?  <LF>
                    ----
OPTION:  Proceed to section E2550.


E2540  REFRESH-MULTIPLE COPIES USING MOVING HEAD DISK
       ---------------------------------------------
       WITH THREE RF11'S OR RS03/04'S
       ------------------------------

OPTION:  REFRESH              (Type REFRESH)
         -------
DD-MMM-YY?  DD-MMM-YY         (<LF> IF OK)
            ---------          ----
HH:MM?  HH:MM                 (<LF> IF OK)
        -----                  ----
DISK?  DP, DB, DM OR DF       (CURRENT SYSTEM DISK)
```

```
        ---------------
UNIT?  0
        -
CLEAN?  YES
        ---
DISK IS BEING CLEANED - WAIT...

REFRESH SUBOPTION?  BADS
                    ----
BADS?  LIST
       ----
THERE ARE NO BAD BLOCKS        (This message or actual BAD
                                blocks will be printed)
BADS?  <LF>
       ----
REFRESH SUBOPTION?  CHANGE
                    ------
SWAP.SYS CHANGES?  YES
                   ---
DELETE?  <LF>
         ----
SIZE?  CHOOSE APPROPRIATE VALUE FROM TABLE

COPIES      RF11/RS03      RS04
------      ---------      ----

  2            1025         128
  3            1664         640
  4            2304        1280

BASE?  SEE TABLE 1 BELOW

SWAP0.SYS CHANGES?  <LF>
                    ----
SWAP1.SYS CHANGES?  <LF>
                    ----
SWAP3.SYS CHANGES?  <LF>
                    ----
OVR.SYS CHANGES?  <LF>
                  ----
ERR.SYS CHANGES?  <LF>
                  ----
BUFF.SYS CHANGES?  <LF>          (If DECtape on system answer YES)
                   ----           Size? = % dectape drives X3
BASE?  SEE TABLE 1 BELOW        (Asked only if dectape present)
CRASH.SYS CHANGES?  YES
                    ---
SIZE?  80
       --
BASE?  SEE TABLE 1 BELOW
OTHER FILES?  <LF>
              ----
REFRESH SUBOPTION?  LIST
                    ----
```

```
     (File status table is printed, operator should
      note that the status of all files are "OK".)

REFRESH SUBOPTION?  <LF>
                    ----
TABLE 1:  Choose the appropriate base for the system disk from
table below:

DISK                 BASE
----                 ----
RK06                14000
RP02                20000
RP03                40000
RP04                80000
RP05                80000
RP06               160000

OPTION:  REFRESH
         -------
DD-MMM-YY?  DD-MMM-YY          (<LF> IF OK)
            ---------           ----
HH:MM?  HH:MM                  (<LF> IF OK)
        ------                  ----
DISK?  DF or DS
       --------
UNIT?  0
       -
CLEAN?  YES
        ---
REFRESH SUBOPTION?  BADS
                    ----
BADS?  LIST
       ----
THERE ARE NO BAD BLOCKS?       (This message or actual BAD
                                blocks will be printed)
BADS?  <LF>
       ----
REFRESH SUBOPTION?  CHANGE
                    ------
SWAP.SYS CHANGES?  <LF>
                   ----
SWAP0.SYS CHANGES?  YES
                    ---
SIZE?  600 for RS03, or RF11, 1600 for RS04
       ---                    ----
BASE?  <LF>
       ----
SWAP1.SYS CHANGES?  <LF>
                    ----
SWAP3.SYS CHANGES?  <LF>
                    ----
OVR.SYS CHANGES?  <LF>
                  ----
ERR.SYS CHANGES?  <LF>
                  ----
```

```
BUF.SYS CHANGES?  <LF>
                  ----
CRASH.SYS CHANGES?  <LF>
                    ----
OTHER FILES?  <LF>
              ----
REFRESH SUBOPTION?  LIST
                    ----

     (File status table is printed, operator should
      note that the status of all files are "OK".)

REFRESH SUBOPTION?  <LF>
                    ----
OPTION:  REFRESH
         -------
DD-MMM-YY?  DD-MMM-YY          (<LF> IF OK)
            --------            ----
HH:MM?  HH:MM                  (<LF> IF OK)
        -----                   ----
DISK?  DF or DS
       --------
UNIT?  1
       -
CLEAN?  YES
        ---
REFRESH SUBOPTION?  BADS
                    ----
BADS?  LIST
       ----
THERE ARE NO BAD BLOCKS        (This message or actual BAD
                                blocks will be printed)
BADS?  <LF>
       ----
REFRESH SUBOPTION?  CHANGE
                    ------
SWAP.SYS CHANGES?  <LF>
                   ----
SWAP0.SYS CHANGES?  <LF>
                    ----
SWAP1.SYS CHANGES?  <LF>
                    ----
SIZE?  600 for RS03, or RF11, 1600 for RS04
       ---                     ----
BASE?  <LF>
       ----
SWAP3.SYS CHANGES?  <LF>
                    ----
OVR.SYS CHANGES?  <LF>
                  ---
ERR.SYS CHANGES?  <LF>
                  ----
BUF.SYS CHANGES?  <LF>
                  ----
CRASH.SYS CHANGES?  <LF>
```

```
                    ----
OTHER FILES?  <LF>
              ----
REFRESH SUBOPTION?  LIST
                    ----

     (File status table is printed, operator should
      note that the status of all files are "OK".)

REFRESH SUBOPTION?  <LF>
                    ----
OPTION:  REFRESH
         -------
DD-MMM-YY?  DD-MMM-YY          (<LF> IF OK)
            ---------           ----
HH:MM?  HH:MM                  (<LF> IF OK)
        -----                   ----
DISK?  DF or DS
       --------
UNIT?  2
       -
CLEAN?  YES
        ---
REFRESH SUBOPTION?  BADS
                    ----
BADS?  LIST
       ----
THERE ARE NO BAD BLOCKS        (This message or actual BAD
                                blocks will be printed)
BADS?  <LF>
       ----
REFRESH SUBOPTION?  CHANGE
                    ------
SWAP.SYS CHANGES?  <LF>
                   ----
SWAP0.SYS CHANGES?  <LF>
                    ----
SWAP1.SYS CHANGES?  <LF>
                    ----
SWAP3.SYS CHANGES?  YES
                    ---
SIZE?  600 for RS03, or RF11, 1600 for RS04
       ---                     ----
BASE?  <LF>
       ----
OVR.SYS CHANGES?  <LF>
                  ----
ERR.SYS CHANGES?  <LF>
                  ----
BUFF.SYS CHANGES?  <LF>
                   ----
CRASH.SYS CHANGES?  <LF>
                    ----
OTHER FILES?  <LF>
              ----
```

```
REFRESH SUBOPTION?  LIST
                    ----

     (File status table is printed, operator should
      note that the status of all files are "OK".)

REFRESH SUBOPTION?  <LF>
                    ----
OPTION:  Proceed to section F2550.
```

E2900 PERFORMING SYSTEM SHUT DOWN - SHUTUP

The shut down procedures for the RSTS/E VETP system are critically important. If system shut down is not conducted in an orderly and careful fashion, much valuable user data can be irretrievably lost.

The SHUTUP system program can be run only from the console terminal. The program is stored in its compiled form in the system library with protection code <124>. The following sample dialogue shows the use of SHUTUP.

```
RUN $SHUTUP
SHUTUP   V06B-03 RSTS   V06B-02   ACCEPTNCE TEST

********   Set-up Dialogue Phase  ********

Type 'ESC('ALT') to any query to backup one (1) step

'OPSER' not running

Minutes until system shutdown (0-99) <5>?  0
                                           --

********  Warning Message Phase  ********
Further LOGINs are now disabled
0 minute warning message sent

********  Initial Job Killing Phase  ********

********  'ERRCPY' Shutdown Phase  ********

********  Unload/Remove PTS Phase  ********

********  SWAP File Removal Phase  ********

********  Disk DISMOUNT Phase  ********

********  Final Shutdown Phase  ********

Please wait for system to re-boot itself



RSTS V06B-02 ACCEPTNCE TEST (DK0)


Option:
```

When SHUTUP runs, it prints its header line, followed, on a second line, by the first of two queries. The first query asks how long a time is necessary before the system can be shut down, and the second

query asks the intervals between warning messages. After the two queries are answered, the SHUTUP program proceeds with its action. Further logins are disabled to prevent more users from entering the system. Messages are sent to all on-line terminals and pseudo keyboards at the interval specified by the operator. Each message tells how many minutes are left until the system shutdown. When no time is left, all terminals and pseudo keyboards still logged into the system are automatically logged out. Jobs still active are terminated by the KILL action. All non-system disks are then dismounted. When SHUTUP terminates, it actually reboots the system and the OPTION query appears on the system console.

E2910 RESTARTING RSTS/E AFTER A SHUTDOWN

To re-boot RSTS/E follow the appropriate bootstrap procedure as described in Section E2100, Then proceed as follows:

```
     RSTS V06B-02 ACCEPTNCE TEST

     OPTION:  <LF>
              ----
     SYSTEM DISK IS BEING CLEANED - WAIT ...

     YOU CURRENTLY HAVE: JOB MAX = 15, SWAP MAX = 16K,

     YOU CURRENTLY HAVE CRASH DUMP ENABLED

     DD-MMM-YY? DD-MMM-YY (Type current date or <LF> if OK)
                ---------
     HH:MM?   HH:MM (Type current time or <LF> if OK)
              -----

     SYSTEM INITIALIZATION PROGRAM          V06B-03

     COMMAND FILE NAME?  ^C
                         --
     Ready

     To mount all public disks see section E2610,

     Reference section E2621,

     HELLO 1,44                -Login
     ----------
     PASSWORD:  UETP           -Password will not echo
                ----
```

Proceed to section E2640 to load scratch media and run ACCTST,

E2020 RECOVERING FROM A SYSTEM CRASH

Whenever a trap occurs to vector 4 or vector 10, the system distinguishes the trap as one of two categories: it is either,

a. A catastrophic error which affects only one particular user, or

b. A system crash for which some software or hardware problem is possibly responsible. The handling of system crashes is treated below.

The handling of catastrophic errors is as follows. The system determines which user was responsible for the error-trap. It flags that user's job with a special code which causes the system to reinitialize that user's job area completely when it is next his turn to run. The system prints on that user's terminal the message

```
CATASTROPHIC ERROR
PROGRAM LOST-SORRY
```

The reinitialized user is in the same state as he would be if he had just logged into the system. The system resumes normal time sharing operations.

When the system detects a condition from which it cannot recover, it performs an automatic restart only if both of two conditions are fulfilled:

a. The crash-dump facility must have been enabled at system start up time (possible only when the CRASH.SYS file exists), and

b. The CPU's Switch Register must currently be set to 777777.

If either condition is not fulfilled, the system does not take the automatic restart path but simply halts at address 54.

If the system halts at address 54, the operator may choose one of two procedures.

a. He depresses the CPU Console CONTinue switch, which causes the system to be bootstrapped into normal system start up mode.

b. The operator starts the CPU at address 52 with CPU Switch Register set to 777777. This causes the system first to write the contents of memory onto the CRASH.SYS file (provided the crash-dump facility had been enabled) and then to be bootstrapped from disk in the special automatic restart mode described below.

If the system takes the automatic restart path, no halt occurs. Instead, the system first writes the critical contents of memory into the CRASH.SYS file and then bootstraps itself into memory from the system disk. After the system has been bootstrapped into memory, control jumps to the initialization routines. At this point the system recognizes the fact that it was not activated through a normal system start up but rather through an automatic restart and consequently initializes itself in automatic restart mode. If two system crashes occur within the same minute (more accurately stated, two error-traps within the same minute), the system halts at address 54. This protects the system against an infinite loop of error-traps caused by some repeating hardware malfunction.

When the system is initialized in automatic restart mode, control by-passes all parts of the initialization code which call for operator intervention and initializes the system using information already stored in memory. A sample printout follows:

```
SYSTEM HAS BEEN RELOADED; ATTEMPTING AUTO-RESTART.
YOU CURRENTLY HAVE: JOBMAX=15, SWAPMAX=16K
YOU CURRENTLY HAVE CRASH DUMP ENABLED
RSTS/E V06B-02 ACCEPTNCE TEST
SYSTEM INITIALIZATION PROGRAM        V06B-03
DISK IS BEING CLEANED - WAIT ...

INIT  V06B-03 RSTS V06B-02 ACCEPTANCE TEST
TYPE ^C AS SOON AS THE ABOVE MESSAGE IS
TYPED ON THE CONSOLE.

Ready
```

The operator must now MOUNT, and CLEAN each public disk that was mounted on the system at the time of the crash. Proceed as follows:

```
RUN $UTILTY              -use the system UTILITY program
-----------
UTILTY V06B-03 RSTS  V06B-02 ACCEPTNCE TEST
$MOUNT dev:PACK          -mount disk where dev is the disk
 ------------            -type and id is the pack id
$CLEAN dev:              -clean disk
 ----------
$UNLOCK DEV:
 -----------
```

Repeat the above three commands for each disk, when all disks are completed, type:

```
$EXIT
 ----
Ready
```

Proceed to section F2621 to get the system back on the air and restart the UFTP.

B10

R11

E3100 INTRODUCTION TO THE TEST DEFINITION LANGUAGE

The five ACCTST control programs are written in the Test Definition Language (TDL). TDL consists of BASIC-PLUS and some extensions that appear as comments within the BASIC-PLUS source code. The extensions are primitives that permit the user to define the control structure for parallel execution of several programs and to pass parameters to the programs. Because the TDL is an extension of BASIC-PLUS, the full facilities of the BASIC-PLUS language are available to the user in setting up arguments for the test primitives and in specifying the flow of control through the test.

E3110 TDL TEST PRIMITIVES

The following instructions are written as comments in the BASIC-PLUS program. They are replaced by appropriate function calls by a preprocessor program (TDL or TDL1) before the program is executed.

!LOG <project>,<programmer>,<password>,0%

This instruction causes a pseudo keyboard to be opened and a job logged in under account <project>,<programmer> with a password of <password>. This instructs each program executed to be logged in under the specified account.

!PROGRAM <prog-spec>

This instruction causes a RUN command to be generated (but not executed) with <prog-spec> as its argument. Parallel execution of several programs is effected by giving several !PROGRAM instructions, followed by a single !EXECUTE instruction.

!PAR <string>

This instruction causes <string> to be sent as a line of keyboard input to the program last specified by a !PROGRAM instruction. <string> can be any valid BASIC-PLUS string specification.

!EXECUTE

This instruction calls for the execution of all programs specified by !PROGRAM instructions since the last !EXECUTE. Lines of input specified by !PAR instructions will be sent to each as it requests keyboard input.

EXAMPLE

Suppose we want to run three simultaneous copies of a DECtape exerciser called DTTEST. The program asks what unit it is to use. We want to use units 1 - 3. The test could be specified as follows:

```
100  FOR I = 1 TO 3
110  !PROGRAM "DTTEST"
120  !PAR CHR$(64 + I)
130  NEXT I
140  !EXECUTE
```

If we would like to specify the number of units at test execution time, we could write:

```
 90  INPUT "NR DECTAPE DRIVES"; N
100  FOR I = 1 TO N
110  !PROGRAM "DTTEST"
120  !PAR CHR$ (64 + I)
130  NEXT I
140  !EXECUTE
```

Upon completion of each EXECUTE, the following variables will be set up, and can be tested by the TDL program:

Z%	Job number for which error occurred, or 0% if there was no error. Job 1 corresponds to the first SCRIPT or PROGRAM for this EXECUTE, job 2 for the second, etc.
Z$(x)	Last error message or advisory message received from job x. Error messages are those beginning with a backslash character; advisory messages are those beginning with a percent sign.

!TIMELIMIT <n$>

The TIMELIMIT instruction specifies the maximum run time (in minutes) to be permitted any job step. If the time limit is exceeded, there will be a fatal error message from the job still running, and this will be treated just as other fatal errors. If several jobs are still running, one will get the error and the others will be aborted as the result.

!SCRIPT <script-spec>

This instruction has essentially the same effect as the !PROGRAM instruction, except that instead of a program being executed, an entire script is executed when the next !EXECUTE instruction is reached. In this case strings specified by !PAR instructions are substituted for dummy arguments strings in the script. The string on the first !PAR replaces any occurrence of 1 in the script. The argument of the second !PAR replaces 2, etc. The dummy arguments must be written with no spaces or other intervening characters. The three characters that make up the dummy argument are deleted from the script and replaced by the argument of the corresponding !PAR. Up to nine dummy arguments, 1 through 9 may be included in a script.

Example. Suppose we want to use PIP to cause some disk transfers. We would like to run multiple copies of the job without file name conflicts. We might write the following script:

```
!   PIP.SCP
!   THIS SCRIPT DOES FILE TRANSFERS USING PIP

HELLO
111,1
PASSWORD
RUN$PIP
FIL$1$.TMP KB:
1234567890
1234567890
Z
!DO 10
FIL$1$.TMP<FIL$1$.TMP,FIL$1$.TMP
!GOBACK
FIL$1$.TMP/DF
!END
```

To run three simultaneous copies of this script, using file names FILA.TMP, FILB.TMP, and FILC.TMP we could write the following TDL program:

```
100  !SCRIPT "PIP.SCP"
110  !PAR "A"
120  !SCRIPT "PIP.SCP"
130  !PAR "B"
140  !SCRIPT "PIP.SCP"
150  !PAR "C"
160  !EXECUTE
```

Here we could use the facilities of BASIC-PLUS, if it were more convenient than writing straight line code.

```
100  FOR I = 1 TO 3
110  !SCRIPT "PIP.SCP"
120  !PAR CHR$ (ASCII("A") + I - 1)
130  NEXT I
140  !EXECUTE
```

Perhaps the most important capability provided by TDL is the ability to write a single rather complex test in which the overall flow of control is specified according to parameters given at run time. In the following example we have scripts to exercise magtape drives (MTA.SCP) and DECtape drives (DTA.SCP). The unit number is a dummy argument in each script. We want to run as many copies of each script as we have units of that type. Also, we permit the entire test for each type of device to be repeated a number of times specified by the operator.

