RSTS/E Release Notes

Order No. AA-GK62A-TC

October 1985

These Release Notes describe new features of the RSTS/E system and explain the differences between this version and previous versions of RSTS/E. System managers and system maintainers should read this document prior to system installation.

OPERATING SYSTEM AND VERSION:	RSTS/E	V9.1
SOFTWARE VERSION:	RSTS/E	V9.1

digital equipment corporation, maynard, massachusetts

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

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

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

The following are trademarks of Digital Equipment Corporation:

digital™
DEC
DECmail
DECmate
DECnet
DECtape
DECUS
DECwriter
DIBOL
FMS-11
LA
MASSBUS
PDP
P/OS
Professional
Q-BUS
Rainbow
ReGIS
RSTS
RSX
RT
UNIBUS
VAX
VMS
VT
Work Processor

CONTENTS

Preface

The *RSTS/E V9.1 Release Notes* describe new features in RSTS/E for this release as well as differences between this and the previous release, RSTS/E V9.0.

In V9.0 of RSTS/E, the installation process was changed considerably from previous releases of RSTS/E. For RSTS/E V9.1, the installation and update procedure follows that same general process either to install a new RSTS/E V9.1 system, or to upgrade an existing V9.0 RSTS/E system to V9.1.

Because of the V9.0 changes to the installation process, users of previous versions of RSTS/E should read the *Installation and Update Guide* prior to attempting an installation of RSTS/E V9.1.

If you are installing a new RSTS/E V9.1 system, refer to PART I of the *System Installation and Update Guide*.

If you are upgrading a pre-V9.0 system to V9.1, refer to PART II of the *System Installation and Update Guide*.

If you are updating a RSTS/E V9.0 system to V9.1, refer to PART IV of the *System Installation and Update Guide*.

1 NEW FEATURES OF RSTS/E V9.1

1.1 New Device Support

This section describes some of the new devices which are supported in RSTS/E V9.1.

1.1.1 TK50 Cartridge Tape

RSTS/E V9.1 contains new device support for TMSCP (Tape Mass Storage Control Protocol) class tape drives. The TK50 is a Qbus cartridge tape subsystem with a storage capacity of approximately 95Mb of data per cartridge and follows the standard TMSCP protocol. The user interface to the TK50 tape subsystem is the same as with conventional 9track tape subsystems. All TMSCP class tape drives are accessed through the standard device designator of "MU". The TK50 will be made available as a Digital standard software distribution device.

The introduction of the TK50 will solve a long standing performance problem for RSTS/E Micro/PDP11 sites, namely backup performance. The streaming speed of the TK50 coupled with the added streaming tape support of RSTS/E V9.1 will allow Micro users to decrease the amount of time spent backing up their system files. Also, with the large capacity of the TK50, fewer volumes will be used in the backup procedure.

Since the TK50 is a single speed streaming tape drive, there is a noticeable change in performance when using a synchronous I/O package like SAVRES or PIP as opposed to a asynchronous I/O capable package like BACKUP. Synchronous I/O does not allow the type of command buffering available with asynchronous I/O, thus, causing extra repositioning by the tape drive. This means that the process of backing up your system disk, for instance, will be much quicker using the asynchronous BACKUP package rather than the synchronous SAVRES or PIP programs.

1.1.2 RD53 5-1/4 Inch Fixed Media Disk

The RD53 is the next generation of Winchester disk products offered for the Micro/PDP-11 systems. The RD53 is a 5-1/4 inch fixed disk with a storage capacity of approximately 71Mb of data.

1.2 Monitor

1.2.1 Use of PEEK in Future Releases of RSTS/E

The BASIC-PLUS PEEK() function (or .PEEK monitor directive) is intended for use only with software provided by DIGITAL.

The UU.SYS SYS() function, available since RSTS/E V7.0, which was released in September 1979, returns most of the useful job related data which the PEEK() function is commonly used to return.

DIGITAL strongly recommends that any program which uses the PEEK() function to obtain job related data be updated to use UU.SYS. Changes in a future release of RSTS/E will severely impact programs which do not follow this recommendation.

Every attempt will be made to avoid invalidating the PEEK() sequences which have been documented in the *RSTS/E Programming Manual*, although, as stated in the Programming Manual, DIGITAL has no commitment to do so.

If there is some piece of information which is not returned by UU.SYS which would be useful, please submit a "Suggestion - Priority 5" SPR explaining what data you would like returned, and why.

1.2.2 New .SPEC Function for Setting/Checking Density of Magnetic Tapes

A new .SPEC/MAGTAPE function has been added that allows you to set the density of a magnetic tape drive, check a density for legality, or return the current density of a drive. User programs wishing to use a tape density other than 800 bpi or 1600 bpi will require changes for this new .SPEC function. Please note that this call is the only supported means of setting or checking densities of the TK25 or TK50 magnetic tape drives. Since the TK25 and the TK50 are single density drives, this call can be used to just display the density of these drives. Refer to the *RSTS/E System Directives Manual* or the *RSTS/E Programming Manual* for a complete description.

1.3 File Processor

There are two new features in the file processor which are expected to increase system throughput significantly.

1.3.1 Multi-Streamed Window Turns

Previous to RSTS/E V9.1, window turns (the process whereby file block numbers are translated into disk block numbers) were handled by the RSTS/E file processor (FIP). In RSTS/E V9.1, this process has been removed from the RSTS/E file processor and scheduled as a separate, multi-streamed process in the monitor. Separating this process from FIP means that user level disk I/O does not require the assistance of the RSTS/E file processor to perform window turns. This feature also allows multiple window turns to be processed concurrently with each other (and other FIP requests).

The performance gain from this feature will be most noticeable on systems that allocate a large portion of memory to XBUF and have disks with moderately fragmented files. Small memory systems may, or may not, notice increased system performance depending upon the amount of XBUF available. Any additional XBUF used as a result of the use of this feature is automatic, and can be seen only by the SHOW CACHE command.

1.3.2 Resident SAT

Would be nice if we had memory!

The Resident SAT enhancement is an optional feature, selected by the user during timesharing, that will eliminate the need to perform disk I/O to obtain new disk space for files or directory structures. With the "resident SAT" feature, it is now possible to direct the file processor to keep the SATT.SYS file (a table of allocated or unallocated disk space) of each disk in memory. As a result, the file processor does not need to do disk I/O when trying to allocate new disk space or release old disk space. This will result in overall increased file processor performance.

The DCL commands LOAD/INDEX and UNLOAD/INDEX are used to load and unload the SAT of a disk, respectively. Refer to section 1.5.4 for further details on these new DCL commands.

1.3.3 Clustersize Optimization Fallback

RSTS/E V9.0 introduced the feature of clustersize optimization. This feature causes the file processor to calculate a file clustersize based on file size when creating a new file, for the purpose of reducing file processor overhead.

In some cases, when the free disk space on a disk was badly fragmented, this feature could cause programs to fail with the error "?No room for user on device", even though overall, there was sufficient free disk space.

For RSTS/E V9.1, if the file processor uses clustersize optimization and fails to allocate space, it will re-try the request using the pack clustersize, before returning an error to the user.

This change is transparent to user programs.

1.3.4 UU.CFG New Density Support

The UU.CFG .UUO subfunction now supports passing a system default magtape density value. This value may be in the range 2 to 32766. If a tape drive does not support the default density