```
100  INPUT "NR DECTAPES"; D
110  INPUT "NR MAGTAPES"; M
120  INPUT "NR REPETITIONS"; N
130  IF D = 0 THEN 300
140  PRINT "MOUNT A SCRATCH TAPE WRITE ENABLED"
150  PRINT "ON EACH DRIVE,  TYPE G WHEN READY."
170  INPUT A$;
180  IF A$ <> "G" THEN 140
190  FOR N1 = 1 TO N
200  FOR I = 1 TO D
210  !SCRIPT "DTA.SCP"
220  !PAR CHR$ (64 + I)
230  NEXT I
240  !EXECUTE
245  NEXT N1
250  PRINT "DECTAPE TEST COMPLETE"
300  IF M = 0 THEN 420
310  PRINT "MOUNT A SCRATCH TAPE WRITE ENABLED"
320  PRINT "ON EACH MTA,  TYPE G WHEN READY."
330  INPUT A$
350  IF A$ <> "G" THEN 310
355  FOR N1 = 1 TO N
360  FOR I = 1 TO M
370  !SCRIPT "MTA.SCP"
380  !PAR CHR$ (64 + I)
390  NEXT I
400  !EXECUTE
405  NEXT N1
410  PRINT "MAGTAPE TEST COMPLETE"
420  PRINT "ENTIRE TEST COMPLETE"
```

Any program may signal failure to reach normal termination by typing an error message beginning with a backslash. Such a message will result in all jobs of that step being aborted, and the TDL program continuing. In order to specify an alternate procedure the test writer can include !IFERROR specification.

!IFERROR <lines)

Any time after an !IFERROR has been processed, an abnormal step termination will cause the TDL program to continue at the specified line number. If a system error occurs control will be transferred to line 32000 of the control program. The preprocessor program will generate line 32600 which will transfer control to the line number in the !IFERROR statement after the system error has been processed. When no !IFERROR has been given, the TDL program continues with the next line after the !EXECUTE on which the error occurred.

E3200 TDL PREPROCESSORS

E3210 TDL

TDL is a program designed to preprocess TDL (Test Definition Language) source programs. It creates a file which consists of the TDL functions (TDLFNS.BAS) and the TDL preprocessed source file.

TDL recreates the source program by substituting function calls in place of the TDL statements. These function are !PROGRAM, !LOG, !SCRIPT, !PAR, !EXECUTE, !TIMELIMIT, and !IFERROR.

E3220 TDL1

TDL1 is a program designed to preprocess AC0TDL, AC1TDL, AC2TDL, AC3TDL, AC4TDL, and AC5TDL, and create six new files, designated AC0TST, AC1TST, AC2TST, AC3TST, AC4TST and AC5TST.

Operation is identical to that of TDL.

E3230 TDLFNS

This is a program made almost completely of BASIC-PLUS function calls. The program is prefixed to the "compiled", via TDL OR TDL1, ACCTST programs. It is the linking of the ACCTST program and the actual pseudo batch stream drivers. This program passes the parameters from the TDL function calls to the script file creation and pseudo keyboard drivers.

TDLFNS will perform the necessary housekeeping for the control program and will make the following variables available to the control program:

- L9% This numeric variable contains the maximum number of Pseudo Keyboards configured for this system.
- Q7$ This string variable contains the programmer number of the account under which it is running.
- Q4$ This string variable contains the password of the accouunt under which it is running. It is normally used with Q7$ in the !LOG primitive.

E3300 ACCTST CONTROL PROGRAMS

E3305 ACCTST

This is the first UETP control driver. Its function is to interrogate the user about the peripheral configuration, run and error log output and continuous chaining.

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 to the logs for later use by susquent programs.
3. If option pre-test was selected chain to AC4TDL-AC4TST program.
4. Chain to ACT?TDL-AC0TST for "normal" run.
5. Delete certain cusps if this is a small system so UETP can run.

E3310 AC0TDL-AC0TST

This is the second UETP control driver. Its function is to run the CPU and RP03/RP04 exerciser tests. When this has been completed, "AC1TST" will be called.

The job steps for ACCTST 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's are present, run NEWRP or NEWRR on Drive 0 for two iterations.

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

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

9. Chain to ACITST.

F3320 AC1TDL-AC1TST

This is the second UETP control driver. Its function is to exercise the RX01 disk drives, RK05 disk drives and TU56 DECtape drives, followed by exercising multiple RP03/RP04, RK05, RF11, RS03/04, RX01, and TU56 drives. When this has been completed, "AC2TST" will be called.

The job steps for AC1TST 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 RX01 disks are present, run DXEXEP on drive 0 for two iterations, then run all drives for three iterations.

4. If RK05 disks are present, run DKFXER 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 DKFXER 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 DT (ER on TU56, Drive 0, then run DTFXER on all TU56 drives.

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

9. If present run 8 copies of DMEXEP on RK06 drive 0.

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

11. Chain to AC2TST.

E3330 AC2TDI-AC2TST

This is the third UETP control driver. Its function is to test the mechanical and data reliability of magtape and to execise all devices simultaneously. When this is completed, "AC3TST" will be called.

The job steps 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. If multi-level testing and the account is 1,44 then use all possible magtape and DECtape units until a maximum of eight units has been reached. Then use the disk drives in the following order: DP/DB, RK, RX.

   c. If not multi-level testing the units are selected using the following priority: DP/DB, RK, RX, MT, DT.

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

   e. 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, then print them.

8. Chain to AC3TST.

E3340 AC3TDL-AC3TST

This is the fourth UFTP control program. Its function is to simulate the user environment by executing a series of scripts which simulate users interacting with RSTS/E. Also run is a series of user application packages which will also simulate the user environment. When this is completed, AC0TST will be recalled if chaining was selected in the initial dialogue, otherwise, RSTS/E "READY" mode will be entered.

The job steps 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 PANMAK. 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.
8. If floppy disks are present, execute FLOPPY-MERGE/SORT.
9. If DECtape is present, execute DECMRG-DECTAPE MERGE/SORT.
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 return to RSTS/E "READY" mode.

E3345 AC4TDL-AC4TST

This is the fifth UETP control program. Its function is to exercise specific hardware in the option pre-test environment in volume manufacturing. This program controls sub programs which are primarily electro-mechanical exercises of specific devices e.g., RP04, TU16, RK05, and RX01. This control program was designed to run in a unique environment as follows:

PDP-11/70
64K
KW11L
RK11D-RK05 (unit 0 system disk)
DL11-A (2400 baud)

The job steps for AC4TST 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 the CPFILE program that creates files to be used later.
4. If magtape (TU16) is present execute TAPSRT-MAGTAPE sort.
5. If RX01 disks are present execute FLOPPY-RX01 merge.
6. If RP04 disks are present execute DPRPPR-RP04 mechanical.
7. Delete the created files.
8. Print the error log and if selected, the expanded job statistics.
9. Kill the PERDTA.DTA and STATS.DTA files and return to RSTS/E "READY" mode.

E3346 AC5TDL-AC5TST

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

1. Pick up peripheral data from virtual file PERDTA.DTA.
2. Run the keyboard exerciser test if desired.
3. Set up to log the subjobs under the current user account.
4. If paper tape punch is present execute PPEXER.

5. If paper tape reader is present execute PRFXEP.

6. If a card reader is present execute CRFXEP.

7. Chain to ACOTST if desired.

E3350 SCRIPT - SCRPTB

The SCRIPT program reads one or more script files and compiles them into interpretive code. SCRPTB then executes the code, which sends simulated TTY inputs to the system being exercised, and accepts the outputs to those 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 other jobs.

SCRIPT requires a certain amount of information from its user at run time, and can accept a number of run time parameters at the user's option. The user must initially give SCRIPT a single command. Further instructions, called control statements, are read from a file which has been supplied by the control program's TDLFNS section.

E3400 ACCTST DEVICE EXERCISERS

E3410 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 are also included to verify consistent results of duplicate calculations. Finally, a few miscellaneous tests are performed which have detected hardware failures in the past.

CPEXER test descriptions:

1.	SIN(X)	-Uses SIN(X) extended function.
2.	SIN(X)	-Uses polynominal approximation to SINF 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.

E3420 MTEXFR

The magtape exerciser is used to check normal operation of the TM11 magtape control or RH11/TM02 controller and up to eight seven-track or nine-track TU10 drives or TU16 drives. MTEXFR 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, the tape is zeroed, a file is opened, and data is written until the specified length of tape has been used. The tape is then rewound, the file is open for input, and the data is read and verified. If errors are detected, a count of the number of bytes found to be incorrect is printed before processing to the next iteration.

The data pattern used is a worst case NRZ pattern for nine-track drives. This pattern is not worst case for seven-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 idential 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.

E3430 DTEXFR

The DECtape exerciser is designed to test 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, DTEXEP will print the error count.

E3440 DKEXER

This disk exerciser is designed to test 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 for each pattern. Upon completion of all iterations for a drive, a status report will be printed.

E3441 DMEXER

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

E3442 DFEXFR

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

E3444 DSEXER

This program is similar to DKEXER except that it is designed for file

structured RB03/04 disks.

E3446 CRFXER

This program is designed to test the normal operation of the CD11 or CR11 card reader using a fixed card deck (MAINDEC-89-D1A1-C labeled Alpha Card Deck).

E3447 PPFXER

This program is designed to test the normal operation of the paper tape punch.

E3448 PRFXER

This program is designed to test the normal operation of the paper tape reader by reading the tape punched by DPEXER.

E3450 NEWRP

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

NEWRP 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 1000 records on a temporary disk file. Then starting with the first record in the file, each record is rewritten. These operations are performed until all iterations are completed. The temporary file is then closed and killed.

The actual disk data is never checked as the main object of the exerciser is to cause maximum data transfers and maximum unibus switching.

E3460 NEWRB

This disk exerciser is designed to test the normal operation of the RH11 disk controller and up to eight RP04 disk pack drives. NEWRB tests maximum read/write data transfers on the selected RP04 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 starting with the first record in the file, each record is rewritten. These operations are performed until all iterations are completed. The temporary file is then closed and killed.

The actual disk data is never checked as the main object of the exerciser is to cause maximum data transfers and maximum unibus switching.

E3470 DXFXER

This disk exerciser is designed to test 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 blocks are written and checked. The program also displays the number of words read and written for error rate decisions.

E3500 ACCTST INTERACTIVE TESTS

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

E3510 RANDAC.SCP

Random access to disk records. Build file. Then 1 - N users update random records using record IO. Parameters- Size of file, record size.

E3520 VIRSTR.SCP

Virtual arrays. Program types out strings from a virtual array, according to requests from user. Strings are selected randomly in script.

E3530 EDIT.SCP

Editing and compiling. User types in a simple BASIC PLUS program, runs it, gets error, makes correction, saves it, runs again from BAC file, deletes file.

E3540 CLUMSY.SCP

Clumsy user. User edits BASIC PLUS program with mistakes, cycles through a number of changes -- all wrong. After each edit, execution is attempted, but fails.

E3550 CPU.SCP

CPU exercise. User types in a matrix. Program inverts it and types out result. Several matrices, including singular, and almost singular examples.

E3560 FILES.SCP

File transfers. User copies files back and forth within his disk area. Makes new copies, appends, deletes. Reads back and compares.

E3570 TTY.SCP

Lots of TTY output. User constantly runs program that generates lots of TTY output.

E3580 IMMED.SCP

Simple calculations in direct execution mode. User executes commands to print results of simple calculations such as SQRT, LOG, etc. Exercises all BASIC PLUS functions.

E3600 ACCTST SUPPLIMENTARY PROGRAMS

E3610 CVTSCP

Converts the interactive scripts to run under the current account. The directive "!PASSWORD" is replaced by the current project/programmer number and password.

E3620 DSTATS

Generates an expanded breakdown of the monitor kept disk statistics. Each disk is represented by its file unit number (FUN). A summary of the total statistics is also given.

E3630 JSTATS

Using the virtual file "STATS.DTA" as a base, JSTATS generates an expanded breakdown of the monitor kept job statistics. Statistical catagories are "RUN-TIME", "NO NULL TIME", "FIP", "SYSTEM TICS", "EXEC TIME" and "CACHE HIT".

E3640 ERRDPY, ERRDP1, ERRDP2, ERRDPA

These are modified versions of the standard RSTS/E error display programs with the addition of a disk statistic module (ERRDP2).

The error display program will not display all of the RP04 hardware registers. The tables in sections E3641 and E3642 illustrate the registers which will be displayed and under what conditions. When the errors are displayed the output goes to the selected device e.g., LP or disk. When RK11/RK05, RP11/RP03, RH11/RS03/RS04, RH11/RP04/RP05/RP06, RK06, RH11/TM02/TU16/TU45, Or rx11/rx01 errors are detected this data is also displayed on the operators console.

E3641 RP04 REGISTER DISPLAY FOR PDP-11/40 AND PDP-11/45

ERROR REGISTER	RPER2=0 RPER3=0	RPER2=0 RPER3<>0	RPER2<>0 RPER3=0	RPER2<>0 RPER3<>0
RPDC	YES	YES	YES	YES
RPOF	YES	YES	YES	YES
RPER2	NO	NO	YES	YES
RPER1	NO	YES	NO	YES
RPCS1	YES	YES	YES	YES
RPWC	YES	YES	NO	NO
RPBA	YES	NO	YES	NO
RPDA	YES	YES	YES	YES
RPCS2	YES	YES	YES	YES
RPDS	YES	YES	YES	YES
RPER1	YES	YES	YES	YES

F3642 RP04 REGISTER DISPLAY FOR PDP-11/70

ERROR REGISTER	RPER2=0 RPER3=0	RPER2=0 RPER3<>0	RPER2<>0 RPER3=0	RPER2<>0 RPER3<>0
RPDC				

E3650 RANPAK

Random access file creator program. The random file called "INVFNT.DPY" is created for use by the interactive test script RANDAC.SCP.

E3660 CRFILE

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

E3670 DA,UDA

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

E3680 VERIFY

This prorem checks all the new word files of the TDLSRT sort programs to make sure they are in the proper order, Ten'key files are checked and the check status is returned to the terminal device.

E3700 ACCTST USER SIMULATION PROGRAMS

E3710 DISK SORT - (TDLMQK, TDLQQK, TDLXQK, TDLSQK AND TDLSRT)

The disk sort portion of the user mode tests uses a modified version of the RSTS/E Sort Package. Detailed information on the operation of the Sort Package can be found in the RSTS/E Sort Users Guide.

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 "RELFAS,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" must be executed.

E3720 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, bit by bit, with a known output file "MTSORT.ADT".

E3730 FLOPPY DISK MERGE/SORT - (FLOPPY)

This program will start by checking the device configuration and setting up a file for each unit. This job will merge one file into another and check the results. This jobs objective is to test the unibus and RX01 drive switching capability. The program also displays the number of words read and written for error rate decisions.

E3740 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 has many disk requests for data and many "opens" and "closes" for maximum file handling checkout.

The program transposes five forty by forty matrices back and forth between each other and checks the results after each pass. These matrices are virtual arrays.

E3750 DISK/TAPE MERGE - (DECMRG)

This program will start by checking the device configuration file and using its contents for the job set up. This job will merge one file into another, this is not unlike the shuffling of a card deck. The objective is to check the capability of the unibus logic to switch from device to device. Both devices must grab or relinquish the bus very quickly and cleanly.

E3760 THE COMPUTE BOUND PROGRAM (CPUTST)

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

The three 2 dimensioned arrays are then cleared and/or set to specified values. Matrix "A" is then filled with random numbers from the BASIC-PLUS random number generator call. 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 will be reported. If no error, the program goes back to the matrix clear instructions. When the time expires, the program will print the CPU seconds expended for this job.

E3800 ACCTST SYSTEM ERRORS

E3810 TDLSRT ERROR MESSAGES

Error Message	Meaning
BAD SWITCH	Switch not recognized, or no argument on /K command.
BAD RECORD SIZE	Invalid argument to /K command.
BAD FILE SPEC	SRC or DST invalid or missing.
BAD KEY SPEC - RETYPE IT	Invalid key specification; try that specification over.
BAD CLUSTER SIZE	Argument to /R command not valid.
KEY RECORD TOO BIG	Too many large key specifications. (The size of the augmented key record is greater than 508 bytes).
OPEN ERROR - DAT	Open error on data file. Either a bad file spec, or file protected, or hardware problem.
OPEN ERROR - KEY	Open error on key file, same as for the data file.
ALLOCATION ERROR	Could not preallocate the file. Probably not enough disk room. Possibly a bad cluster size of /Fin. File is not deleted if a "partial initialize" was being performed.
OPEN ERROR - WORD FILE	TDLSQK (TDLMQK) could not open the file SORTmn.TM1 or SORTmn.TM2. The value of the RSTS/E ERR variable is printed.
I/O ERROR - DAT	I/O error in TDLXQK or TDLSRT on the data file. Probably a hardware problem.
I/O ERROR - KEY	I/O error in TDLXQK or TDLSRT on the key file. Probably a hardware problem.

TOO MANY RECORDS - DAT	Data file header (type 1) contains an invalid record count, or record size.
KEY LIMIT EXCEEDED	Too many key fields were specified. The question "PROCEED?" allows the user to extract the first 15 fields specified if answered with "YES".
WORK FILE - SORTmn.TM1 NOT FOUND	The work file was not found to exist on a /M command. The "mn" must be the current job number, not necessarily the job number when the merge was suspended.
I/O ERROR - COMMAND FILE	EOF before line feed, or a system problem in processing the specified indirect command file.
NON-EXISTENT COMMAND FILE	Specified file does not exist.
INVALID RECORD COUNT	Invalid response to the query # RECORDS.
SORTING ERROR - n	The decimal integer n returned by the TDLSRT program gives the value of the BASIC-PLUS variable ERR. (The ERR variable is described in Sections 8.4 and C.1 of the BASIC-PLUS Language Manual and in Section C.1 of the RSTS/E User's Guide.)

E4000 RSTS/E SUPPLIMENTARY DOCUMENTATION

E4100 DIFFERENCES BETWEEN RSTS/E V06B-02 and V06A-02

RSTS/E V06B-02 introduces new hardware support and several new software features. Improvements were made in the internal structure of the monitor, overlay, initialization and BASIC-PLUS programs. The system generation procedures were re-implemented to provide a RSTS/E based generation process and to eliminate the DOS dependency. Support for BASIC-PLUS II and RMS-11 have been included as well as facilities for supporting DECNET/E when it becomes available.

All known V06A-02 problems have been corrected. The manual set has been revised to reflect the new hardware changes and software corrections. Several new additions were made to the manual set as well as a complete revision of several of the documents.

NEW PROCESSOR SUPPORT

No new processors are supported under V06B-02. Support for up to 2 million words of memory for the 11/70 has been included. The 11/34 FPP compatible floating-point option is now supported in V06B-02.

NEW DEVICE SUPPORT

Several devices introduced during the period after V06A-02 was released have now been included in the supported device category for V06B-02.

DZ11 ASYNCHRONOUS MULTIPLEXOR

The DZ11 Asynchronous Multiplexor is a PDP-11 peripheral device that provides as many as eight (8) communications channels between a variety of devices and the PDP-11 UNIBUS. The DZ11 operates full-duplex on asynchronous serial data inputs from a modem or terminal, converting the data stream to the parallel format of the UNIBUS. Conversely, parallel data from the UNIBUS is serialized by the DZ11 for transmission to the modem or terminal. Both EIA and 20 ma. current-loop interfacing is available as well as control capability for BELL 103A or equivalent data set modems.

Several DZ11 features provide flexible control of communications parameters such as baud rate, character length, number of stop bits for each line, odd or even parity for each line, and

transmitter/receiver interrupts. Additional features include limited data set control, zero receiver baud rate, break generation and detection, silo buffering of received data and line turn-around.

Each DZ11 provides operation for eight (8) channels. Up to 16 DZ11 multiplexors may be included in a RSTS/E installation.

The DZ11 multiplexor is supported under RSTS/E V06B-02 through the terminal service part of the operating system. All of the terminal characteristics are available, with the exception of SPLIT SPEED. The latter is not available because the DZ11 only permits a single speed selection for both transmit and receive on a given line.

2741 terminal support is also available through the DZ11 support.

DJ11 ASYNCHRONOUS MULTIPLEXOR

RSTS/E now supports the DJ11 serial asynchronous multiplexor interface system as part of the terminal support capability. Since the DJ11 channel characteristics are not programmable in the multiplexor, the terminal characteristics controllable through the terminal characteristics utility TTYSET are the same as for a single channel asynchronous interface such as the DL11. No modem control is provided in the DJ11 hardware.

RK05F DISK DRIVE

RSTS/E V06B-02 now supports the RK05F disk drive as a member of the overall RK05 disk structure.

Since RSTS/E will be distributed only on the RK05J type disk cartridge, a standard RK05 disk drive must appear as drive unit 0 for those RSTS/E systems which will receive RSTS/E on RK05 distribution media.

RK06 HIGH PERFORMANCE DISK CARTRIDGE DRIVE

The RK06 disk drive is DIGITAL's new medium capacity disk drive designed for the small to mid-range PDP-11 configurations. It features a storage capacity of 27104 blocks(512 bytes/block) in a compact cartridge design and uses technology similar to the RP04/RP05/RP06 disk drive family to provide reliability and high performance.

The RK06 interfaces to the PDP-11 through the RK611 controller. Up to eight (8) drives may appear on a single RSTS/E system.

The RK06 disk drive is identified in RSTS/E using the DMn: disk device name. The RK06 may be used as the system disk, as part of the

public structure or as a supporting part of the private disk capability. RSTS/E V06B-02 will also be available for distribution on RK06 media. For systems with no magnetic tape, and no RK05 disk drives, at least two (2) RK06 drives must be configured.

RP05 AND RP06 HIGH-CAPACITY HIGH-PERFORMANCE DISK DRIVES

The RP05 and RP06 disk drives offer the same high level of performance and broad range of features currently provided in the RP04 disk drive. The two new drives use the same PDP-11 device controllers as the RP04 and all three drive types can be intermixed on a single PDP-11 configuration.

The RP05 capacity is the same as that of the RP04 disk drive. The RP05, however, is upgradable in the field to double its capacity to make it an RP06. Thus, RSTS/E users can now have over 320K blocks of storage/drive in their on-line disk sub-systems using the RP06 capability.

RSTS/E V06B-02 will support the RP04/RP05/RP06 drives as part of the same basic structure. All three devices will be named using the DBn: disk device name. RSTS/E will not be distributed on any of these media.

LA180 DECPRINTER I

The LA180 DECprinter I low-cost line printer is now supported by RSTS/E V06B-02. When connected into the PDP-11 UNIBUS as a parallel interfaced device, the LA180 is made known to the RSTS/E monitor as an LP11 printer, with the desired assortment of options. In this regard, the printer is handled as a standard LP11 printer would be handled.

When connected to RSTS/E as an LP11 equivalent printer, the hardware forms length capability of the LA180 is supported only when used under the control of the RSTS/E Spooling package. For details of this feature and its control, consult the Release Note Articles on the Spooling package, the appropriate section of the RSTS/E System Manager's Guide, and the LA180 DECprinter I User's Manual.

The serial interfaced version of the LA180 (henceforth called the LA180S) is conditionally supported under V06B-02 as a Receive-only printing terminal through the Spooling package. For details of this support, consult the Release Note Articles on the Spooling package, the appropriate section of the RSTS/E System Manager's Guide, the LA180 DECprinter I User's Manual, and the LA180S Buffered Serial Interface User's Manual.

MAGNETIC TAPE SUPPORT

RSTS/E V06B-02 now supports the following 7 and 9-track tape drives:

TS03	9-track only
TU10	7 and 9-track
TU16	9-track only
TU45	9-track only

On a given RSTS/E system, all TS03 and TU10 tape drives are treated as part of the same device class, while all TU16 and TU45 tape drives are grouped together. Up to eight (8) units of each class may be installed on the same RSTS/E installation.

SYSTEM GENERATION

System generation is now done under a pre-built RSTS/E V06B-02 monitor instead of DOS/BATCH. All system generation media are RSTS/E file structured. While similar in flow to V06A-02 system generation procedure, the V06B-02 System Generation Manual (DEC-11-ORGNA-B-D) should be read in full prior to attempting a sysgen.

INITIALIZATION CODE

The system initialization code has been completely re-organized. The System Generation Manual (DEC-11-ORGNA-B-D) and System Manager's Guide (DEC-11-ORSMD-B-D) should be carefully read to become familiar with the new options and procedures.

The new INIT does a complete system device check every time it is loaded (booted). Any non-existant device is automatically disabled. In addition, all devices are forced to interrupt and their vector locations and priority levels are determined. Devices which won't interrupt, interrupt at too high a priority level, or interrupt to the same vector as another device are also automatically disabled.

MONITOR AND FILE PROCESSING

The internal system tables and data structures were completely revised and re-organized to improve the internal memory utilization in the monitor and to implement various standards throughout the monitor. This permitted the removal of a large number of inconsistencies between various structures which have developed as RSTS/E evolved.

A standard form for the Device Data Block (DDB) has been developed and all device drivers in the monitor now use this format. V06A-02 programs which PEEKed at locations in DDB's may now have to be modified to account for the new structure.

All internal system table structures have been revised. For control of jobs, the Job Data Block format has changed.

The memory control structure was changed to allow for up to 2 million words of on-line memory for 11/70 processor systems. The structure was also modified as part of upgrading Run-Time System support in V06B-02.

Miscellaneous areas of further improvement were the areas of Run-Time Systems handling and support, general device control, standardized device naming structures and device handler support.

A central device time-out facility was implemented in the monitor and all system device drivers use that facility in controlling their devices.

In the disk control section of the monitor, an improvement was made which allows both device type and unit number to be selected at bootstrap time (V06A-02 allowed only the unit number to be changed).

Improvements in the control of the monitor memory mapping registers were implemented which allow several monitor segments (PIP, EMT services, I/O driver routines, 2780 handler, and future network controllers) to be mapped outside of the lower 20K words of physical memory. This allows the amount of monitor required to be permanently mapped into the first 20K words of memory to be minimized and allows greater flexibility in assigning the proper number of small buffers to optimize system performance.

A new feature, called "system logical names" has been implemented. This feature allows the specification of up to 30 system-wide logical assignments of device and/or account in terms of six (6) character identifiers, such as LIB: to mean DK1:[1,100]. V06A-02 provided only one system-wide logical name, that is SY0: for the system disk. These names are specified during startup, through the utility program UTILITY and exist for all users.

The send/receive message facility of V06A-02 was completely re-written to provide facility for transmitting messages within a system, or in the future, through DECNET. A new set of system function SYS calls has been provided to interface to the new facility, with the old form SYS calls retained in the interest of compatibility. SEND/RECEIVE messages may now contain up to 532 bytes of information. Provision has been provided for selective receive of messages based on sending job # as well as providing for selective packing of message buffers during reception.

The Concise Command Language (CCL) facility of the BASIC-PLUS Run-Time system was moved into the RSTS/E monitor. Commands have been added to allow the on-line addition and deletion of CCL commands from the active list as well as allow use of the small buffer memory for storage of the CCL command list. Each CCL has a unique abbreviation specified by the command installer. Additionally, the specified CCL commands are available to all other Run-Time Systems, such as FORTRAN IV.

For file processing, four (4) additional file OPEN modes have been added. The first, known as "read-only", allows a file to be OPENed for read privileges only, even if the file is already OPEN elsewhere with both read and write privileges.

The second OPEN mode, known as "read-regardless", now allows OPENing of files, even if they are already OPEN in UPDATE mode, with reading allowed of any record in the file, even those which might be locked.

The third OPEN mode allows for the creation of contiguous files. Contiguous files do not require "window turns" for random access processing and result in greater system performance in data base applications.

The fourth OPEN mode keeps a sequential output file's length information always updated in the file's directory. This ensures that all of the file's data can be recovered after a system crash/power fail.

A new device, called the NULL device, is available to all users. This device returns an immediate end-of-file error on reads (GETS) and serves as a "bit bucket" on writes (PUTS).

Update MODE has been enhanced to allow the locking of multiple blocks (up to 15). In addition, multiple locks can be achieved using the RMS-11 Package.

A central file name string scan facility has been added to the monitor. This central facility will be used by all Run-Time Systems to provide for a uniform method of scanning and parsing file specification strings for RSTS/E files. As networking is introduced, the importance of a central facility such as this increases.

The support of Run-Time Systems now includes an automatic loading capability which is invoked when the user "runs" the RTS. This facility allows automatic selection of the location of the RTS in memory and prevents conflicts between Run-Time Systems competing for memory space. An improvement to the handling of privilege by privileged programs has been provided whereby a program may request the system to "drop" its privileges, temporarily, such that the program can re-gain them at a later time. This allows a privileged program to use the file protection capability of RSTS/E at file OPEN time, after which the program re-gains privileges if it needs them.

The swapping file structure was modified to allow for the addition/removal of system swap files during time-sharing operations. The placement of swap files may now be made on any RSTS/E disk system with a file structure, or on other devices in non-file structure mode.

Lastly, the error logging sections of the monitor and all device drivers were overhauled. The amount of information logged for each error was increased, and a new set of error control utility programs provided.

TERMINAL SERVICE

The Terminal Service sections of the RSTS/E V06B-02 monitor were revised in order to integrate with the general overall monitor re-implementation, and to allow for the handling of the DZ11 and DJ11 multiplexor systems.

Additionally, a new operation mode, called the "echo control" mode was implemented to facilitate development and usage of "screen form" types of input procedures.

Each keyboard in the system may now have one extra, user-specified delimiter character assigned to it, performing the same input text termination functions as <cr>, <lf>, <esc> and <ff>. For example, the character <tab> could be used in conjunction with the "echo control" mode feature to provide an improved data entry utility.

BASIC-PLUS LANGUAGE AND RUN-TIME SYSTEM

Several changes were made to the BASIC-PLUS language and run-time system to improve old functionality and/or add new features.

The LENGTH command now reports both the current private memory size along with the maximum size. For example:

```
LENGTH
5(8K) of memory used
```

will appear for a program using 5K of memory, with a maximum allowed size of 8K.

To avoid having to copy data from one I/O buffer into another when copying files, an "alternate I/O buffer" capability has been added to GET/PUT operations in BASIC-PLUS. The general form of the GET statement which uses an alternate I/O buffer is as follows:

```
GET #SWAP%(B%)+I% [,RECORD R%]
```

The input file is still OPEN on channel I%. If B% = 0%, then the GET reverts to the standard operation using the buffer associated with channel I%. If B% <> 0%, then the resulting data fetched with the GET will be placed in the buffer associated with channel B%, where 1% <= B% <=12%. In either case, the GET statement will never output a "?" prompt if the input file is a keyboard.

The form of the PUT statement using an alternate I/O buffer is as follows:

```
PUT #SWAP%(R%)+I% [,RECORD P%] [,COUNT C%]
```

with the GET statement, R% = 0% reverts to the standard GET operation. If R% <> 0%, then the data used in the PUT operation will be found in the buffer associated with channel R%.

For both the GET and PUT statements, channel R% must be OPEN. The new RSTS/E null device can be very useful in this case for providing for the OPEN channel without requiring an actual file to be OPEN on disk.

To illustrate the new GET/PUT statements, we show a sample copy loop program for copying two files whose record sizes are the same.

```
10        OPEN "FOO.OLD" FOR INPUT AS FILE 1%
        \ OPEN "FOO.NEW" FOR OUTPUT AS FILE 2%
                ! OPEN BOTH OLD AND NEW FILES

20        GET #1%
        \ PUT #SWAP%(1%)+2%
        \ GO TO 20
                ! GET RECORD FROM INPUT,
                ! OUTPUT RECORD, USING INPUT CHANNEL BUFFER
                ! CLOSED LOOP.  LOOP IS EXITED VIA END-OF-FILE
                ! ERROR (ERR=11)
```

The rules concerning the specification and processing of the RECORDSIZE modifier for the file OPEN statement have been modified for V06B-02. Assuming a RECORDSIZE value of S%, and a RSTS/E default value for RECORDSIZE of D% (consult the RSTS/E System User's Guide for default RECORDSIZE values for peripheral devices handled by RSTS/E), the rules for processing the RECORDSIZE are as follows:

A) Force S% to be an even quantity(i.e. S% = S% AND -2%)
B) The RSTS/E monitor assumes a default value for D% according to the associated device type of the file being OPENed.
C) If S% = 0%, then use the RSTS/E default value of D%.
D) If S% > 0% (positive value), then:
   D1) Use the user supplied value of S% if S% > D%
   D2) Use the RSTS/E default value D% if S% <= D%.

F) If S% < 0% (negative value), first remove the sign bit; i.e. S% = S% AND 32767%. Then:
   E1) If the resulting S% = 0%, then use the RSTS/E default value of D%.
   E2) If the resulting S% <> 0%, then use the resulting value of S%.

As can be seen, all of the rules, except E2, are the same in V06B-02 as they were in V06A-02. Rule E2 is the new variation provided in V06B-02.

As an example of the utility of the new handling of RECORDSIZE, statement 10 of the previous GET/PUT example is re-written below:

```
10        OPEN "FOO.OLD" FOR INPUT  AS FILE 1%
        \ OPEN "FOO.NEW" FOR OUTPUT AS FILE 2%,
              RECORDSIZE (32767%+1%+2%)
                      ! OPEN BOTH OLD AND NEW FILES
```

By using RECORDSIZE (32767%+1%+2%) on the output file OPEN, the associated buffer on channel 2% is established at a length of two (2) bytes. This means that 510 bytes of data space would be saved while accomplishing the same copy operation.

A modified version of the NUM$ function, called NUM1$, has been added to the BASIC-PLUS language. The NUM1$ function performs just like NUM$ except that:

No spaces are returned
The E format is never returned

The NUM$ function is still available.

An optional package for doing arithmetic operations on numeric data which is stored in character strings has been made available. The package may be included in a RSTS/E installation by answering the appropriate question during the system generation process.

The string arithmetic package provides the following functions:

SUM$	A$ = SUM$(B$,C$)	(Addition)
DIF$	A$ = DIF$(B$,C$)	(Subtraction)
PROD$	A$ = PROD$(B$,C$,P%)	(Multiplication)
QUO$	A$ = QUO$(B$,C$,P%)	(Division)
PLACE$	A$ = PLACE$(B$,P%)	(Truncation)
COMP%	T% = COMP%(A$,B$)	(Comparison)

where A$, B$, and C$ are string of numeric digits, and P% is a precision length value. See the BASIC-PLUS Language Manual for further details on how these functions are used.

Error messages returned by the BASIC-PLUS run-time system have been improved as follows:

- If necessary, a <cr>-<lf> is output to restore the left-hand margin of the terminal
- Error messages begin with either "?" or "%" prefixed to the message. If the message is informational only, neither prefix character will appear.

These "?" (FATAL) and "%" (WARNING) designations are utilized in the Spooling package BATCH facility for conditional control of batch job streams.

A new operating mode has been added to the BASIC-PLUS Language and Run-Time System which allows the following extensions of the language:

- Variable names of up to 30 characters are allowed. These names consist of a leading alpha character and up to 29 additional characters(alphanumeric and "."). In addition, a name may have an optional "FN" prefix(for functions) and/or either of the "%" or the "$" suffix.
- Because of the long variables name feature, spaces and tabs in the language and command syntax of BASIC-PLUS are significant. For example, FOR I = 5 TO P is now required, as is RUN $PIP.

- A line ending in

  &[<space/tab> sequence]<cr>

  will be treated as a continuation of the previous line (similar to the V06A-02 <lf> convention).

- If a line ends with

  !<comment not including "!">&[<space/tab>]<cr>

  the comment will be treated as an in-line comment with continuation of the program text on the next line.

This new operation mode is called the "extend" mode of BASIC-PLUS. Language and system commands allow for switching between the old compatibility mode(NO EXTEND) and the new mode(EXTEND).

The EXTEND/NO EXTEND commands can be used in immediate mode as well as appear in BASIC-PLUS programs. If a mode change command appears in a program, its effect is local to the program. If the command is typed in immediate mode, the effect is permanent until the next immediate mode change.

In either of the two operating modes of BASIC-PLUS, there is now no restriction on the total number of characters which can occur in a given BASIC-PLUS language statement. This means that the BASIC-PLUS programmer now is freed of any restrictions on how to format his program for readability; only restrictions in the BASIC-PLUS language remain.

The STATUS variable provided in BASIC-PLUS has been changed with regard to results returned after performing a "file name string scan"(See the System Programmer's Manual on system (SYS) functions). The old meaning for bit 12(value 4096%) has been changed to mean "device accepts modifiers". The following table shows the meanings of the STATUS result depending on the returned values of bit 12, bit 13 and bit 15:

Bit 15	Bit 13	Bit 12	
100000	040000	010000	
-32768	16384	4096	Meaning
0	0	0	Block sequential, no modifiers
0	0	1	Block sequential, modifiers (Note 1)
0	1	0	Byte sequential, no modifiers
0	1	1	Byte sequential, modifiers (Note 2)
1	0	0	Block random, no modifiers
1	0	1	Block random, modifiers (Note 1)
1	1	0	** Impossible **
1	1	1	** Impossible **

Note 1: This case can not be handled by BASIC-PLUS.
Note 2: This case is handled by using the RECORD modifier value on GETs, PUTs, and/or PRINTs to the device in question.

Several changes have been made to the non-privileged system function calls (SYS).

The system function calls SYS(CHR$(x%)), where x = 0, 1, 2, 3, 4, and 11, have been changed to the following general format:

```
SYS(CHR$(x%)[+CHR$(C%)[+CHR$(K%)]])
```

where: a) No C% or K% means use job's keyboard(KB:)
b) C% and no K% means use the terminal on channel C%
c) C% and K% both means use terminal keyboard _# K% which is a multiple service terminal on channel C%

SYS call 5% has been further modified to cause the following actions:

a) Do not "wipe out" program
b) Do not return to private run-time system
c) Set up to allow a CONTinue command to be typed
d) Do not issue a prompt message ("Ready")

SYS call 9% has been modified to cause the following actions:

a) Close all channels without cleaning up partial buffers
b) Always "wipe out" program
c) Always return to private run-time system
d) Tell the run-time system to issue a prompt

In addition, SYS call 9% has one additional format:

```
SYS(CHR$(9%)+CHR$(N0%)+CHR$(N1%)+CHR$(N2%)+CHR$(N3%))
```

The actions invoked by this form of SYS call 9% are:

a) Close all channels without cleaning up partial buffers
b) Always "wipe out" program
c) Return to the run-time system whose RTS name is encoded in values N0% through N3% in RAD50 format
d) Furthermore, establish the indicated run-time system

as the user's private default run-time system
e) Tell the run-time system to issue a prompt

The user should read the V06B-02 manuals for more information on the use and control of run-time systems.

As part of the new "echo control" mode features of the RSTS/E monitor terminal handler, a new SYS call 11% has been provided to cause the following action:

Cancel all currently pending input("type ahead")

This SYS call permits clearing of terminal input buffers prior to declaring a field on the terminal; thus, any characters which had been typed, but not echoed, will not overwrite any previously written characters on the terminal display.

The privileged SYS call -21% has been modified as follows. The form

SYS(CHR$(6%)+CHR$(-21%))

causes a permanent dropping of privileges for the program in which the SYS call is executed.

The form

SYS(CHR$(6%)+CHR$(-21%)+CHR$(255%))

causes the program to lose its privileges, but does so in a way that they can be re-gained at a later time by the SYS call

SYS(CHR$(6%)+CHR$(-21%)+CHR$(0%))

RSTS/F UTILITY LIBRARY PROGRAMS

Rather than describe the entire set of Commonly Used System Programs (CUSPs) as a single library, the programs are grouped as a set of packages which are included in each RSTS/E System Generation Distribution kit. The packages are listed as follows:

RSTS/F System Utilities Package
RSTS/F Data Manipulation Package
RSTS/E Error Control Package
RSTS/E Backup Package
RSTS/E Spooling and Operator Services Package
RSTS/E Run-Time System Support Package
RSTS/F User Environment Test Package

The changes in V06B-02 versions of programs in each of these packages is described below, along with abstract descriptions of new programs being introduced with the V06B-02 release of RSTS/E.

Whether indicated or not, some changes occurred in nearly all of the programs, which were not rewritten entirely, with regard to implementing aspects of the BASIC-PLUS Coding Standard. Additionally, a modification history log capability was inserted in every program and any patches made during maintenance procedures will appropriately update the edit level and modification history of the programs.

RSTS/F SYSTEM UTILITIES PACKAGE

The RSTS/F System Utilities Package part of the RSTS/E Utility Library contains the following:

BUILD
DIRECT
DISPLY(VT05, VT50, VT52)
DSKINT
GRIPE
INIT
INUSE
LOGIN
LOGOUT
MONFY
ODT
PIPASE
PRIOR
QUOLST
REACT
REORDR
SHUTUP
SYSCAT

SYSTAT
TALK
TTYSET
UMOUNT
UTILTY
UTILT1

BUILD

The BUILD program now allows libraries to be built onto some specified disk instead of the system disk, and/or into a library account other than [1,2].

DIRECT

The DIRECT disk directory listing program has been extended to display more information about each file in a /S (slow) directory listing. Specifically, the name of the Run-Time system by which the file was created and any attribute information are shown.

DISPLY(VT05, VT50, VT52)

DISPLY is the base program used in conjunction with VT05.DPY and VT50.DPY to form programs for dynamic display of system status information on VT05, VT50 and VT52 terminals. The modules VT05.DPY and VT50.DPY are appended to DISPLY to form the programs VT5DPY and VT50PY which output to VT05 and VT50(VT52) terminals, respectively.

The system status display conforms to that of the SYSTAT utility program, within limitations on available screen space for display.

New commands added to the DISPLY base program allow sleeping jobs to be included/excluded from the job list, and provide for display of each job's Run-Time system to be displayed instead of the job's name.

The VT05 control module functions are unchanged from Version V06A-02. The program coding was modified to conform more with the internal DIGITAL BASIC-PLUS coding standard.

The VT50 control module was modified to provide several program startup switches which permit the user to specify use of VT50 and VT52 features, such as 24 lines/screen, direct cursor addressing, hardware tab functions, and echo control.

The module program coding was modified to more closely conform with the internal DIGITAL BASIC-PLUS coding standard.

DSKINT

The DSKINT program was re-written to permit it to initialize all disk types supported by RSTS/E. Additionally, any RSTS/E disk can now be

re-initialized by DSKINT. In V06A-02, no re-initialization was possible, nor were any disk types other than RK05 and RP02 supported.

The DSKINT program can not format a disk, nor can it perform the pattern checks to locate bad blocks. If the disk being initialized is formatted, and contains a valid RSTS/E structure, DSKINT will retain bad block data if it is present. If the disk is formatted, but not necessarily in the RSTS/E structure, the bad block data will not exist.

During initialization, the user may specify the placement of the storage allocation table file (SATT.SYS), for example, at the mid point of the disk being initialized.

GRIPE

The GRIPE program functions in V06B-02 as it did in V06A-02.

INIT

The INIT program now has a built-in time-out interval of 30 seconds when the command file name is requested. If no reply has been given in that time, INIT automatically uses START.CTL.

When re-attaching to a terminal, INIT now uses the REATTACH SYS call instead of going through a LOGIN sequence.

Commands have been added to INIT to ADD SWAP files as well as OVERLAY and ERROR system files. (See System Manager's Guide.)

INUSE

The INUSE program functions in V06B-02 as it did in V06A-02.

LOGIN

The default LOGIN SWAP MAX value has been changed. In V06A-02, the default value was 8K. In V06B-02, this value has been increased to 28K.

Individual installations should re-evaluate their requirements with regard to default setup of jobs at LOGIN time.

LOGOUT

LOGOUT now deletes all files that match ????nn.TMP (where nn is the job number); it previously deleted only TMPnn.TMP files.

MONEY

The VOLFY program functions in V06B-02 as it did in V06A-02.

ODT

The ODT program functions in V06B-02 as it did in V06A-02.

PLEASE

PLEASE has been re-written in V06B-02 as part of the Spooling and Operator Services Package. If the Operator Service program OPSER is not running, messages input to PLEASE will be sent to the system console terminal (KB0:) as occurred in V06A-02. When OPSER is present, PLEASE may be used to send either messages or commands to OPSER. Messages are printed by OPSER on some system designated terminal, known as the Operator Services Console (OSC), while commands are passed to OPSER for interpretation and execution.

PRIOR

The PRIOR program functions in V06B-02 as it did in V06A-02.

QUOLST

The QUOLST program functions in V06B-02 as it did in V06A-02.

REACT

The REACT program functions in V06B-02 as it did in V06A-02.

REQRDR

The REQRDR program was re-written to conform to the coding standards specified for RSTS/E CUSPs. Additionally, the program now permits the sorting of the files in a User File Directory (UFD) according to either creation date/time or date of last access, with sorting in either ascending or descending order on the parameter(s) chosen.

SHUTUP

The SHUTUP program was re-written completely in order to provide for a single program which would be used to shutdown the RSTS/E system under all operating conditions. The program functions appropriately whether the Spooling Package is in use or not.

Warning messages, detailing the remaining minutes until the shutdown procedure begins, now appear at intervals whose lengths decrease as the shutdown time approaches, resulting in fewer messages being broadcast. As SHUTUP runs, it performs its own system consistency

checks and attempts to shutdown in the fastest possible way.

SHUTUP will operate in cooperation with the Operator Services Program OPSER to close down spooling operations if desired by the operator. SHUTUP and OPSER provide for either an immediate shutdown or a controlled shutdown, where the spoolers finish the job they are currently working on.

SHUTUP now takes care of shutting down all Run-Time Systems present on the system, as well as removing SWAP files from usage and ending the error logging process. When shutdown is finished, the RSTS/E system initialization code INIT is re-booted so that the system manager may proceed as desired.

SYSCAT

The SYSCAT program functions in V06B-02 as it did in V06A-02.

SYSTAT

Several features have been added to SYSTAT in the V06B-02 release. New options available to the user are as follows:

Option	Purpose
/A	list attached jobs
/M	list message receiver status
/N	list jobs in non-privileged accounts
/P	list jobs in privileged accounts
/U	list detached jobs(replaces the /DET option)
/n,*	list all jobs in project # <n>

Any combination of the basic options (A, B, D, F, M, N, P, R, S and U) is permissible. A minus "-" prefix may be included with any of the basic job status options to force the printing of an actual account number instead of the standard output which might include "[SELF]" and/or "[OPR]".

The information listed in most of the existing sections of the SYSTAT output has been expanded. All of the new features have also been added to the ANALYS (Crash analysis program) and, where necessary, to the DISPLY base program. The extensions are:

Section	Additions
Job Status	Run-Time system under which a job is running, additional state information, and, where applicable, pseudo-keyboard and controlling job number information

Disk Structure	Disk logical name or pack identification, expanded comments on state of disk
Buffer Status	Count of system errors, job count information
Run-Time Systems	Default extension, extensive new comments

TALK

The TALK program functions in V06B-02 as it did in V06A-02.

TTYSET

The DH BURST command has been removed as it is no longer necessary.

A new command, DELIMITER, has been added to allow specification of a private delimiter for a given terminal.

The PRINT command allows setting up of start-up escape sequences for use with more intelligent terminals, such as the VT52 display.

The NO ESC and ESC commands now determine translation characteristic for special characters. All these features are explained in the RSTS/F System User's Guide in the section on the TTYSET program.

UMOUNT

The UMOUNT program is accessed through the MOUNT and DISMOUNT CCL commands. The following new switches to the commands have been added in V06B-02.

Switch	Description
/LOG[ICAL]:<text>	Use <text> as the logical identifier by which to refer to the pack, rather than using the PACK ID.
/NOL[OGICAL]	Do not put any logical identification into the system for this pack.
/RO[NLY]	Mount the disk specified as read-only disk. Must be used if the drive is physically set up with WRITE disabled.

Any non-privileged user attempting to DISMOUNT a disk is now required to specify the PACK ID for the disk in the DISMOUNT CCL command. If a mis-match occurs, the pack will not be logically dismounted. In order to prevent a disk from being DISMOUNTed by a non-privileged user, the system manager should mount the disk initially using the /LOGICAL specification of a different logical identification. Since the

logical identification is the "name" shown on SYSTAT printouts, the non-privileged users will be unable to find out what the true PACK ID of the disk is, thus preventing them from DISMOUNTing the disk.

UTILTY
UTILT1

In order to provide for access the major new features of the RSTS/F monitor, the UTILTY program's command set has been greatly extended. New features include commands to ADD and REMOVE system files(swapping files, overlay files, error message file); manipulate system-wide logical device names; manipulate run-time systems, enable and disable disk caching; add, remove and list CCL commands; and enhance the system manager's control over jobs in the system.

In order to provide all of the UTILTY features while running in an 8K word user space, the UTILTY program was divided into two programs, UTILTY and UTILT1.

RSTS/F DATA MANIPULATION PACKAGE

The RSTS/F Data Manipulation Package part of the RSTS/F Utilitiy Library contains the following:

COPY
EDIT
EDITCH
FILCOM
FLINT
PIP
PIPEXT
RUNOFF

COPY

The COPY program has been modified to allow the use of the "=", as well as the "<" to separate the input and output sides of commands.

EDIT
EDITCH

The EDIT and EDITCH programs function in V06B-02 as they did in V06A-02 except for the "hidden" option PATCH to be used for CUSP maintenance purposes.

FILCOM

The FILCOM program was modified to produce a "?" (FATAL) error prefix to the "xxx DIFFERENCES FOUND" message if <xxx> is non-zero. This

permits FILCOM to be used properly in a BATCH stream, obeying the new FATAL and WARNING error level conventions.

FLINT

The FLINT (FLoppy INTerchange) program is a new program appearing in the V06B-02 release. The program provides the user with the means to interchange IBM floppy disk data sets to and from RSTS/E data files.

FLINT allows a RSTS/E user to translate a single RSTS/E file (ASCII) to an IBM (EBCDIC) floppy disk or perform an Image mode (byte-to-byte) copy. Multi-volume output is allowed on RSTS/E to IBM transferrals. FLINT also permits single or multi-volume translation from an IBM data set to a RSTS/E file. This feature is necessitated by the fact that IBM permits multi-volume (maximum of 99) data sets.

PIP

The PIP program (8K word version) functions in V06B-02 as it did in V06A-02, except that it now accepts an = as well as a < to separate the input and output sides of a command.

PIPFXT

PIPFXT was modified to make use of the alternate I/O buffer feature now available under BASIC-PLUS. The feature is used in those cases where the input and output transfer type is the same (/RI and /CO only) and the input and output buffer lengths are equal.

RUNOFF

The RUNOFF program functions in V06B-02 as it did in V06A-02.

2.8.3 RSTS/E Error Control Package

The RSTS/E Error Control Package part of the RSTS/E Utility Library contains the following programs:

```
ANALYS
ANALY1
ERRINT
ERRCPY
ERRDIS
ERRDET
```

ANALYS
ANALY1

The crash dump analyzing portion of the Error Control Package is comprised of two programs in the V06B-02 release. The ANALYS program obtains all user input specifications and outputs the first part of a SYSTAT-compatible crash status report. ANALY1 (chained to by ANALYS) produces the remainder of this status report, an expanded/reformatted "Octal Dump of Status" and, optionally, a "Core Dump of the Monitor". ANALY1 also retrieves error logging information saved at the time of a crash (thereby replacing the V06A-02 ERRCRS program) and stores it in a specially formatted file, by default $ERRCRS.FIL.

The new crash analysis programs are designed to operate in one of two modes:

1. Default mode - the file [0,1]INIT.SYS is examined to ascertain which .SIL (Save Image Library) file from which to extract pertinent monitor table information. This data is used when examining the file [0,1]CRASH.SYS (the crash dump file) for information needed to produce crash status reports and to create the $ERRCRS.FIL error logging file.

2. Non-default mode - this mode has been added to aid in the trackdown of system problems reported in SPRs. The ANALYS program will allow the user to specify the crash dump .SIL and error message files. Therefore, it is highly desirable for each installation to include machine-readable copies of the crash dump and .SIL files when submitting crash dumps as part of an SPR.

The status reports produced by the analysis programs incorporate all of the new SYSTAT output formats and features with one specific change: the WHO column will always explicitly list the account number for each job.

ERPINT
ERPCPY
ERPRTS
ERPDET

The Error Control Package includes a completely new release of a series of programs designed to take advantage of expanded monitor error detection/logging capabilities. The package now provides the user with the means to obtain a list of possibly bad blocks on disk media as well as more informative summary and detail error log reports.

A list of the programs in the package is presented below, along with a brief description of each program's mode of operation and function(s):

Program	Description
ERPINT	Must be RUN to start error logging procedures during time-sharing. Its purpose is to validate or, if

necessary, initialize the error logging file $ERRLOG.FIL. The program also determines if the error crash file $ERRCRS.FIL is to be appended to the error logging file.

ERRCPY — CHAINed to from ERRINT, extracts, formats and stores error messages in the error logging file $ERRLOG.FIL. If directed to by ERRINT, appends the error crash file $ERRCRS.FIL to the error logging file.

ERRDIS — When RUN, the program determines user report requirements by means of an interactive dialogue. Will zero the error logging file if directed to by the user. Produces a summary report of the contents of the error logging file.

ERRDFT — CHAINed to from ERRDIS, produces the full detail report from the error logging file, will compile the possible bad blocks report if necessary.

The Error Control Package utilizes three (3) files during its operation. $ERRLOG.FIL and $ERRCRS.FIL, created by ERRINT and ANALYS respectively, are specially formatted disk files used to store error messages. The third file, $ERRDAT.FIL, is supplied as part of the package and must be present if reports are to be generated. This data file contains specific error analysis information, such as error mnemonics, related registers and register bit descriptions.

ERRDFT scans for possibly bad disk blocks by checking logged disk errors for such things as unreadable data on the last retry for a given read operation. Certain disk errors, such as those on RK06 disk drives, undergo extended checks. The program automatically checks disk errors for possibly bad blocks if the full detail report is requested; a separate listing, containing a brief description of disks with possibly bad blocks is also obtainable.

The Error Control Package provides a valuable diagnostic tool in the trackdown of hardware and/or system level software problems. Each installation is encouraged to read the RSTS/E System Manager's Guide for further information on the package.

E4200 USING THE DEFAULT OPTION

The operator requests the DEFAULT option by typing DEFAULT in response to the initialization code OPTION query. The DEFAULT routines print a series of three queries to determine which start up conditions are to be changed. A brief description of the DEFAULT queries is provided below. Valid responses and additional related options are described in Sections F4210 thru F4240.

DEFAULT Option Queries

Query	Meaning
JOB OR SWAP MAX CHANGES?	Allows operator to change (or establish) either the number of jobs allowed to run during time sharing (JOB MAX) or the length of the job swap area in the swapping files (SWAP MAX).
ANY MEMORY ALLOCATION CHANGES?	Allows operator to locate the Run Time System in an area of memory and to prohibit or allow use of certain portions of memory by the system.
CRASH DUMP?	Allows operator to either enable or disable the crash dump facility.

Additional Related Queries and Responses To ANY MEMORY ALLOCATION CHANGES Query

Additional Related Query	Response(s)	Result
TABLE SUBOPTION:	RETURN key	Causes a list of the options to be to be printed after which the TABLE OPTION query is reprinted and user is allowed to type his response.
	LIST	Causes the current memory allocation table to be printed.
	PARITY	Used as a diagnostic tool to identify various types of parity memory on the system.

	LOCATE	Indicates the user desires to load Run Time System (BASIC-PLUS Monitor) beginning at a certain available portion of memory. Causes the query "FW RUN-TIME SYSTEM ADDRESS IS? to be printed.
	LOCK	Indicates user desires to prevent the system from using some portion of memory and printed LOCKOUT ADDRESS IS?.
	UNLOCK	Indicates the operator desires to allow a currently locked portion of memory to be available for a user job and causes an additional query (UNLOCK ADDRESS IS?) to be printed.
	RESET	Allows the initialization code to set up the memory allocation table. Unlocks all locked memory, locates RTS immediately after the monitor, and makes available for use any new memory added to the system.
	XBUF	Allows allocation of certain portion of memory for disk caching operations. Causes additional query DISK CACHE ADDRESS RANGE to be printed.
	EXIT	Causes exit from routine which changes memory allocation and prints next DEFAULT option query, CRASH DUMP?.
RTS	RETURN key	Causes an explanatory message to be printed and the query to be reprinted, after which the user types the four to seven digit octal number, xxxxxxx.
	xxxxxxx	The four to seven digit octal number representing the 22-bit address of the portion of memory at which the Run-Time System is loaded. (Loading is done from low memory to high memory.) The octal number can be gained by referring to the printout of the current memory breakdown.

LOCKOUT ADDRESS IS?	RETURN key	Causes explanatory message to be printed after which query ADDRESS OF 1K MEMORY SECTION IS? query is printed, allowing user to enter the four to seven digit octal number, xxxxxxx.
	xxxxxxx	The four to seven digit octal number which corresponds to the 22-bit address of the 1K portion of memory to be unlocked for use. The octal number can be gained by referring to the printout of the current memory breakdown.
	xxxxxxx-xxxxxxx	Two numbers, separated by a dash, specifies a range of successive 1K portions of memory to be unlocked.
EXTENDED BUFFER ADDRESS RANGE?	xxxxxxx-xxxxxxx	Two octal numbers, separated by a dash, specifies a range of 1K portions of memory to be received for disk caching operations.
	REMOVE	Remove the memory from disk cache and return it to user job space.

E4210 DEFAULT TABLE OPTION--LIST

The LIST table option prints the current memory allocation breakdown on the console terminal. The breakdown provides useful information on the size of the RSTS/E monitor, the size of the BASIC-PLUS Run Time System (compiler and interpreter), the amount of memory available for user jobs, which portions of memory (if any) are disabled or assigned to disk caching, and the total size of the memory on the machine. The table covers the full physical address space and shows the allocation or state of each range of addresses with appropriate symbols.

E4220 DEFAULT TABLE OPTION--XBUF

The XBUF table option allows the operator to reserve 1K portions of memory for disk caching by the FIP buffering module(1). Any memory reserved for caching is employed solely by the FIP buffering module. The cache memory allows the operator to maintain a minimum amount of FIP buffering on his system because all of the memory reserved for caching must be claimed before small buffers are used by the FIP buffering module.

When the CACHE command is typed, an additional query DISK CACHE ADDRESS RANGE? is printed. A four- to seven- digit octal number must be entered to specify the 22-bit address of the 1K section of memory to be reserved for disk caching. If two numbers, separated by a dash, are entered, a range of successive 1K sections is reserved. The range consists of the 1K sections between the first address and including the 1K section specified by the second address.

The lowest 1K section can start at or above the lowest 28K words of memory (adress 154777(octal)). The range of memory must be available as user space and not reserved for other usage. There is no limit on the address of the upper 1K section. However, the FIP buffering module can use only 132K words for disk caching.

The capacity of FIP buffering depends on the range of memory reserved for caching. The first 1K section of cache holds three disk blocks; each additional 1K section reserved up to 33K words holds 4 disk blocks. The 34th 1K section holds 3 disk blocks and each 1K section up to the 66th 1K section holds 4 disk blocks. In the same manner, the 67th and 110th 1K sections each hold 3 disk blocks and each intervening 1K section up to the 132nd 1K section holds 4 disk blocks. Sections outside the range of 132K words can be reserved for cache but hold no disk blocks since the FIP buffering module cannot use more than 132K words of memory.

---

(1) The FIP buffering module is optional and is not present on all RSTS/E systems. See Section 2.7.22 of the RSTS/E SYSTEM Managers Guide for more information on FIP buffering.

To release disk caching memory for user job area, type REMOVE in response to the DISK CACHE ADDRESS RANGE query. To reserve a portion of memory for disk caching when a portion is currently reserved, type the new range. The CACHE routines automatically release the current disk cache memory to user job space and reserve the new portion specified.

The following sequence demonstrates the use of the CACHE table option.

```
TABLE OPTION ? XBUF
               ----

DISK CACHE ADDRESS RANGE ? 700000-1100000
                           --------------

TABLE OPTION ? LIST
               ----

0000000 - 0123777 ( 21K): EXEC
0124000 - 0213777 ( 14K): BASIC
0214000 - 0603777 ( 62K): LOCKED
0604000 - 0677777 ( 15K): USER
0700000 - 1103777 ( 33K): CACHE
1104000 - 2777777 (239K): USER
3000000 -   END         : NXM

TABLE OPTION ?
```

F4230 DEFAULT TABLE OPTION--RESET

The RESET option allows the initialization code to set up the memory allocation table. All memory which was locked is unlocked. The Run Time System is moved to follow immediately after the Monitor in low physical memory. All memory reserved for disk caching is released to user space. Any new memory added to the system is made available for use. In fact, THE RESET TABLE OPTION MUST BE USED WHENEVER ADDITIONAL NEW MEMORY IS ADDED. The initialization code recognizes and does not attempt to use memory which has been removed. However, the initialization code does not alter the memory allocation table to include previously nonexistent memory unless told to do so.

The types of memory allocation changes described above are most often permanent changes. The effect of RESET is demonstrated in the example below.

```
     ANY MEMORY ALLOCATION CHANGES ? YES
                                    ---

     TABLE OPTION ? LIST
                    ----

     MEMORY ALLOCATION BREAKDOWN:

     0000000 - 0117777 ( 20K) : EXEC
     0120000 - 0237777 ( 20K) : USER
     0240000 - 0253777 (  4K) : LOCKED
     0254000 - 0407777 ( 22K) : USER
     0410000 - 0477777 ( 16K) : BASIC
     0500000 -     END        : NXM

     TABLE OPTION ? RESET
                    -----

     TABLE OPTION ? LIST
                    ----

     MEMORY ALLOCATION BREAKDOWN:

     0000000 - 0117777 ( 20K) : EXEC
     0120000 - 0207777 ( 16K) : BASIC
     0210000 - 0577777 ( 62K) : USER
     0600000 -     END        : NXM

     TABLE OPTION ?
```

F4240 DEFAULT TABLE OPTION--PARITY

THE PARITY table options is used primarily as a diagnostic tool to identify and locate the various types of parity memory on the system. The option is useful to DIGITAL Field Service personnel to verify the memory configuration when the system is installed. It is also useful to identify the physical address of MOS memory on PDP-11/45 systems. The types of memory (MOS or core) are determined by looking at the hardware parity registers. It is impossible for the software to determine the type for non-parity memories.

On a PDP-11/70 configuration, the listing of the parity configuration is always as follows.

```
TABLE OPTION ? PARITY
               ------

PARITY REGISTER BREAKDOWN:
ALL MEMORY IS 11/70 PARITY MEMORY.
```

THE FOLLOWING EXAMPLE SHOWS ALL THE POSSIBILITIES FOR A PDP-11/40 or PDP-11/45 system.

```
TABLE OPTION ? PARITY
               ------

PARITY REGISTER BREAKDOWN:

0000000 - 0077777 ( 16K): 00(C)
0100000 - 0177777 ( 16K): 14(C)
0200000 - 0277777 ( 16K): 00/14
0300000 - 0377777 ( 16K): 02(C)
0400000 - 0477777 ( 16K): 20(M)
0500000 - 0577777 ( 16K): 24(M)
0600000 - 0637777 (  8K): 02(C)
0640000 - 0677777 (  8K): NO
0700000 -   END         : NXM
```

The following are the meanings of the codes used to indicate the type of parity on each section of memory.

NO	No parity memory
NN(C)	Parity core memory (non-interleaved)
NN(M)	Parity MOS memory
NN/MM	Parity core memory (interleaved)
NXM	Non existent memory

NN and MM are the last two octal digits of the address of the parity register responsible for that section of memory. Up to 16 parity registers are in the UNIBUS address range 772100 to 772136. When a parity error is detected, the parity register responsible for that section of memory contains information on the location of the last error detected. One parity register controls up to 24K (MM11-LP) or 32K (MM11-UP) of parity core memory. If core memory is interleaved, two parity registers are used for up to 48K (MM11-LP) or 64K (MM11-UP). For MOS memory one parity register handles one 16K bank. MOS memory cannot be interleaved.

Interleaving means that one bank of memory responds to "even" memory addresses while another bank responds to "odd" addresses as shown below:

ADDRESS		PARITY REGISTER
000000	EVEN	NN
000002	ODD	MM
000004	EVEN	NN
000006	ODD	MM
000010	EVEN	NN
000012	ODD	MM

Interleaving is used because core memories are destructive read out devices. After each read from a core memory, the original data must be restored. However, once the processor receives the data requested, it can go on to other things (presumably another memory reference) during the restore cycle in the memory bank just referenced. The bank remains busy until the restore cycle completes. When memory is interleaved, the probability is low that the same bank will be referenced on the next memory cycle. Hence, interleaving allows some overlap of memory operations with a resultant speed up in program execution.

All of the possibilities are included in the example above. The list below identifies the types of memory and the associated parity register shown in the example.

START ADDRESS OF SECTION	LENGTH OF SECTION	MEMORY TYPE	PARITY REGISTER ADDRESS
000000	16K	Non Interleaved Core	772100(00)
100000	16K	Non Interleaved Core	772114(14)
200000	16K	Interleaved Core	772100(00) EVEN; 772114(14) ODD
300000	16K	Non Interleaved Core	772102(02)
400000	16K	MOS	772120(20)
500000	16K	MOS	772124(24)
600000	8K	Non Interleaved Core	772102(02)
640000	8K	Non Parity Memory	No parity register
700000		Non Existent Memory (NXM)	

Eagan using the peppish option
****
note re added

E4400 RSTS/E ERROR MESSAGE LIST - RECOVERABLE

ERR	Message Printed	Meaning
1	BAD DIRECTORY FOR DEVICE	The directory of the evice referenced is in an unreadable format.
2	ILLEGAL FILE NAME	The filename specified is not acceptable. It contains unacceptable characters or the filename specification format has been violated.
3	ACCOUNT OR DEVICE IN USE	Removal or dismounting of the account or device cannot be done since one or more users are currently using it.
4	NO ROOM FOR USER ON DEVICE	Storage space allowed for the current user on the device specified has been used or the device as a whole is too full to accept further data.
5	CAN'T FIND FILE OR ACCOUNT	The file or account number specified was not found on the device specified.
6	NOT A VALID DEVICE	Attempt to use an illegal or non-existent device specification.
7	I/O CHANNEL ALREADY OPEN	An attempt was made to open one of the twelve I/O channels which has already been opened by the program.
8	DEVICE NOT AVAILABLE	The device requested is currently reserved by another user.
9	I/O CHANNEL NOT OPEN	Attempt to perform I/O on one of the twelve channels which has not been previously opened in the program.

10	PROTECTION VIOLATION	The user was prohibited from performing the requested operation because the kind of operation was illegal (such as input from a line printer) or because the user did not have the privileges necessary (such as deleting a protected file).
11	END OF FILE ON DEVICE	Attempt to perform input beyond the end of a data file; or a BASIC source file is called into memory and is found to contain no END statement.
12	FATAL SYSTEM I/O FAILURE	An I/O error has occurred on the system level. The user has no guarantee that the last operation has been performed.
13	USER DATA ERROR ON DEVICE	One or more characters may have been transmitted incorrectly due to a parity error, bad punch combination on a card, or similar error.
14	DEVICE HUNG OR WRITE LOCKED	User should check hardware condition of device requested. Possible causes of this error include a line printer out of paper or high-speed reader being off-line.
15	KEYBOARD WAIT EXHAUSTED	Time requested by wait statement has been exhausted with no input received from the specified keyboard.
16	NAME OR ACCOUNT NOW EXISTS	An attempt was made to rename a file with the name of a file which already exists, or an attempt was made by the system manager to insert an account number which is already within the system.
17	TOO MANY OPEN FILES ON UNIT	Only one open DECtape output file is permitted per DECtape drive. Only one open file per magtape drive is permitted.
18	ILLEGAL SYS() USAGE	Illegal use of the SYS system function.

19	DISK BLOCK IS INTERLOCKED	The requested disk block segment is already in use (locked) by some other user.
20	PACK IDS DON'T MATCH	The identification code for the specified disk pack does not match the identification code already on the pack.
21	DISK PACK IS NOT MOUNTED	No disk pack is mounted on the specified disk drive.
22	DISK PACK IS LOCKED OUT	The disk pack specified is mounted but temporarily disabled.
23	ILLEGAL CLUSTER SIZE	The specified cluster size is unacceptable.
24	DISK PACK IS PRIVATE	The current user does not have access to the specified private disk pack.
25	DISK PACK NEEDS "CLEANING"	Non-fatal disk mounting error; use the CLEAN operation in UTILTY.
26	FATAL DISK PACK MOUNT ERROR	Fatal disk mounting error. Disk cannot be successfully mounted.
27	I/O TO DETACHED KEYBOARD	I/O was attempted to a hung up dataset or to the previous, but now detached, console keyboard for the job.
28	PROGRAMMABLE ^C TRAP	ON ERROR-GOTO subroutine was entered through a program trapped CTRL/C. See a description of the SYS system function.
30	DEVICE NOT FILE STRUCTURED	An attempt is made to access a device, other than a disk, DECtape, or magtape device, as a file-structured device. This error occurs, for example, when the user attempts to gain a directory listing of a non-directory device.
31	ILLEGAL BYTE COUNT FOR I/O	The buffer size specified in the RECORDSIZE option of the OPEN statement or in the COUNT OPTION OF THE PUT statement is not a multiple of the block size of the device being used for I/O.

32	I/O BUFF AVAILABLE FOR FCB	When the user accesses a file under programmed control in RSTS/E, a system control structure called an FCB requires one small buffer and one small buffer is not available for the FCB.
33	UNIBUS TIMEOUT FATAL TRAP	This hardware error occurs when an attempt is made to address nonexistent memory or an odd address using the PEEK function. An occurrence of this error message in any other case is cause for an SPR.
34	RESERVED INSTRUCTION TRAP	An attempt is made to execute an illegal or reserved instruction or an FPP instruction when floating point hardware is not available.
35	MEMORY MANAGEMENT VIOLATION	This hardware error occurs when an illegal Monitor address is specified using the PEEK function. Generation of the error message in situations other than using PEEK is cause for an SPR.
36	SP(R6) STACK OVERFLOW	An attempt to extend the hardware stack beyond its legal size is encountered.
37	DISK ERROR DURING SWAP	A hardware error occurs when a user's job is swapped into or out of memory. The contents of the user's job area are lost but the job remains logged into the system and is reinitialized to run the NONAME program.
38	MEMORY PARITY ERROR	A parity error was detected in the memory occupied by this job.
39 - 41	not assigned	
42	VIRTUAL BUFFER TOO LARGE	Virtual core buffers must be 512 bytes long.
43	VIRTUAL ARRAY NOT ON DISK	A non-disk device is open on the channel upon which the virtual array is referenced.

44	MATRIX OR ARRAY TOO BIG	In-core array size is too large.
45	VIRTUAL ARRAY NOT YET OPEN	An attempt was made to use a virtual array before opening the corresponding disk file.
46	ILLEGAL I/O CHANNEL	Attempt was made to open a file on an I/O channel outside the range of the integer numbers 1 to 12.
47	LINE TOO LONG	Attempt to input a line longer than 255 characters (which includes any line terminator). Buffer overflows.
48	FLOATING POINT ERROR	Floating point overflow or underflow. If no transfer is made to an error handling routine, a 0 is returned as the floating point value.
49	ARGUMENT TOO LARGE IN EXP	Acceptable arguments are within the approximate range -89<arg<+88. The value returned is zero.
50	not assigned	
51	INTEGER ERROR	Attempt to use a number as an integer when that number is outside the allowable integer range. If no transfer is made to an error handling routine, a 0 is returned as the integer value.
52	ILLEGAL NUMBER	Improperly formed input or value. For example, "1..2" is an improperly formed number.
52	ILLEGAL ARGUMENT IN LOG	Negative or zero argument to log function. Value returned is the argument as passed to the function.
54	IMAGINARY SQUARE ROOTS	Attempt to take square root of a number less than zero. The value returned is the square root of the absolute value of the argument.
55	SUBSCRIPT OUT OF RANGE	Attempt to reference an array element beyond the number of elements created for the array when it was dimensioned.

56	CAN'T INVERT MATRIX	Attempt to invert singular or nearly singular matrix.
57	OUT OF DATA	The DATA list was exhausted and a READ requested additional data.
58	ON STATEMENT OUT OF RANGE	The index value in an ON-GOTO or ON-GOSUB statement is less then one or greater than the number of line numbers in the list.
59	NOT ENOUGH DATA IN RECORD	An INPUT statement did not find enough data in one line to satisfy all the specified variables.
60	INTEGER OVERFLOW, FOR LOOP	The integer index in FOR loop attempted to go beyond 32766 or below -32766.
61	DIVISION BY 0	Attempt by the user program to divide some quantity by zero. If no transfer is made to an error handler routine, a 0 is returned as the result.

E4900 RSTS/E ERROR MESSAGE LIST - NON-RECOVERABLE

Message Printed	Meaning
ARGUMENTS DON'T MATCH	Arguments in a function call do not match, in number or in type, the arguments defined for the function.
BAD LINE NUMBER PAIR	Line numbers specified in a LIST or DELETE command were formatted incorrectly.
BAD NUMBER IN PRINT-USING	Format specified in the PRINT-USING string cannot be used to print one or more values.
CAN'T COMPILE STATEMENT	
CAN'T CONTINUE	Program was stopped or ended at a spot from which execution cannot be resumed.
CATASTROPHIC ERROR	The user program data structures are destroyed. This normally indicates a BASIC-PLUS malfunction and, if re-producible, should be reported to DEC on a Software Performance Report form.
DATA TYPE ERROR	Incorrect usage of floating-point, integer, or character string format variable or constant where some other data type was necessary.
DEF WITHOUT FNEND	A second DEF statement was encountered in the processing of a user function without an FNEND statement terminating the first user function definition.
END OF STATEMENT NOT SEEN	Statement contains too many elements to be processed correctly.
EXECUTE ONLY FILE	Attempt was made to add, delete or list a statement in a compiled (.BAC) format file.

EXPRESSION TOO COMPLICATED	This error usually occurs when parentheses have been nested too deeply. The depth allowable is dependent on the individual expression.
FIELD OVERFLOWS BUFFER	Attempt to use FIELD to allocate more space than exists in the specified buffer.
FILE EXISTS-USE REPLACE	A file of the name specified in a SAVE command already exists. In order to save the current program under the name specified, use the REPLACE command.
FNEND WITHOUT DEF	An FNEND statement was encountered in the user program without a previous DEF statement being seen.
FNEND WITHOUT FUNCTION CALL	A FNEND statement was encountered in the user program without a previous function call having been executed. Function has been placed incorrectly among executable statements or an extra FNEND statement has been found.
FOR WITHOUT NEXT	A FOR statement was encountered in the user program without a corresponding NEXT statement to terminate the loop.
ILLEGAL CONTIONAL CLAUSE	Incorrectly formatted condition expression.
ILLEGAL DEF NESTING	The range of one function definition crosses the range of another function definition.
ILLEGAL DUMMY VARIABLE	One of the variables in the dummy variable list of a user-defined function is not a legal variable name.
ILLEGAL EXPRESSION	Double operators, missing operators, mismatched parentheses, or some similar error has been found in an expression.

ILLEGAL FIELD VARIABLE	The FIELD variable specified is unacceptable.
ILLEGAL FN REDEFINITION	Attempt was made to redefine a user function.
ILLEGAL FUNCTION NAME	Attempt was made to define a function with a function name not subscribing to the established format.
ILLEGAL IF STATEMENT	Incorrectly formatted IF statement.
ILLEGAL IN IMMEDIATE MODE	User issued a statement for execution in immediate mode which can only be performed as part of a program.
ILLEGAL LINE NUMBER(S)	Line number reference outside the range 1<n<32767.
ILLEGAL MAGTAPE() USAGE	Improper use of the MAGTAPE function.
ILLEGAL MODE MIXING	String and numeric operations cannot be mixed.
ILLEGAL STATEMENT	Attempt was made to execute a statement that did not compile without errors.
ILLEGAL SYMBOL	An unrecognizable character was encountered. For example, a line consisting of a * character.
ILLEGAL VERB	The BASIC verb portion of the statement cannot be recognized.
INCONSISTENT FUNCTION USAGE	A function is being redefined in a manner inconsistent in the number or type of arguments with one or more calls to that function existing in the program.
INCONSISTENT SUBSCRIPT USE	A subscripted variable is being used with a different number of dimensions from the number with which it was originally defined.

K OF CORE USED	Message printed by LENGTH command, preceded by the appropriate number describing the user program currently in core to the nearest 1K.
LITERAL STRING NEEDED	A variable name was used where a numeric or character string was necessary.
MATRIX DIMENSION ERROR	Attempt was made to dimension a matrix to more than two dimensions, or an error was made in the syntax of a DIM statement.
MATRY OR ARRAY WITHOUT DIM	A matrix or array element was referenced beyond the range of an implicitly dimensioned matrix.
MAXIMUM CORE EXCEEDED	User program grew to be too large to run or compile in the area of core assigned to each user at the given installation.
MISSING SPECIAL FEATURE	User program employs a BASIC-PLUS feature not present on the given installation.
MODIFIER ERROR	Attempt to use one of the statement modifiers (FOR, WHILE, UNTIL, IF, or UNLESS) incorrectly.
NEXT WITHOUT FOR	A NEXT statement was encountered in the user program without a previous FOR statement having been seen.
NO LOGINS	Message printed if the system is full and cannot accept additional users or if further logins are disabled by the system manager.
NOT A RANDOM ACCESS DEVICE	Attempt to perform random access I/O to a non-random access device.
NOT ENOUGH AVAILABLE CORE	The already compiled user program is too large to run in the area of core assigned to each user at the given installation.
NUMBER IS NEEDED	A character string or variable name was used where a number was necessary.

1 OR 2 DIMENSIONS ONLY	Attempt was made to dimension a matrix to more than two dimensions.
ON STATEMENT NEEDS GOTO	A statement beginning with ON does not contain a GOTO or GOSUB clause.
PLEASE SAY HELLO	User not logged into the system has typed something other than a legal, logged-out command to the system.
PLEASE USE THE RUN COMMAND	A transfer of control (as in a GOTO, GOSUB or IF-GOTO statement) cannot be performed from immediate mode.
PRINT-USING BUFFER OVERFLOW	Format specified contains a field too large to be manipulated by the PRINT-USING statement.
PRINT-USING FORMAT ERROR	An error was made in the construction of the string used to supply the output format in a PRINT-USING statement.
PROGRAM LOST-SORRY	A fatal system error has occurred which caused the user program to be lost.
REDIMENSIONED ARRAY	Usage of an array or matrix within the user program has caused BASIC-PLUS to redimension the array implicitly.
RESUME AND NO ERROR	A RESUME statement was encountered where no error had occurred to cause a transfer into an error handling routine via the ON ERROR-GOTO statement.
RETURN WITHOUT GOSUB	RETURN statement encountered in the user program without a previous GOSUB statement having been executed.
STATEMENT NOT FOUND	Reference is made within the program to a line number which is not within the program.
STOP	STOP statement was executed. The user can usually continue program execution by typing CONT and the RETURN key.

STRING IS NEEDED	A number or variable name was used where a character string was necessary.
SYNTAX ERROR	BASIC-PLUS statement was incorrectly formatted.
TEXT TRUNCATED	No BASIC-PLUS statement can be more than 255 characters long.
TOO FEW ARGUMENTS	The function has been called with a number of arguments not equal to the number defined for the function.
TOO MANY ARGUMENTS	A user-defined function may have up to five arguments.
UNDEFINED FUNCTION CALLED	BASIC-PLUS interpreted some statement component as a function call for which there is no defined function (system or user).
WHAT?	Command or immediate mode statement entered to BASIC-PLUS could not be processed. Illegal verb or improper format error most likely.
WRONG MATH PACKAGE	Program was compiled with in incompatible version of RSTS. Program source must be recompiled.

E5000 HARDWARE CONSIDERATIONS

E5100 ASSIGNMENT OF FLOATING ADDRESSES

E5200 FLOATING VECTOR SEQUENCES
---

more to be added

E6000 DISTRIBUTION MEDIA COPY PROCEDURES

E6100 ROLLIN COPY PROCEDURES

ROLLIN is a stand-alone program which when loaded into memory, either from its distribution medium or from a ROLLIN formatted DECtape or magtape, overwrites any resident monitor present, and runs by itself in memory. ROLLIN is used to transfer data quickly between a disk and either DECtape or magtape or between DECpack disk cartridges or disk packs. Disk devices handled by ROLLIN are the RF11, RC11, RK11 and RP11-C. ROLLIN assumes no file structure on disk or DECtape; transfers are performed in image mode. Magtapes are treated as file structured devices in that each ROLLIN file is preceded by a DOS/BATCH-11 compatible file label.

ROLLIN DOES NOT SUPPORT RP04, RK06 OR TU16 DRIVES.

When transferring data onto DECtape or magtape, ROLLIN automatically writes an initial record containing a tape sequence number called a reel label. For DECtape transfers, the reel label also contains the number of blocks of data transferred. The reel label guards against mounting tapes out of sequence when returning data to a disk device.

Preceding all data records on DECtape or the first file on a magtape, ROLLIN copies a core image of itself. This image permits ROLLIN to be bootstrapped from DECtape or magtape to load the remainder of the tape.

E6110 BOOTSTRAPPING--BOLLIN

Mount the DECtape or magtape containing BOLLIN on unit 0. Consult Section F2100 to perform the actual bootstrapping procedure.

When BOLLIN has successfully been read into memory it will print:

```
BOLLIN V07
*
```

and wait for a command string to be typed.

E6120 COPY AND VERIFYING OPERATION--BOLLIN

The BOLLIN program allows the user to perform a copy operation and, optionally, to specify a verify action in conjunction with the copy operation. The verify action is specified by the use of the /VERIFY switch in the copy command string. Devices upon which a copy is performed are the RK01 or RK05 disk drive, and RP11-C/RP01 disk drives.

NOTE: BOLLIN DOES NOT SUPPORT RP04, RK06 OR TU16 DRIVES.

In the execution of the copy operation, no header or label information is automatically written. An exact image of the medium on the input device is transferred to the same type medium on the output device. The verify action, if specified, is performed as a second pass of the copy operation. The following sections describe procedures to perform copy operations for DECpack and disk pack devices.

E6130 DECPACK COPY PROCEDURES--ROLLIN

Copies of DECpack cartridges can be made by using either RK03 or RK05 disk drives. To safeguard the input cartridge, the drive on which it is mounted can be write locked by using the /WL switch, either in a single action or in combination with the copy command string. Either of the following two commands is acceptable.

```
$DK1:/WL
$DK0:<DK1/WL
```

To format a cartridge prior to writing a copy to it, the user specifies the /FORMAT switch on the output side of the command string. Verification can be specified also by using the /VE switch. The following command string demonstrates the use of the /FORMAT and /VE switches with write lock.

```
$DK1:/FO<DK0:/WL/VE
```

Upon initialization of execution of the above command string, the following message is printed.

```
STARTING RK FORMAT PASS
```

When formatting is completed and copying begins, the following message is printed.

```
END RK FORMAT PASS
```

When copying is complete, the following message is printed.

```
STARTING VERIFICATION PASS
```

If any other messages are printed prior to the pound sign (#), the copy operation was not successful. The error message indicates the drive unit number on which the error occurred.

E6140 DISK PACK COPY PROCEDURES--ROLLIN

The user copies a disk pack mounted on an RP11-C/RP03 drive in the same manner as he copies DECpack cartridges. He may use the /FORMAT and /VERIFY switches similarly, with one variation.

Formatting a disk pack on an RP03 drive requires enabling of the format toggle switch on the RP11-C controller. The ROLLIN program prints messages which instruct the user to set and disable the format switch. The following command string causes the disk pack mounted on RP03 drive unit 1 to be formatted and an image copy to be transferred to it and verified from RP03 drive unit 0.

```
*DP1:/FO<DP0:/VE
```

The messages informing the user of the actions taken are the same as those occurring when a DECpack cartridge is formatted, copied and verified. The only exception is that the messages to set and disable the format toggle switch on the RP11-C controller replace the messages STARTING RK FORMAT PASS and END RK FORMAT PASS.

E6150 ERROR AND INFORMATION MESSAGES--ROLLIN

Error messages printed by the ROLLIN program are simple and require little interpretation by the operator.

SYNTAX ERROR, COMMAND IGNORED.

DISK ERROR,--REQUEST KILLED.

SPECIFIED DEVICE DOES NOT EXIST.

DISK ERROR ON UNIT N -- REQUEST KILLED.

NO OUTPUT FILENAME SPECIFIED.

VERIFICATION ERROR-COPY IS BAD

VERIFY IS NOT IMPLEMENTED FOR THIS COMMAND

ERROR DURING FORMAT PASS - RESTART

DISK NOT READY - TYPE CR TO TRY FORMAT AGAIN

```
E7000  SYSTEM GENERATION QUESTION SUMMARY
       ----------------------------------

     **10-SEP-76**


     BEGINNING OF RSTS/E SYSTEM GENERATION.


     QUESTIONS COME IN LONG AND SHORT  FORMS.   IF
     YOU  ARE  FAMILIAR  WITH THEM, ANSWER "S" FOR
     SHORT;  OTHERWISE, ANSWER "L" FOR LONG FORM.

     FORM ?                          *S * L/Q

     THE SYSTEM WILL AUTOMATICALLY SUPPLY  ANSWERS
     TO  ALL  OF ITS PROMPTS.  YOU HAVE THE OPTION
     OF  ACCEPTING   THE   ANSWER,   SUPPLYING   A
     DIFFERENT  ANSWER OR REQUESTING THE FULL
     PRINTED MESSAGE.  THE HARDWARE ANSWERS WILL
     BE  MEANINGFUL  ONLY  IF YOU ARE GENERATING A
     SYSTEM  FOR  THE  COMPUTER   THAT   YOU   ARE
     CURRENTLY  RUNNING ON.  IF YOU ARE GENERATING
     A SYSTEM FOR THIS COMPUTER TYPE  "YES",  ELSE
     TYPE "NO"

     SAME SYSTEM ?                   *Y *

     THE RSTS/E SYSTEM IS DISTRIBUTED  ON  DECTAPE
     (DT),   MAGTAPE,  RK05  (DK)  AND  RK06  (DM)
     CARTRIDGE DISKS, OR MAY HAVE BEEN  ALREADY
     TRANSFERRED  TO THE SYSTEM DISK (SY).  IN THE
     CASE OF MAGTAPE, A DISTINCTION MUST  BE  MADE
     BETWEEN TWO CLASSES OF DRIVES.  TS03 AND TU10
     DRIVES  ARE  REFERENCED  BY  THE  DEVICE NAME
     "MT".   TU16  AND  TU45  DRIVES  USE THE NAME
     "MM".  ENTER THE TYPE OF DISTRIBUTION  MEDIUM
     AND  DRIVE  TYPE  (IF MAGTAPE) BEING USED FOR
     THIS SYSTEM GENERATION (DT,MT,MM,DK,DM,SY).

     DISTRIBUTION MEDIUM ?          *SY* MM

     THE GENERATED SYSTEM CAN BE WRITTEN ONTO  THE
     CURRENT  SYSTEM DISK (SY), ANOTHER DISK DRIVE
     (DF,DK,DM,DP, OR DR), TS03  OR  TU10  MAGTAPE
     DRIVE  (MT), TU16 OR TU45 MAGTAPE DRIVE (MM),
     OR DECTAPE (DT).

     OUTPUT MEDIUM ?                *SY*

     A RF DISK DRIVE HAS LIMITED STORAGE CAPACITY.
     IF  A  RF  DRIVE  IS BEING USED AS THE SYSTEM
     DISK DURING THE SYSGEN PROCEDURE, FILES  WILL
     HAVE  TO BE DELETED FROM THAT DISK AFTER THEY
     ARE USED.  WILL THE SYSTEM DISK BE A RF  (YES
```

```
OR NO).

DELETE FILES

LP FOR SYSGEN ?                 *Y * N

THIS PROGRAM CAN BE USED TO GENERATE A
MONITOR AND/OR A BASIC-PLUS RUN-TIME SYSTEM.
DO YOU WISH TO GENERATE A MONITOR (YES OR
NO).

GENERATE MONITOR ?              *Y *

THE MONITOR SAVE IMAGE LIBRARY (SIL) WILL
HAVE A NAME OF FROM 1 TO 6 ALPHANUMERIC
CHARACTERS AND AN EXTENSION OF "SIL". PLEASE
SPECIFY THE NAME YOU WANT.

MONITOR NAME ?                  *RSTS*

YOU HAVE THE OPTION OF GENERATING THE
BASIC-PLUS RUN TIME SYSTEM IN ADDITION TO
GENERATING THE MONITOR. DO YOU WANT TO
GENERATE BASIC-PLUS AT THIS TIME (YES OR NO).

GENERATE BASIC-PLUS?            *Y *

THE BASIC-PLUS SAVE IMAGE LIBRARY (SIL) WILL
HAVE A NAME OF FROM 1 TO 6 ALPHANUMERIC
CHARACTERS AND AN EXTENSION OF "RTS". PLEASE
SPECIFY THE NAME YOU WANT.

BASIC-PLUS RTS NAME ?           *BASIC*

NOW YOU MUST SPECIFY THE HARDWARE
CONFIGURATION ON WHICH THIS RSTS/E SYSTEM
WILL RUN.

THE RSTS/E SYSTEM CLOCK CAN BE EITHER A
KW11-L LINE FREQUENCY CLOCK OR A KW11-P
PROGRAMMABLE CLOCK. THE KW11-P CLOCK HAS AN
INTERNAL CRYSTAL OSCILLATOR WHICH CAN BE USED
AS THE SYSTEM TIME BASE IN AREAS WHERE THE AC
POWER FREQUENCY IS NOT ACCURATE. IF YOU HAVE
THE KW11-L CLOCK, ANDSWER "L". FOR THE
KW11-P, ANSWER "P" TO USE THE AC LINE
FREQUENCY AS THE TIME BASE, OR ANSWER "C" TO
USE THE CRYSTAL OSCILLATOR (L, P, OR C).

CLOCK ?                         *C *

THE AC POWER FREQUENCY IS NORMALLY 60 HERTZ
IN THE UNITED STATES, BUT ELSEWHERE IT CAN BE
50 HERTZ. WHAT IS THE AC POWER FREQUENCY AT
WHICH THIS SYSTEM WILL RUN (50 OR 60).
```

```
AC FREQ ?                       #60#

THE KW11-P 10KHZ CRYSTAL OSCILLATOR WILL BE
USED AS THE SYSTEM TIME BASE.  THIS FREQUENCY
IS DIVIDED IN THE HARDWARE TO PROVIDE THE
DESIRED  INTERRUPT RATE.  FOR RSTS/E THE
INTERRUPT RATE CAN BE ANY MULTIPLE OF 50HZ
BETWEEN 50 AND 1000 HZ.  THE LARGER MULTIPLES
PROVIDE BETTER TIME RESOLUTION FOR  JOB
ACCOUNTING AND SCHEDULING PURPOSES AT THE
SMALL EXPENSE OF ADDITIONAL CLOCK INTERRUPT
OVERHEAD.  PLEASE ENTER THE INTERRUPT RATE
DESIRED (50,100,150,...,1000).

KW11P INTERRUPT RATE ?          #100#

THE NEXT FEW QUESTIONS DEAL WITH THE  NUMBERS
AND  TYPES  OF  TERMINAL  INTERFACES  ON  THE
SYSTEM.

SERIAL ASCII TERMINALS CONNECTED DIRECTLY  TO
THE  COMPUTER  AND  THOSE  CONNECTED  THROUGH
LEASED PRIVATE TELEPHONE LINES (NOT  DIAL-UP)
MAY  USE EITHER OF TWO CLASSES OF SINGLE LINE
INTERFACES    (OR    SEVERAL    MULTIPLEXERS,
DISCUSSED  LATER).  THE FIRST CLASS INCLUDES
THE KL11, LC11, DL11A, AND DL11B  INTERFACES.
SPECIFY THE TOTAL NUMBER OF THESE SINGLE SINE
INTERFACES (1 TO 16  -  INCLUDE  THE  CONSOLE
TERMINAL).

KL11,LC11,DL11A,DL11B'S ?       #02#

THE SECOND CLASS OF  SINGLE  LINE  INTERFACES
USED  TO CONNECT TERMINALS LOCALLY OR THROUGH
LEASED TELEPHONE LINES (NOT DIAL-UP) INCLUDES
THE  DL11C  AND  DL11D  INTERFACES.  HOW MANY
DL11C'S AND DL11D'S ARE ON THIS SYSTEM (0  TO
31).

DL11C, DL11D'S ?                #01#

THERE ARE TWO KINDS OF INDIVIDUAL  INTERFACES
FOR  AUTOMATIC  ANSWER  DATASETS  USED ON THE
DIAL-UP  TELEPHONE  NETWORK.  THE  FIRST  OF
THESE IS THE DC11.  HOW MANY DC11-DA LINES DO
YOU HAVE (0 TO 32).

DC11'S ?                        #00#

THE SECOND KIND IF  INDIVIDUAL  LINE  DIAL-UP
INTERFACE IS THE DL11E.  HOW MANY DL11E'S ARE
ON THIS SYSTEM (0 TO 31).

DL11E'S ?                       #00#
```

```
FACH DJ11 MULTIPLEXER CAN CONNFCT UP TO 16
TERMINALS TO THE SYSTEM.  PLEASE ENTFR THE
TOTAL NUMBER OF DJ11'S ATTACHED TO THIS
SYSTEM (0 TO 16).

DJ11'S ?                        *00*

EACH DH11 MULTIPLFXER CAN CONNECT UP TO 16
TERMINALS TO THE SYSTEM.  PLEASE ENTER THE
TOTAL NUMBER OF DH11'S ATTACHED TO THIS
SYSTEM (0 TO 16).

DH11'S ?                        *03*

FOR THIS DH11 UNIT, ENTER THE NUMBER OF LINES
WHICH WILL BE USED NOW OR IN THE FORSEEABLE
FUTURE.  A RESPONSE OF 16 WILL PERMIT ALL
LINFS TO BE USED.  IF THE NUMBER OF LINES
FNABLED (N) IS LFSS THAN 16 RSTS/E WILL NOT
BE CONFIGURFD FOR AND WILL NOT RECOGNIZE
LINES N THRU 15 ON THIS DH11 UNIT.  FNTFR THE
NUMBER OF LINES ENABLED (0 TO 16).

DH11 UNIT 00 LINES ENABLED ? *16*

DH11 UNIT 01 LINES ENABLED ? *16*

DH11 UNIT 02 LINES ENABLED ? *16*

DIAL-UP TELEPHONE LINES CAN BE CONNECTED
THROUGH AN AUTOMATIC ANSWER DATASET TO A DH11
MULTIPLEXER WITH A DM11-BB MODEM CONTROLLER.
NOTE ALL DH11-AD'S INCLUDE A DM11-BB AS AN
INTEGRAL PART OF THE DH11.  IF YOU WISH TO
INCLUDE SUPPORT FOR DATASETS ON DH11'S ANSWER
YES.  OTHERWISE ANSWER NO.

DATASET SUPPORT FOR DH11'S ? *Y *

EACH DZ11 MULTIPLEXER CAN CONNECT UP TO 8
TERMINALS TO THF SYSTEM.  PLEASE ENTER THE
TOTAL NUMBER OF DZ11'S ATTACHED TO THIS
SYSTEM (0 TO 16).

DZ11'S ?                        *00*

PSEUDO KEYBOARDS PERMIT INTERACTIVE JOBS TO
BE RUN WITHOUT TYING UP A REAL TERMINAL.
THEY ARE PRIMARILY INTENDED FOR USE BY A
BATCH CONTROL PROGRAM WHICH FEEDS COMMANDS TO
ONE OR MORE PSEUDO KEYBOARDS DEDICATED TO
RUNNING BACKGROUND TASKS.  HOW MANY PSEUDO
KEYBOARDS WOULD YOU LIKE TO HAVE (0 TO 127).

PSEUDO KEYBOARDS ?              *04* 20
```

F1

```
RSTS/E IS CAPABLE OF SUPPORTING IBM 2741
COMPATIBLE TERMINALS ON DL11D, DL11E, AND
DC11 SINGLE LINE INTERFACES, OR ON DH11 OR
DZ11 MULTIPLEXERS.  IF YOU DO NOT WANT ANY
2741 SUPPORT THEN ANSWER "NO".  OTHERWISE
ANSWER "YES".

2741 SUPPORT ?                  *NO*

AN OPTIONAL FEATURE OF THE RSTS/E TERMINAL
SERVICE ALLOWS ONE JOB TO INTERACT WITH
SEVERAL TERMINALS THROUGH SPECIAL FORMS OF
THE RECORD I/O GET AND PUT STATEMENTS.  THIS
FEATURE IS USEFUL IN APPLICATIONS WHERE THE
SAME BASIC FUNCTION IS PERFORMED ON SEVERAL
TERMINALS AND A SEPARATE JOB FOR EACH IS
UNDESIREABLE OR AT LEAST INEFFICIENT.  WOULD
YOU LIKE TO INCLUDE THIS FEATURE (YES OR NO).

MULTI-TERMINAL SERVICE ?        *Y *

ECHO CONTROL IS AN OPTIONAL FEATURE OF THE
RSTS/E TERMINAL SERVICE WHICH ALLOWS ANY FULL
DUPLEX TERMINAL TO FUNCTION LIKE A BLOCK MODE
TERMINAL.  THIS FEATURE ENABLES A PROGRAM TO
DEFINE FIXED LENGTH INPUT FIELDS AND TO DEFER
ECHOING OF ALL TYPED CHARACTERS UNTIL THEY
ARE ACTUALLY REQUESTED.  IT IS USEFUL IN DATA
ENTRY APPLICATIONS OR WHEREVER THE APPEARANCE
OF TERMINAL OUTPUT MUST BE PRECISELY
CONTROLLED.  WOULD YOU LIKE TO INCLUDE THIS
FEATURE IN YOUR SYSTEM (YES OR NO).

ECHO CONTROL ?                  *Y *

THE NEXT QUESTIONS DEAL WITH THE NUMBERS AND
KINDS OF DISK UNITS ON THIS SYSTEM.

THE RC11 CONTROLLER IS USED TO CONTROL UP TO
4 OF THE RS64 64K WORD FIXED-HEAD DISKS.  IF
THERE ARE NONE OF THESE UNITS, ANSWER NO;
OTHERWISE, ANSWER YES.

RC11/RS64'S ?                   *NO*

THE RF11 CONTROLLER IS USED TO CONTROL UP TO
8 OF THE RS11 256K WORD FIXED-HEAD DISKS.  IF
THIS SYSTEM HAS NONE OF THESE DISKS, ANSWER
NO;  OTHERWISE, ANSWER YES.

RF/RS11'S ?                     *NO*

THE RJS04/RJS03 DISK SYSTEM CONSISTS OF AN
RH11 DEVICE CONTROLLER AND FROM ONE TO EIGHT
RS03 (256K) OR RS04 (512K WORD) FIXED-HEAD
```

```
DISK  DRIVES.   PLEASE ENTER THE TOTAL NUMBER
OF RS03 AND RS04 DRIVES ON THIS SYSTEM (0  TO
8).

RS03/RS04'S ?                  *02*

THE RK11 CONTROLLER IS USED TO CONTROL UP  TO
8  OF  THE  RK05 1.2 MILLION WORD MOVING-HEAD
CARTRIDGE DISK  DRIVES.   NOTE  AN  RK05F  IS
EQUIVALENT TO 2 RK05'S.  HOW MANY RK05 DRIVES
ARE THERE (0 TO 8).

RK05'S?                        *01* 2

IF  YOU  WISH  TO  USE  THE  OVERLAPPED  SEEK
DRIVERS  TYPE  YES.  OTHERWISE IF YOU WISH TO
CONSERVE MEMORY, AT  THE  SACRIFICE  OF  SOME
EFFICIENCY    IN    I/O,    BY    USING    THE
NON-OVERLAPPED DRIVE TYPE NO.

OVERLAPPED SEEK ?              *Y *

THE RK611 CONTROLLER IS USED TO CONTROL UP TO
EIGHT  OF  THE  RK06  SIX  MILLION  WORD  TOP
LOADING CARTRIDGE  DISK  DRIVES.   ENTER  THE
NUMBER  OF  RK06  DRIVES ON THIS SYSTEM (0 TO
8).

RK06'S ?                       *00*

THE RP11 CONTROLLER IS USED  TO  CONTROL  ANY
COMBINATION  OF  UP TO EIGHT RP02 (10 MILLION
WORD) OR RP03 (20 MILLION  WORD)  MOVING-HEAD
DISK  PACK DRIVES.  ENTER THE TOTAL NUMBER OF
RP02 AND RP03 DRIVES ON THIS SYSTEM (0 TO 8).

RP02/RP03'S ?                  *00*

THE RJP04/5/6 DISK SYSTEM CONSISTS OF AN RH11
CONTROLLER AND ANY COMBINATION OF UP TO EIGHT
RP04 (44  MILLION  WORD),  RP05  (44  MILLION
WORD),  OR RP06 (88 MILLION WORD) MOVING-HEAD
DISK PACK DRIVES.  TYPE THE TOTAL  NUMBER  OF
RP04,   RP05,   AND   RP06   DRIVES  IN  THIS
CONFIGURATION (0 TO 8).

RP04/RP05/RP06'S ?             *03*

OVERLAPPED SEEK ?              *Y *


THE  NEXT  FEW  QUESTIONS  DEAL  WITH  THE
PERIPHERAL  DEVICES  ATTACHED  TO THIS RSTS/E
SYSTEM.
```

```
THE TJU16 MAGTAPE SYSTEM CONSISTS OF AN  RH11
MASSBUS  INTERFACE, TM02 CONTROLLER, AND FROM
ONE TO EIGHT TU16 OR TU45  TAPE  DRIVES.   IF
THIS SYSTEM HAS NO TU16 OR TU45 MAGTAPE UNITS
ANSWER 0;  OTHERWISE, HOW MANY TU16 AND  TU45
DRIVES ARE ON THIS SYSTEM (1 TO 8).

TU16'S ?                        #01# 2

THE TM11 MAGTAPE CONTROLLER CAN SUPPORT UP TO
EIGHT  TU10  OR  TS03 MAGTAPE DRIVES.  PLEASE
ENTER THE NUMBER OF TU10 AND TS03  DRIVES  ON
THIS SYSTEM (0 TO 8).

TU10'S ?                        #00#

THE TC11 DECTAPE CONTROLLER CAN CONTROL UP TO
8  SINGLE  DRIVES.   IF  THIS  SYSTEM  HAS NO
DECTAPE AT ALL,  ANSWER  0;   OTHERWISE,  HOW
MANY  SINGLE  DECTAPE DRIVES, NOT TU56'S, ARE
THERE (1 TO 8 - NOTE  THAT  EACH  TU56  DUAL
DRIVE HAS 2).

DECTAPES ?                      #02#

THE RSTS/E SYSTEM CAN HAVE UP TO  EIGHT  LINE
PRINTERS  OF  THE  LP11,  LS11, OR LV11 TYPE.
THESE PRINTERS ARE REFERRED TO BY THE  DEVICE
NAMES  LP0:   THROUGH  LP7:.   IF THERE IS NO
LINE PRINTER, ANSWER  0;   OTHERWISE;   ENTER
THE NUMBER OF PRINTERS (1 TO 8).

PRINTERS ?                      #02#

THE RX11 FLOPPY DISK  SYSTEM  CONSISTS  OF  A
UNIBUS  INTERFACE  AND  THE  RX01 FLOPPY DISK
SUBSYSTEM.   THE   RX01   INCLUDES   A   DISK
CONTROLLER   AND   TWO  FLOPPY  DISK  DRIVES.
RSTS/E SUPPORTS UP TO FOUR RX11 SYSTEMS FOR A
MAXIMUM  OF  EIGHT  DRIVES.  PLEASE ENTER THE
NUMBER OF FLOPPY DISK DRIVES ATTACHED TO THIS
SYSTEM (0 TO 8).

RX01'S ?                        #00#

DOES THIS SYSTEM HAVE  A  CR11  PUNCHED  CARD
READER  OR  A CM11 MARKED CARD READER (YES OR
NO).

CR11/CM11 CARD READER ?         #NO#

DOES  THIS  SYSTEM  HAVE  A  CD11  HIGH-SPEED
PUNCHED CARD READER (YES OR NO).

CD11 CARD READER ?              #NO#
```

```
IS THERE A HIGH-SPEED PAPER TAPE READER  (YES
OR NO).

P.T.READER ?                    *Y *

IS THERE A HIGH-SPEED PAPER TAPE  PUNCH  (YES
OR NO).

P.T.PUNCH ?                     *Y *

DECNET/E IS A SET OF MODULES WHICH  IMPLEMENT
THE   NETWORK   SERVICES PROTOCOL  AND  USER
INTERFACES REQUIRED FOR DECNET COMMUNICATION.
THE DECNET/E SOFTWARE IS SOLD AND DISTRIBUTED
AS A SEPARATE PACKAGE.  IT IS NOT INCLUDED IN
THE  STANDARD  RSTS/E  KIT.   DO  YOU  HAVE A
DECNET/E  KIT  AND  WISH  TO  INCLUDE  DECNET
SUPPORT IN THIS SYSTEM (YES OR NO).

DECNET NETWORK SUPPORT ?      *NO*

THE  RSTS/E   2780   PACKAGE   EMULATES   THE
OPERATION  OF  THE  IBM  2780  MODEL  1  DATA
TRANSMISSION TERMINAL.  THE PACKAGE PERMITS
THE RSTS/E SYSTEM TO COMMUNICATE WITH ANY IBM
SYSTEM WHICH SUPPORTS THE DEVICE, OR  ANOTHER
RSTS/E 2780 SYSTEM.  THE 2780 PACKAGE IS SOLD
AND  SUPPORTED  BY  THE  DEC  COMMUNICATIONS
GROUP.   IT  IS  NOT INCLUDED IN THE STANDARD
RSTS/E KIT.  DO YOU HAVE THE 2780 PACKAGE AND
WISH  TO  INCLUDE  IT  IN THIS SYSTEM (YES OR
NO).

2780 SUPPORT ?                  *NO*

THE  REMAINING  QUESTIONS  DEAL  WITH  THE
CAPACITY  AND  FEATURES OF THIS RSTS/E SYSTEM
PROVIDED AT THE SYSTEM MANAGER'S OPTION.

WITH SUFFICIENT HARDWARE RSTS/E CAN HANDLE UP
TO 63 SIMULTANEOUS JOBS.  WHAT IS THE MAXIMUM
NUMBER OF JOBS (JOBMAX) TO BE  USED  AT  THIS
INSTALLATION (1 TO 63).

MAXIMUM JOBS ?                  *10* 40

SMALL BUFFERS ARE 16 WORD BLOCKS  IN  MONITOR
STORAGE  THAT  ARE  DYNAMICALLY  ALLOCATED AS
INPUT  AND  OUTPUT  BUFFERS,  FILE  PARAMETER
BLOCKS, ETC.  STORAGE MAY BE ALLOCATED FOR 30
TO 999 SMALL BUFFERS.  THE RECOMMENDED NUMBER
IS  AT  LEAST  10 FOR EACH POSSIBLE JOB.  HOW
MANY SMALL BUFFERS WOULD  THIS  INSTALLATION
LIKE TO HAVE (30 TO 999).
```

```
SMALL BUFFERS ?                 #440# 500

LOGICAL NAMES CAN BE ASSIGNED TO DEVICES ON A
SYSTEM WIDE BASIS.  THE ASSIGNMENT TABLE
CONSISTS OF FIVE WORDS FOR EACH ALLOWED
ASSIGNMENT.  PLEASE ENTER THE MAXIMUM NUMBER
OF SYSTEM WIDE LOGICAL ASSIGNMENTS THAT WILL
BE IN USE AT ANY TIME (0-30).

SYSTEM WIDE LOGICALS ?          #10#

WHEN POWER IS RESTORED AFTER A POWER FAILURE,
RSTS/E CAN DELAY THE RECOVERY FOR 1 TO 300
SECONDS (5 MINUTES).  ANSWER WITH YOUR DELAY
FACTOR (1 TO 300).

DELAY FACTOR ?                  #001#

AN OPTIONAL FEATURE OF THE RSTS/E SYSTEM
ALLOWS THE FILE PROCESSOR (FIP) TO USE FREE
BUFFERS, OR DEDICATED SECTIONS OF MEMORY
(CALLED THE EXTENDED BUFFER POOL) TO STORE
DIRECTORY INFORMATION.  THIS IMPROVES THE
SPEED OF DIRECTORY PROCESSING BY NOT
REREADING FREQUENTLY ACCESSED DIRECTORY
INFORMATION, IF YOU WANT FIP BUFFERING TYPE
YES, OTHERWISE TYPE NO.

FIP BUFFERING ?                 #Y #

IF THERE IS SUFFICIENT MEMORY AVAILABLE ON
THIS SYSTEM, THE MOST FREQUENTLY USED
NON-RESIDENT DISK HANDLING CODE CAN BE MADE
RESIDENT IN ORDER TO PROVIDE BETTER SYSTEM
PERFORMANCE.  SHOULD THIS DISK PROCESSING BE
DONE BY RESIDENT CODE (YES OR NO).

RESIDENT DISK HANDLING ?        #Y #

THE ROUTINES WHICH IMPLEMENT THE SEND AND
RECEIVE SYSTEM FUNCTION CALLS ARE ALSO
NON-RESIDENT.  IF YOUR APPLICATIONS REQUIRE
INTER-JOB COMMUNICATION, YOU MAY WANT
SEND/RECEIVE RESIDENT.  PLEASE ANSWER YES OR
NO.

RESIDENT SEND/RECEIVE ?         #NO#

THE ROUTINES TO IMPLEMENT THE SIMPLE SYS
CALLS ARE NORMALLY NON-RESIDENT.  OVERALL
SYSTEM PERFORMANCE CAN BE IMPROVED BY MAKING
THIS CODE RESIDENT.  DO YOU WANT THE SIMPLE
SYS CALL CODE RESIDENT (YES OR NO).

RESIDENT SIMPLE SYS CALLS ?  #NO# Y
```

```
NON-RESIDENT CODE IS USED TO DELETE OR RENAME
A FILE.  IF YOU WANT THE FILE DELETE/RENAME
CODE TO BE RESIDENT TYPE "YES";  OTHERWISE
TYPE "NO".

RESIDENT FILE DELETE/RENAME ?  $NO$

NON-RESIDENT CODE IS USED TO DO LOGINS,
ATTACHES AND ASSIGN ATTRIBUTES.  IF YOU WANT
THE LOGIN, ATTACH AND ATTRIBUTE CODE
RESIDENT, TYPE "YES", ELSE ANSWER "NO".

RES. LOGIN/ATTACH/ATTRIBUTE ?  $NO$

NON-RESIDENT CODE IS USED TO LIST DISK
DIRECTORIES (CATALOG COMMAND) AND DO FILE
LOOKUPS.  IF YOU WANT THE DIRECTORY LISTER
AND FILE LOOKUP CODE RESIDENT, TYPE "YES",
OTHERWISE ANSWER "NO".

RESIDENT CATALOG/LOOKUP ?    $NO$


THE FOLLOWING QUESTIONS DEAL WITH THE
BASIC-PLUS RUN TIME SYSTEM

WILL THIS SOFTWARE RUN ON A COMPUTER WITH A
FLOATING POINT PROCESSOR (YES OR NO).

FPP ?                        $Y $

FLOATING POINT NUMBERS ARE REPRESENTED
INTERNALLY AS TWO 16-BIT WORDS, GIVING SEVEN
SIGNIFICANT DIGITS.  IT IS POSSIBLE TO
MAINTAIN 17 SIGNIFICANT DIGITS BY USING 4
WORDS PER NUMBER.  THE FOUR WORD MATH
PACKAGES ALSO INCLUDE THE SCALED ARITHMETIC
FEATURE.  WOULD THIS INSTALLATION PREFER TO
USE 2 OR 4 WORD MATH.

MATH PRECISION ?             $02$

IT IS POSSIBLE TO SAVE SPACE IN THE
BASIC-PLUS SYSTEM BY OMITTING THE EXTENDED
FUNCTIONS SIN, COS, SQR, LOG, ETC.,  IF THEY
ARE NOT NEEDED.  DOES THIS INSTALLATION NEED
TO COMPUTE THESE FUNCTIONS (YES OR NO).

FUNCTIONS ?                  $Y $

THE RSTS/E SYSTEM NORMALLY REPORTS THE TIME
OF DAY AS AM/PM (E.G., 9:13 PM).  IT IS
POSSIBLE TO HAVE 24-HOUR TIME (E.G., 21:13)
REPORTED INSTEAD.  DO YOU WANT "AM/PM" OR
"24-HOUR" TIME REPORTED.
```

```
TIME FORMAT ?                    $AM$

THE RSTS/E SYSTEM NORMALLY PRINTS THE MONTH
AS THREE ALPHABETIC CHARACTERS (E.G.,
04-JUL-76).  IF DESIRED THE NUMERIC VALUE OF
THE MONTH CAN BE PRINTED INSTEAD (E.G.,
76,07,04).  DO YOU WANT AN ALPHABETIC MONTH
(YES OR NO).

ALPHABETIC MONTH ?               $Y $

SPECIAL OUTPUT FORMATTING CAN BE DONE USING
THE "PRINT USING" STATEMENT.  WOULD THIS
INSTALLATION LIKE TO HAVE THIS OPTIONAL
FEATURE (YES OR NO).

PRINT USING ?                    $Y $

BASIC-PLUS PERMITS THE USER TO OPERATE ON AN
ENTIRE MATRIX USING JUST A SINGLE STATEMENT.
THESE STATEMENTS ARE THE "MAT" STATEMENTS.
WOULD THIS INSTALLATION LIKE TO HAVE THIS
OPTIONAL FEATURE.

MATRICES ?                       $NO$ Y

AN OPTIONAL FEATURE OF BASIC-PLUS ALLOWS
ARITHMETIC OPERATIONS TO BE DONE BETWEEN
NUMBERS REPRESENTED BY STRINGS.  THIS FEATURE
CAN BE USED TO OBTAIN GREATER ACCURACY IN
ARITHMETIC OPERATIONS.  DO YOU WANT STRING
ARITHMETIC (YES OR NO).

STRING ARITHMETIC ?              $NO$ Y


THE SYSTEM GENERATION DIALOG IS FINISHED.  IF
YOU HAVE ANY SPECIAL REQUIREMENTS WHICH
REQUIRE EDITING THE GENERATED FILE
CONFIG.MAC, SYSTEM CONFIGURATION FILE, OR
SYSGEN.CTL, BATCH CONTROL FILE, YOU MAY DO IT
NOW.  WHEN READY TYPE "R SYSBAT".

Ready
```

E7100 SAMPLE SYSTEM GENERATION (11/34 AND RK)

Note: The following is a sample SYSGEN on an 11/34 with programmer's console and using RSTS/E "MT" distribution media for the SYSGEN.

```
$MT

RSTS V06B (MT0)

OPTION: DS

DD-MMM-YY? 20-OCT-76
           ---------
00:00? 10:47
       -----

DISK? RP
      --
UNIT? 0
      --
PACK ID? SYSPAC
         ------
PACK CLUSTER SIZE? 4
                   --
SATT,SYS BASE? <LF>
               ----
MFD PASSWORD? SYSMFD
              ------
MFD CLUSTER SIZE? 4
                  --
PUB, PRI, OR SYS? SYS
                  ---
LIBRARY PASSWORD? SYSLIB
                  ------
LIBRARY UFD CLUSTER SIZE? 8
                          --
DATE LAST MODIFIED? Y
                    --
FORMAT? N
        --
PATTERNS? 0
          --
PROCEED (Y OR N)? Y
                  --

OPTION: COPY
        ----

20-OCT-76? <LF>
```

```
          ----
10:48? <LF>
       ----

TO WHICH DISK? DP0:
               ----


RSTS V06B (DP0)


OPTION: HA LI
        -----

   NAME  ADDRESS VECTOR  COMMENTS
   TT:     177560   060
   RP:     176700   254   RP03 UNITS: NONE
   TM:     172520   224
   LP0:    177514   200
   RX0:    177170   264
   KL0:    176500   300
   KL1:    176510   310


OPTION: RE
        --
   20-OCT-76? <LF>
   10:49?<LF>

   DISK? RP
         --
   UNIT? 0
         --

   CLEAN? Y
          --

   DISK IS BEING CLEANED - WAIT ...

REFRESH SUBOPTION? CH
                   --

SWAP.SYS CHANGES? Y
                  --

  SIZE? 224
        ---

  BASE? <LF>
        ----

SWAP0.SYS CHANGES? <LF>
```

```
                    ----

SWAP1.SYS CHANGES? <LF>
                    ----

SWAP3.SYS CHANGES? <LF>
                    ----

OVR.SYS CHANGES? <LF>
                  ----

ERR.SYS CHANGES? <LF>
                  ----

BUFF.SYS CHANGES? <LF>
                   ----

CRASH.SYS CHANGES? Y
                   --

  SIZE? 80
        --

  BASE? <LF>
        ----

OTHER FILES? <LF>
             ----

  REFRESH SUBOPTION? <LF>
                     ----


OPTION: IN
        --

  SIL? SYSGEN
       ------


OPTION: DF
        --

NO DEFAULTS ARE CURRENTLY SET

YOU CURRENTLY HAVE: JOB MAX = 2, SWAP MAX = 8K.

JOB MAX OR SWAP MAX CHANGES? Y
                             --

  NEW JOB MAX? <LF>
               ----

  NEW SWAP MAX? 28
                --
```

```
YOU CURRENTLY HAVE: JOB MAX = 2, SWAPMAX = 28K.

JOB MAX OR SWAP MAX CHANGES? <LF>
                             ----

RUN TIME SYSTEM? RT11
                 ----

ERROR MESSAGE FILE? ERR
                    ---

INSTALLATION NAME? UETP TEST

  MEMORY ALLOCATION TABLE:

     0K: 00000000 - 00107777 (  18K) : EXEC
    18K: 00110000 - 00127777 (   4K) : RTS (RT11)
    22K: 00130000 - 00377777 (  42K) : USER
    64K: 00400000 - END              : NXM


  TABLE SUBOPTION?                  .

YOU CURRENTLY HAVE CRASH DUMP DISABLED.

CARSH DUMP? Y

  MAGTAPE LABELLING DEFAULT (NONE)? DOS


  OPTION:

  YOUR CURRENTLY HAVE: JOB MAX = 2, SWAP MAX = 28K.

  YOU CURRENTLY HAVE CRASH DUMP ENABLED.

  20-OCT-76?
  10:51?
  BUFF.SYS NOT FOUND OR TOO SMALL - DECTAPE DISABLED

  13 DEVICES DISABLED

  ?CAN'T FIND FILE OR ACCOUNT
  .R MTO:CREATE




  ^C
  HELLO 1 /2
  PASSWORD:
  1 OTHER USER IS LOGGED IN UNDER THIS ACCOUNT

  .
  .ASSIGN MTO:IN
```

```
.R IN:PIP.SAV
*SY:*.*=IN:$PIP.SAV,$LOGOUT.SAV,$UTILTY.SAV
*SY:*.*=IN:$MACRO.SAV,$CREF.SAV,$LINK.SAV,$SILUS.SAV,$HOOK.SAV/N
*SY:*.*=IN:$SYSGEN.SAV,$SYSBAT.SAV,$ERR.STB/N
*^C

.R LOGOUT
CONFIRM: Y
SAVED ALL DISK FILES: 268 BLOCKS IN USE
JOB 2 USER 1,2 LOGGED OFF KB1 AT 20-OCT-76 10:53 AM
1 OTHER JOB STILL LOGGED IN UNDER THIS ACCOUNT
SYSTEM RSTS V06B-01 UETP TEST
RUN TIME WAS 0.8 SECONDS
ELAPSED TIME WAS 2 MINUTES
GOOD MORNING
```

```
**20-OCT-76**

BEGINNING OF RSTS/E SYSTEM GENERATION.

QUESTIONS COME IN LONG AND SHORT FORMS.  IF YOU
ARE FAMILIAR WITH THEM, ANSWER "S" FOR SHORT;
OTHERWISE, ANSWER "L" FOR LONG FORM.

FORM ?                          *S * S/O

SAME SYSTEM ?                   *Y * <LF>

DISTRIBUTION MEDIUM ?           *SY* MT
                                 --

OUTPUT MEDIUM ?                 *SY* <LF>

DELETE FILES ?                  *NO* <LF>

LP FOR SYSGEN ?                 *Y * <LF>

GENERATE MONITOR ?              *Y * <LF>

MONITOR NAME ?                  *RSTS* <LF>

GENERATE BASIC-PLUS ?           *Y * <LF>

BASIC-PLUS RTS NAME ?           *BASIC* <LF>

NOW YOU MUST SPECIFY THE HARDWARE CONFIGURATION ON
WHICH THIS RSTS/E SYSTEM WILL RUN.

CLOCK ?                         *L * <LF>

AC FREQ ?                       *60* <LF>

KL11,LC11,DL11A,DL11B'S ?       *03* <LF>

DL11C, DL11D'S ?                *00* <LF>

DC11'S ?                        *00* <LF>

DL11E'S ?                       *00* <LF>

DJ11'S ?                        *00* <LF>

DH11'S ?                        *00* <LF>

DZ11'S ?                        *00* <LF>

PSEUDO KEYBOARDS ?              *04* 10
                                 --
```

F2

```
MULTI-TERMINAL SERVICE ?          *Y * <LF>
ECHO CONTROL ?                    *Y * <LF>
RC11/RS64'S ?                     *NO* <LF>
RF/RS11'S ?                       *NO* <LF>
RS03/RS04'S ?                     *00* <LF>
RK05'S ?                          *00* <LF>
RK06'S ?                          *00* <LF>
RP02/RP03'S ?                     *08* 2
                                    --
OVERLAPPED SEEK ?                 *Y * <LF>
RP04/RP05/RP06'S ?                *00* <LF>
TU16'S ?                          *00* <LF>
TU10'S ?                          *08* 1
                                    --
DECTAPES ?                        *00* <LF>
PRINTERS ?                        *01* <LF>
RX01'S ?                          *08* 2
                                    --
CR11/CM11 CARD READER ?           *NO* <LF>
CD11 CARD READER ?                *NO* <LF>
P.T.READER ?                      *NO* <LF>
P.T.PUNCH?                        *NO* <LF>
DECNET NETWORK SUPPORT ?          *NO* <LF>
2780 SUPPORT ?                    *NO* <LF>
MAXIMUM JOBS ?                    *10* 15
                                    --
SMALL BUFFERS ?                   *150 <LF>
SYSTEM WIDE LOGICALS ?            *10* <LF>
DELAY FACTOR ?                    *001* 2
                                    --
```

```
FIP BUFFERING ?                 *Y * <LF>

RESIDENT DISK HANDLING ?        *Y * <LF>

RESIDENT SEND/RECEIVE ?         *NO* <LF>

RESIDENT SIMPLE SYS CALLS ?     *NO* <LF>

RESIDENT FILE DELETE/RENAME ?   *NO* <LF>

RES. LOGIN/ATTACH/ATTRIBUTE ?   *NO* <LF>

RESIDENT CATALOG/LOOKUP ?       *NO* <LF>

THE FOLLOWING QUESTIONS DEAL WITH THE  BASIC-PLUS
RUN TIME SYSTEM

FPP ?                           *NO* <LF>

FIS ?                           *NO* <LF>

MATH PRECISION?                 *02* <LF>

FUNCTIONS ?                     *Y * <LF>

TIME FORMAT ?                   *AM* <LF>

ALPHABETIC MONTH ?              *Y * <LF>

PRINT USING ?                   *Y * <LF>

MATRICES ?                      *NO* Y
                                  --

STRING ARITHMETIC ?             *NO* <LF>


THE SYSTEM GENERATION DIALOG IS FINISHED.  IF YOU
HAVE  ANY  SPECIAL REQUIREMENTS WHICH REQUIRE
EDITING THE GENERATED FILE CONFIG.MAC, SYSTEM
CONFIGURATION FILE, OR SYSGEN.CTL, BATCH CONTROL
FILE, YOU MAY DO  IT  NOW.  WHEN READY TYPE "R
SYSBAT".

.R SYSBAT
--------


SYSGEN  BATCH  PROCESSING  HAS  STARTED.  IF  ANY
PROBLEMS  DEVELOP  DURING THE BATCH PROCESS IT MAY
BE ABORTED BY TYPING "CONTROL/C".  TO RESTART TYPE
"R SYSBAT".
```

```
^C
HELLO 1 /2
PASSWORD:
1 OTHER USER IS LOGGED IN UNDER THIS ACCOUNT

.

MOUNT DEC-11-ORSPA-F-MC9 OR MC7 ON MAGTAPE DRIVE

WITH NO "WRITE RING" AND SET TO "ON LINE"

MOUNT MT:"ORSPAE"-WRITE LOCKED
UNIT ? 0
        --
  .ASSIGN MT0:

  .ASSIGN MT0: IN

  .R PIP.SAV
  *SY:*.*=IN:$COMMON.MAC
  *^C

  .R PIP.SAV
  *SY:*.*=IN:$*.MAC/H/N
  *^C

  .R PIP.SAV
  *SY:*.*=IN:$*.OBJ/H/N
  *^C

  .R PIP.SAV
  *SY:*.*=IN:$*.SAV/H/N
  *^C

  .R PIP.SAV
  *SY:*.*=IN:$*.STB/H/N
  *^C

DISMOUNT MT0:
  .DEASSIGN IN

  .DEASSIGN MT0:


  .ASSIGN SY: ORSPAE

  .R PIP.SAV
  *TBL.LST,TTDVR.LST/D
  ?CAN'T FIND FILE OR ACCOUNT - FILE TBL    .LST -
  CONTINUING
  ?CAN'T FIND FILE OR ACCOUNT - FILE TTDVR .LST -
  CONTINUING
  *^C
```

```
.R PIP.SAV
*RSTS.SAV,EMT.SAV,FIP.SAV,OVR.SAV,NSP.SAV/D
?CAN'T FIND FILE OR ACCOUNT - FILE RSTS  .SAV -
CONTINUING
?CAN'T FIND FILE OR ACCOUNT - FILE EMT   .SAV -
CONTINUING
?CAN'T FIND FILE OR ACCOUNT - FILE FIP   .SAV -
CONTINUING
?CAN'T FIND FILE OR ACCOUNT - FILE OVR   .SAV -
CONTINUING
?CAN'T FIND FILE OR ACCOUNT - FILE NSP   .SAV -
CONTINUING
*^C

.R PIP.SAV
*RSTS.MPA,EMT.MPA,FIP.MAP,OVR.MAP,NSP.MAP/D
?CAN'T FIND FILE OR ACCOUNT - FILE RSTS  .MAP -
CONTINUING
?CAN'T FIND FILE OR ACCOUNT - FILE EMT   .MAP -
CONTINUING
?CAN'T FIND FILE OR ACCOUNT - FILE FIP   .MAP -
CONTINUING
?CAN'T FIND FILE OR ACCOUNT - FILE OVR   .MAP -
CONTINUING
?CAN'T FIND FILE OR ACCOUNT - FILE NSP   .MAP -
CONTINUING
*^C

.R PIP.SAV
*RSTS.STB,EMT.STB,FIP.STB,OVR.STB,NSP.STB/D
?CAN'T FIND FILE OR ACCOUNT - FILE RSTS  .STB -
CONTINUING
?CAN'T FIND FILE OR ACCOUNT - FILE EMT   .STB -
CONTINUING
?CAN'T FIND FILE OR ACCOUNT - FILE FIP   .STB -
CONTINUING
?CAN'T FIND FILE OR ACCOUNT - FILE OVR   .STB -
CONTINUING
?CAN'T FIND FILE OR ACCOUNT - FILE NSP   .STB -
CONTINUING
*^C

.R MACRO.SAV
*TBL,TBL/C=ORSPAE:COMMON,KERNEL,SY:CONFIG,ORSPAE:
CHECK,TBL
ERRORS DETECTED: 0
FREE CORE: 9294. WORDS

*^C

.R MACRO.SAV
*TTDVR,TTDVR/C=ORSPAE:COMMON,KERNEL,SY:CONFIG,ORS
PAE:CHECK,KBDEF,TTDVR
ERRORS DETECTED: 0
FREE CORE: 8438. WORDS
```

```
*^C

.R LINK.SAV
*RSTS/Z,PSTS,RSTS=TRL,TTDVR,$ERR,STB/X/R:0/U:$100
0/I/C
*ORSPAE:RSTS
ROUND SECTION:
? MORBUF
LIBRARY SEARCH:
? DPSFEK
? BUF
?


*^C

.R LINK.SAV
*EMT/Z,EMT,EMT=ORSPAE:EMT,SY:RSTS.STB/X/R:$117000
/U:$1000/C
*ORSPAE:RSTS
ROUND SECTION:
? EMTPAT


*^C

.R LINK.SAV
*FIP/Z,FIP,FIP=ORSPAE:FIP,SY:RSTS.STB/X/R:$117000
?U:$1000/I/C
*ORSPAE:RSTS
ROUND SECTION:
? FIPPAT
LIBRARY SEARCH:
? OPN
? LIN
?


*^C

.R LINK.SAV
*OVR/Z,OVR,OVR=ORSPAE:OVR,SY:FIP.STB,ORSPAE:RSTS/
X/R:$1000


*^C

.R SILUS.SAV
*[0,1]RSTS  .SIL=RSTS,EMT/M,FIP/M/C
*OVR/M/C
*ORSPAE:ODT,DEFALT
*^C

.R PIP.SAV
*BASIC.SAV,BASIC.MPA,BASIC.STB/D
```

```
    ?CAN'T FIND FILE OR ACCOUNT - FILE BASIC .SAV -
    CONTINUING
    ?CAN'T FIND FILE OR ACCOUNT - FILE BASIC .MAP -
    CONTINUING
    ?CAN'T FIND FILE OR ACCOUNT - FILE BASIC .STB -
    CONTINUING

MOUNT DEC-11-ORSPA-3-MC9 OR MC7 ON MAGTAPE DRIVE

WITH NO "WRITE RING" AND SET TO "ON LINE"

MOUNT MT:"      "-WRITE LOCKED
UNIT ? 0
      --
  *^C

  .ASSIGN MT0:

  .ASSIGN MT0: IN

  .R PIP.SAV
  *SY:*.*=IN:$RTS.OBJ/N/H
  *^C

  .R PIP.SAV
  *SY:*.*=IN:$*.OBJ/N
  *^C

DISMOUNT MT0:
  .DEASSIGN IN

  .DEASSIGN MT0:

  .R LINK.SAV
  *BASIC/Z,BASIC,BASIC=ORSPAE:RTS,SY: ERR,STB/X/H:0
  177776/U:0:4000/C
  *ORSPAE:MA2/C
  *ORSPAE:XL2/C
  *ORSPAE:XT2/C
  *ORSPAE:IO/C
  *ORSPAE:PU/C
  *ORSPAE:MX/C
  *ORSPAE:SN/C
  *ORSPAE:TI/C
  *ORSPAE:DA/C
  *ORSPAE:VE
  ROUND SECTION:
  ? PA

  *^C

  .R SILUS.SAV
  *BASIC .RTS=BASIC
  *^C
```

L2

```
.R PIP.SAV
*SY:[0,1]*.*/MODE:16,=BASIC .RTS
*^C

.R LOGOUT
CONFIRM: Y
SAVED ALL DISK FILES; 3304 BLOCKS IN USE
JOB 2 USER 1,2 LOGGED OFF KB1 AT 20-OCT-76 11:15
AM
1 OTHER USER STILL LOGGED IN UNDER THIS ACCOUNT
SYSTEM RSTS V06B-01 UETP TEST
RUN TIME WAS 8 MINUTES, 7.8 SECONDS
ELAPSED TIME WAS 19 MINUTES
GOOD MORNING



BATCH JOB COMPLETED

.R UTILTY
 --------
*NO LOGINS
 --------
*SHUTUP
 ------


RSTS V06B-01 UETP TEST (DP0)

OPTION: IN
        --

  SIL? RSTS
       ----

OPTION: RF
        --

20-OCT-76? <LF>
           ----
11:16? <LF>
       ----

DISK? RP
      --
UNIT? 0
      --
```

```
CLEAN? Y
       --

DISK IS BEING CLEANED - WIAT ...

  REFRESH SUBOPTION? CH
                     --

  SWAP.SYS CHANGES? <LF>

  SWAP0.SYS CHANGES? Y
                     --

    SIZE? 832

    BASE? 2000

  SWAP1.SYS CHANGES? <LF>

  SWAP3.SYS CHANGES? <LF>

  OVR.SYS CHANGES? <LF>

  ERR.SYS CHANGES? <LF>

  BUFF.SYS CHANGES? <LF>

  CRASH.SYS CHANGES? <LF>

  OTHER FILES? <LF>

  REFRESH SUBOPTION? <LF>

OPTION: DE
        --

NO DEFAULTS ARE CURRENTLY SET

YOU CURRENTLY HAVE: JOB MAX = 15, SWAP MAX = 8K,

JOB MAX OR SWAP MAX CHANGES? Y
                             --

  NEW JOB MAX? <LF>

  NEW SWAP MAX? 16
                --

YOUR CURRENTLY HAVE: JOB MAX = 15, SWAP MAX = 16K,

JOB MAX OR SWAP MAX CHNAGES? <LF>

RUN TIME SYSTEM? BASIC
                 -----
```

A3

```
ERROR MESSAGE FILE? ERR
                    ---

INSTALLATION NAME? UETP TEST
                   ---------

  MEMORY ALLOCATION TABLE:

     0K: 00000000 - 00123777 (  21K) : EXEC
    21K: 00124000 - 00223777 (  16K) : RTS (BASIC)
    37K: 00224000 - 00377777 (  27K) : USER
    64K: 00400000 - END              : NXM


  TABLE SUBOPTION? <LF>
                   ----

YOUR CURRENTLY HAVE DRASH DUMP DISABLED.

CRASH DUMP? Y
            --

MAGTAPE LABELLING DEFAULT (NONE)? DOS
                                  ---


OPTION: <LF>

YOU CURRENTL HAVE : JOB MAX = 15, SWAP MAX = 16K.

YOU CURRENTLY HAVE CRASH DUMP ENABLED.

20-OCT-76? <LF>
11:19? <LF>

1 DEVICE DISABLED

?CAN'T FIND FILE OR ACCOUNT
?PROGRAM LOST-SORRY

READY
```

E8150 SAMPLE SYSTEM GENERATION (PDP11/34 AND RK)

The system generation dialog is finished. If you have any special requirements which require editing the generated file config.mac, system configuration file, or sysgen.ctl, batch control file, you may do it now. When ready type "R SYSBAT".

```
READY

.R UTILTY
.NO LOGINS
.SHUTUP
RSTS/E V06B-02 ACCEPTNCE TEST
OPTION:
```

```
E8200  SYSTEM INITIALIZATION
       ---------------------

     OPTION: DSKINT
             ------
     DD-MM-YY? 14-AUG-75
               ---------
     HH:MM? 2:55
            ----

     DISK ? RK
            --
     UNIT ? 0
            --
     PACK ID ? SYSPAK
               ------
     PACK CLUSTER SIZE ? 2
                         --
     MFD PASSWORD ? SYSMFD
                    ------
     MFD CLUSTER SIZE ? 2
                        --
     PUB, PRI, OR SYS ? SYS
                        ---
     LIBRARY PASSWORD ? SYSLIB
                        ------
     LIBRARY UFD CLUSTER SIZE ? 2
                                --
     DATE LAST MODIFIED? Y
                         --
     PATTERNS ? 8
              --
     PROCEED (Y OR N) Y
                      --
     PATTERN #8
     PATTERN #7
     PATTERN #6
     PATTERN #5
     PATTERN #4
     PATTERN #3
     PATTERN #2
     PATTERN #1


     OPTION: REFRESH
     SYSTEM DISK IS BEING CLEANED - WAIT ...

     DD-MMM-YY? 14-AUG-75
               ---------
     HH:MM? 3:10
            ----

     DISK? DK
           --
     UNIT? 0
```

```
      --
REFRESH SUBOPTION ? LIST
                  ----

FILE                          CURRENT REQUIRED          START START
NAME        REQUIRED? EXIST STATUS  SIZE     SIZE     CLUSTER SECTOR

BADB  .SYS YES          SYS     OK        0        0
RSTS  .SIL YES          SYS     OK      274      274       24   49
SATT  .SYS YES          SYS     OK        2        2        1    3
SWAP0 .SYS YES          NO      CRE       0      480
SWAP1 .SYS NO           NO      OK        0        0
SWAP2 .SYS NO           NO      OK        0        0
SWAP3 .SYS NO           NO      OK        0        0
OVR   .SYS NO           NO      OK        0       32
ERR   .SYS NO           NO      OK        0        8
BUFF  .SYS NO           NO      OK        0        0
CRASH .SYS NO           NO      OK        0       24
```

```
      REFRESH SUROPTION ? BAD

      BADS ? LIST

      THERE ARE NO BAD BLOCKS

      REFRESH SUROPTION ? CHANGE
      SATT.SYS CHANGES ? <LF>
      SWAP0.SYS CHANGES ? YES
      SIZE ? 960
      BASE ? 3000
      SWAP1.SYS CHANGES ? <LF>
      SWAP3.SYS CHANGES ? <LF>
      OVR.SYS CHANGES ? <LF>
      ERR.SYS CHANGES ? <LF>
      BUFF.SYS CHANGES ? <LF>
      OTHER FILES? <LF>

FILE                                    CURRENT REQUIRED START START
NAME        REQUIRED?  EXIST STATUS     SIZE    SIZE     CLUSTER SECTOR

BADB  .SYS  YES        SYS   OK          0       0
RSTS  .SIL  YES        SYS   OK        274     274         24   49
SATT  .SYS  YES        SYS   OK          2       2          1    3
SWAP0 .SYS  YES        SYS   OK        960     480       1499 2999
SWAP1 .SYS  NO         NO    OK          0       0
SWAP2 .SYS  NO         NO    OK          0       0
SWAP3 .SYS  NO         NO    OK          0       0
OVR   .SYS  NO         NO    OK          0      32
ERR   .SYS  NO         NO    OK          0       8
BUFF  .SYS  NO         NO    OK          0       0
CRASH .SYS  NO         SYS   OK         24      24          4    9
```

```
OPTION: DE

NO DEFAULTS ARE CURRENTLY SET.

YOU CURRENTLY HAVE: JOB MAX = 15, SWAP MAX = 8K.

JOB MAX OR SWAP MAX CHANGES ? YES

NEW JOB MAX ? <LF>

NEW SWAP MAX ? 16
YOU CURRENTLY HAVE: JOB MAX = 15, SWAP MAX = 16K.

JOB MAX OR SWAP MAX CHANGES ? NO

CURRENT MEMORY ALLOCATION BREAKDOWN:

0000000 - 0067777(        14K: EXEC
0070000 - 0163777(        15K): BASIC
0164000 - 0377777(        35K): USER
0400000 - END     :       NXM


TABLE OPTION ? EXIT

YOU CURRENTLY HAVE: CRASH DUMP DISABLED.

CRASH DUMP? YES

MAPGTAPE LABELLING DEFAULT (NONE) : DOS


RSTS V06B-02 ACCEPTANCE TEST

OPTION: <LF>

YOU CURRENTLY HAVE: JOB MAX = 15, SWAP MAX = 16K.

DD-MMM-YY? 14-AUG-75
HH:MM? 3:05

CAN'T FIND FILE OR ACCOUNT
PROGRAM LOST-SORRY
```

E8300 SYSTEM LIBRARY BUILDING PRINTOUT

```
      READY

      RUN MM0:BUILD
      BUILD V06B-03 RSTS V06B-02 ACCEPTNCE TEST
      SYSTEM BUILD? Y
      TARGET SYSTEM DEVICE <SY0:>?
      SOURCE INPUT DEVICE? MM0
      LIBRARY OUTPUT DEVICE <SY:>?
      LIBRARY ACCOUNT <[1,2]>?
      ASSIGN [1,2]
      OLD MM0:$LOGIN
      COMPILE SY0:@LOGIN
      CHAIN 'MM0:$BUILD' 9200

      READY



      READY



      READY


      READY

      ^C
      HELLO

      RSTS V06B-02 ACCEPTNCE TEST Job 2 KB0 03-Dec-76  12:53 PM
      # 1 / 2
      PASSWORD:
      JOB 1 IS DETACHED UNDER THIS ACCOUNT
      JOB NUMBER TO ATTACH TO?
      1 OTHER USER IS LOGGED IN UNDER THIS ACCOUNT




      READY


      ASSIGN [1,2]

      READY

      !********** BUILD.CTL - STANDARD LIBRARY PROGRAMS

      OLD MM0:$LOGOUT
```

```
READY
COMPILE SY0:@LOGOUT
READY
OLD MM0:$UTILTY
READY
COMPILE SY0:@UTILTY
READY
OLD MM0:$UTILT1
READY
COMPILE SY0:@UTILT1
READY
OLD MM0:$INIT
READY
COMPILE SY0:@INIT
READY
OLD MM0:$SHUTUP
READY
COMPILE SY0:@SHUTUP
READY
OLD MM0:$EPRINT
READY
COMPILE SY0:@ERRINT
READY
OLD MM0:$ERRCPY
READY
COMPILE SY0:@ERRCPY
READY
OLD MM0:$PIP
```

```
READY
COMPILE SY0:@PIP<40>
READY
OLD MM0:@DIRECT
READY
COMPILE SY0:@DIRECT
READY
OLD MM0:@TTYSET
READY
COMPILE SY0:@TTYSET
READY
OLD MM0:@SYSTAT
READY
COMPILE SY0:@SYSTAT
READY
OLD MM0:@EDIT
READY
COMPILE SY0:@EDIT<40>
READY
OLD MM0:@EDITCH
READY
COMPILE SY0:@EDITCH<40>
READY
OLD MM0:@BUILD
READY
COMPILE SY0:@BUILD
READY
OLD MM0:@ERRDIS
```

```
READY
COMPILE SY :@ERRDIS
READY
OLD MM0:@ERRDET
READY
COMPILE SY :@ERRDET
READY
OLD MM0:@ANALYS
READY
COMPILE SY :@ANALYS
READY
OLD MM0:@ANALY1
READY
COMPILE SY :@ANALY1
READY
OLD MM0:@SYSCAT
READY
COMPILE SY :@SYSCAT
READY
OLD MM0:@PRIOR
READY
COMPILE SY :@PRIOR
READY
OLD MM0:@ODT
READY
COMPILE SY :@ODT
READY
OLD MM0:@REACT
```

```
READY
COMPILE SY :@RFACT
READY
OLD MM0:@REORDR
READY
COMPILE SY :@REORDR
READY
OLD MM0:@DSKINT
READY
COMPILE SY :@DSKINT
READY
OLD MM0:@UMOUNT
READY
COMPILE SY :@UMOUNT
READY
OLD MM0:@COPY
READY
COMPILE SY :@COPY<40>
READY
OLD MM0:@FILCOM
READY
COMPILE SY :@FILCOM<40>
READY
OLD MM0:@QUOLST
READY
COMPILE SY :@QUOLST
READY
OLD MM0:@MONEY
```

```
READY
COMPILE SY :@MONEY<40>
READY
OLD MM0:$GRIPE
READY
COMPILE SY :@GRIPE
READY
OLD MM0:$TALK
READY
COMPILE SY :@TALK
READY
OLD MM0:$PLEASE
READY
COMPILE SY :@PLEASE
READY

OLD MM0:$INUSE

READY

COMPILE SY :@INUSE

READY

RUN SY0:@PIP
PIP      V06B-03 - RSTS V06B-02 ACCEPTNCE TEST
@SY0:@NOTICE.TXT<40>=MM0:$NOTICE.TXT/FA
@SY0:@HELP  .TXT<40>=MM0:$HELP  .TXT/FA
@SY0:@START .CTL     =MM0:$START .CTL/FA
@SY0:@TTY   .CMD     =MM0:$TTY   .CMD/FA
@SY0:@SPOOL .CMD     =MM0:$POOL  .CMD$/FA
@SY0:@RTS   .CMD     =MM0:$RTS   .CMD/FA
@SY0:@CCL   .CMD     =MM0:$CCL   .CMD/FA
@SY0:@CRASH .CTL     =MM0:$CRASH .CTL/FA
@SY0:@ANALYS.CMD     =MM0:$ANALYS.CMD/FA
@SY0:@UTILTY.TXT     =MM0:$UTILTY.TXT/FA
@SY0:@ERRDAT.FIL     =MM0:$ERRDAT.FIL/FA
@SY0:@PIP   .TXT<40>=MM0:$PIP   .TXT/FA
@SY0:@DIRECT.HLP<40>=MM0:$DIRECT.HLP/FA
@SY0:@TTYSET.SPD     =MM0:$TTYSET.SPD/FA
@SY0:@ERRDIS.HLP     =MM0:$ERRDIS.HLP/FA
```

M3

```
#SY0:#ACCT  .SYS    =MM0:#ACCT .SYS/FA
#SY0:#COPY  .TXT<40>=MM0:#COPY .TXT/FA
#=SY0:#LOGIN .BAC<232>/RF
#=SY0:#LOGOUT.BAC<232>/RF
#=SY0:#DIRECT.BAC<232>/RF
#=SY0:#TTYSET.BAC<232>/RF
#=SY0:#SYSTAT.BAC<232>/RF
#=SY :#UMOUNT.BAC<232>/RF
#=SY :#QUOLST.BAC<232>/RF
#=SY :#GRIPE .BAC<232>/RF
#=SY :#TALK  .BAC<232>/RF
#=SY :#PLEASE.BAC<232>/RF
#=SY :#INUSE .BAC<232>/RE
#^C

READY

RUN SY0:#UTILTY
UTILTY V06B-03 RSTS V06B-02 ACCEPTNCE TEST
#LOGINS
#EXIT

READY

^C

READY

HELLO

RSTS V06B-02 ACCEPTNCE TEST JOB 2 [1,2] KB0 03-DEC-76 01:00 PM
JOB 1 IS DETACHED UNDER THIS ACCOUNT
JOB NUMBER TO ATTACH TO?  1
ATTACHING TO JOB 1

BUILD COMPLETE

READY
```

E8350 SAMPLE SYSTEM LIBRARY BUILD (11/34 AND RK)

```
      NOTE:  REMOVE DISK FROM DK1: AND PLACE RSTS/E DISTRIBUTION PACK
             ON DRIVE.

      READY
      MOUNT DK1:ORSLAE/RO

      RUN DK1:BUILD
      BUILD V06B-03 RSTS V06B-02 ACCEPTNCE TEST
      SYSTEM BUILD? Y
      TARGET SYSTEM DEVICE <SY0:>?
      SOURCE INPUT DEVICE? DK1
      LIBRARY OUTPUT DEVICE <SY:>?
      LIBRARY ACCOUNT <[1,2]>?
      ASSIGN [1,2]
      OLD DK1:$LOGIN
      COMPILE SY0:$LOGIN
      CHAIN 'DK1:$BUILD' 9200

      READY


      READY
```

```
      READY


      READY

      ^C
      HELLO

      RSTS V06B-02 ACCEPTNCE TEST Job 2 KB0 03-Dec-76  12:53 PM
      # 1 / 2
      PASSWORD:
      JOB 1 IS DETACHED UNDER THIS ACCOUNT
      JOB NUMBER TO ATTACH TO?
      1 OTHER USER IS LOGGED IN UNDER THIS ACCOUNT


      READY


      ASSIGN [1,2]

      READY
```

```
!********** BUILD.CTL - STANDARD LIBRARY PROGRAMS
OLD DK1:$LOGOUT
READY
COMPILE SY0:$LOGOUT
READY
OLD DK1:$UTILTY
READY
COMPILE SY0:$UTILTY
READY
OLD DK1:$UTILT1
READY
COMPILE SY0:$UTILT1
READY
OLD DK1:$INIT
READY
COMPILE SY0:$INIT
READY
OLD DK1:$SHUTUP
READY
COMPILE SY0:$SHUTUP
READY
OLD DK1:$ERRINT
READY
COMPILE SY0:$ERRINT
READY
OLD DK1:$ERRCPY
READY
COMPILE SY0:$ERRCPY
```

```
READY
OLD DK1:$PIP
READY
COMPILE SY0:$PIP<40>
READY
OLD DK1:$DIRECT
READY
COMPILE SY0:$DIRECT
READY
OLD DK1:$TTYSET
READY
COMPILE SY0:$TTYSET
READY
OLD DK1:$SYSTAT
READY
COMPILE SY0:$SYSTAT
READY
OLD DK1:$EDIT
READY
COMPILE SY0:$EDIT<40>
READY
OLD DK1:$EDITCH
READY
COMPILE SY0:$EDITCH<40>
READY
OLD DK1:$BUILD
READY
COMPILE SY0:$BUILD
```

```
READY
OLD DK1:$ERRDIS
READY
COMPILE SY :@ERRDIS
READY
OLD DK1:$ERRDET
READY
COMPILE SY :@ERRDET
READY
OLD DK1:$ANALYS
READY
COMPILE SY :@ANALYS
READY
OLD DK1:$ANALY1
READY
COMPILE SY :@ANALY1
READY
OLD DK1:$SYSCAT
READY
COMPILE SY :@SYSCAT
READY
OLD DK1:$PRIOR
READY
COMPILE SY :@PRIOR
READY
OLD DK1:$ODT
READY
COMPILE SY :@ODT
```

```
READY
OLD DK1:$PFACT
READY
COMPILE SY :@PFACT
READY
OLD DK1:$REORDR
READY
COMPILE SY :@REORDR
READY
OLD DK1:$DSKINT
READY
COMPILE SY :@DSKINT
READY
OLD DK1:$UMOUNT
READY
COMPILE SY :@UMOUNT
READY
OLD DK1:$COPY
READY
COMPILE SY :@COPY<40>
READY
OLD DK1:$FILCOM
READY
COMPILE SY :@FILCOM<40>
READY
OLD DK1:$QUOLST
READY
COMPILE SY :@QUOLST
```

```
READY
OLD DK1:$MONEY
READY
COMPILE SY :$MONEY<40>
READY
OLD DK1:$GRIPE
READY
COMPILE SY :$GRIPE
READY
OLD DK1:$TALK
READY
COMPILE SY :$TALK
READY
OLD DK1:$PLEASE
READY
COMPILE SY :$PLEASE
READY
OLD DK1:$INUSE
READY
COMPILE SY :$INUSE
READY
RUN SY0:$PIP
PIP      V06B-03 - RSTS V06B-02 ACCEPTNCE TEST
*SY0:$NOTICE.TXT<40>=DK1:$NOTICE.TXT/FA
*SY0:$HELP  .TXT<40>=DK1:$HELP  .TXT/FA
*SY0:$START .CTL    =DK1:$START .CTL/FA
*SY0:$TTY   .CMD    =DK1:$TTY   .CMD/FA
*SY0:$SPOOL .CMD    =DK1:$SPOOL  .CMD$/FA
*SY0:$RTS   .CMD    =DK1:$RTS   .CMD/FA
*SY0:$CCL   .CMD    =DK1:$CCL   .CMD/FA
*SY0:$CRASH .CTL    =DK1:$CRASH .CTL/FA
*SY0:$ANALYS.CMD    =DK1:$ANALYS.CMD/FA
*SY0:$UTILTY.TXT    =DK1:$UTILTY.TXT/FA
*SY0:$ERRDAT.FIL    =DK1:$ERRDAT.FIL/FA
```

```
#SY0:#PIP    .TXT<40>=DK1:#PIP    .TXT/FA
#SY0:#DIRECT.HLP<40>=DK1:#DIRECT.HLP/FA
#SY0:#TTYSET.SPD    =DK1:#TTYSET.SPD/FA
#SY0:#ERRDIS.HLP    =DK1:#ERRDIS.HLP/FA
#SY0:#ACCT  .SYS    =DK1:#ACCT  .SYS/FA
#SY0:#COPY  .TXT<40>=DK1:#COPY  .TXT/FA
#=SY0:#LOGIN .BAC<232>/RE
#=SY0:#LOGOUT.BAC<232>/RE
#=SY0:#DIRECT.BAC<232>/RE
#=SY0:#TTYSET.BAC<232>/RE
#=SY0:#SYSTAT.BAC<232>/RF
#=SY :#UMOUNT.BAC<232>/RE
#=SY :#QUOLST.BAC<232>/RE
#=SY :#GRIPE .BAC<232>/RE
#=SY :#TALK  .BAC<232>/RE
#=SY :#PLEASE.BAC<232>/RE
#=SY :#INUSE .BAC<232>/RF
#^C

READY

RUN SY0:#UTILTY
UTILTY V06B-03 RSTS V06B-02 ACCEPTNCE TEST
#LOGINS
#EXIT

READY

READY

^C

READY

HELLO

RSTS V06B-02 ACCEPTNCE TEST JOB 2 [1,2] KB0 03-DEC-76 01:00 PM
JOB 1 IS DETACHED UNDER THIS ACCOUNT
JOB NUMBER TO ATTACH TO?  1
ATTACHING TO JOB 1

BUILD COMPLETE

READY
```

```
ER400  UETP LIBRARY BUILDING PRINTOUT
       ------------------------------

      RUN $MM:BUILDR$
      RSTS/E UETP LIBRARY BUILDER V06-05
      ********* RSTS V06B-02 ACCEPTNCE TEST
      BUILD UETP ON WHICH SY: 1, ACCOUNT <44>?
      FROM WHICH DEVICE <MT0>:? MM0
      UETP LIBRARY ACCOUNT FOR MM0: IS <(1,2)>?

      CONTROL FILE ACCBLD.CTL FOR MM0:(1,2) IS CREATED
      PROCEEDING TO BUILD ACCTST FILES ON SY:(1,44)



      ^C
      HELLO

      RSTS V06B-02 ACCEPTNCE TEST JOB 2 KB0 03-DEC-76 01:05 PM
      #1/2
      PASSWORD:
      JOB 1 IS DETACHED UNDER THIS ACCOUNT
      JOB NUMBER TO ATTACH TO?
      1 OTHER USER IS LOGGED IN UNDER THIS ACCOUNT




      READY



      !********** ACCBLD.CTL - UETP PROGRAM BUILD
      RUN $REACT
      REACT  V06R-03 RSTS V06B-02 ACCEPTNCE TEST
      SYSTEM ACCOUNT MANAGER
      FUNCTION? E
      PROJ,PROG? 1,44
      DISK:PASSWORD? UETP
      QUOTA? 0
      CLUSTER SIZE? 8
      ACCOUNT NAME? UETP
      PROJ,PROG? ^C

      READY

      ASSIGN (1,44)

      READY

      OLD MM0:ACCTST$

      READY
```

```
COMPILE SY:ACCTST@

READY

OLD MM0:AC0TST@

READY

COMPILE SY:AC0TST@

READY

OLD MM0:AC1TST@

READY

COMPILE SY:AC1TST@

READY

OLD MM0:AC2TST@

READY

COMPILE SY:AC2TST@

READY

OLD MM0:AC3TST@

READY

COMPILE SY:AC3TST@

READY

OLD MM0:AC4TST@

READY

COMPILE SY:AC4TST@

READY

OLD MM0:AC5TST@

READY

COMPILE SY:AC5TST@

READY

OLD MM0:ERRDPY@

READY
```

```
COMPILE SY:ERRDPY/
READY
OLD MM0:ERRDPA/
READY
COMPILE SY:ERRDPA/
READY
OLD MM0:ERRDP1/
READY
COMPILE SY:ERRDP1/
READY
OLD MM0:ERRDP2/
READY
COMPILE SY:ERRDP2/
READY
OLD MM0:DMEXER/
READY
COMPILE SY:DMEXER/
READY
OLD MM0:DSEXER/
READY
COMPILE SY:DSEXER/
READY
OLD MM0:DFEXER/
READY
COMPILE SY:DFEXER/
READY
OLD MM0:DKFXER/
READY
```

ku

```
COMPILE SY:DKEXER$
READY
OLD MM0:DXEXFR$
READY
COMPILE SY:DXEXER$
READY
OLD MM0:MTEXFR$
READY
COMPILE SY:MTEXER$
READY
OLD MM0:DTEXER$
READY
COMPILE SY:DTEXER$
READY
OLD MM0:CPEXER$
READY
COMPILF SY:CPEXER$
READY
OLD MM0:CREXER$
READY
COMPILF SY:CREXER$
READY
OLD MM0:PPEXER$
READY
COMPILE SY:PPEXER$
READY
OLD MM0:PREXER$
READY
```

```
COMPILE SY:PREXER@

READY

OLD MM0:KBEXFR@

READY

COMPILE SY:KBEXER@

READY

OLD MM0:SCRIPT@

READY

COMPILE SY:SCRIPT@

READY

OLD MM0:SCRPTB@

READY

COMPILE SY:SCRPTB@

READY

OLD MM0:CVTSCP@

READY

COMPILE SY:CVTSCP@

READY

OLD MM0:OPTPRM@

READY

COMPILE SY:OPTPRM@

READY

OLD MM0:OPTPRK@

READY

COMPILE SY:OPTPRK@

READY

OLD MM0:OPTPRB@

READY
```

```
COMPILE SY:OPTPRR@

READY

OLD MM0:TDISRT@

READY

COMPILE SY:TDLSRT@

READY

OLD MM0:TDIXOK@

READY

COMPILE SY:TDLXOK@

READY

OLD MM0:TDLMOK@

READY

COMPILE SY:TDLMOK@

READY

OLD MM0:TDLOOK@

READY

COMPILE SY:TDLOOK@

READY

OLD MM0:TDLSOK@

READY

COMPILE SY:TDLSOK@

READY

OLD MM0:CRFILE@

READY

COMPILE SY:CRFILE@

READY

OLD MM0:TAPSRT@

READY
```

```
PFADY

OLD WW0:NEWRPB

READY

COMPILE SY:NEWRPB

RFADY

OLD WW0:VERIFYB

READY

COMPILE SY:VERIFYB

READY

OLD WW0:PANMARB

READY

COMPILE SY:PANMARB

READY

OLD WW0:JSTATSB

READY

COMPILE SY:NFWRRB

READY

OLD WW0:NFWRRB

READY

COMPILE SY:DFCMRGB

READY

OLD WW0:DFCMRGB

READY

COMPILE SY:TAPSRUB

READY

OLD WW0:TAPSRUB

READY

COMPILE SY:TAPSRTB
```

```
COMPILE SY:JSTATS@

READY

OLD MM0:DSTATS$

READY

COMPILE SY:DSTATS@

READY

OLD MM0:CPUTST$

READY

COMPILE SY:CPUTST@

READY

OLD MM0:FILMIN$

READY

COMPILE SY:FILMIN@

READY

OLD MM0:UDA$

READY

COMPILE SY:UDA@

READY

OLD MM0:FLOPPY$

READY

COMPILE SY:FLOPPY@

READY

OLD MM0:DA$

READY

COMPILE SY:DA@

READY

RUN $PIP

PIP      V06B-03 - RSTS V06B-02 ACCEPTNCE TEST
```

```
$SY:CLUMSY.SCP@<MM0:CLUMSY.SCP$
$SY:FILES.SCP@<MM0:FILES.SCP$
$SY:IMMED.SCP@<MM0:IMMED.SCP$
$SY:FDIT.SCP@<MM0:FDIT.SCP$
$SY:TTY.SCP@<MM0:TTY.SCP$
$SY:RANDAC.SCP@<MM0:RANDAC.SCP$
$SY:RELEAS.SEQ@<MM0:RELEAS.SEQ$
$SY:RELEAS.SRT@<MM0:RELEAS.SRT$
$SY:CPU.SCP@<MM0:CPU.SCP$
$SY:VIRSTR.SCP@<MM0:VIRSTR.SCP$
$SY:ACCBLD.CTL@<SY:ACCBLD.CTL
$^C

READY

RUN $PIP
PIP      V06B-03 - RSTS V06B-02 ACCEPTNCE TEST
$SY:*TSORT.ADT@<MM0:*TSORT.ADT$/BL:512
$^C

READY

^C

READY

HELLO

RSTS V06B-02 ACCEPTNCE TEST JOB 2  [1,2]  KB0 03-DEC-76 01:14 PM
JOB 1 IS DETACHED UNDER THIS ACCOUNT
JOB NUMBER TO ATTACH TO? 1
ATTACHING TO JOB 1

ALL UFTP FILES FOR ACCOUNT [1,44] ARE LOADED

READY
```

JPS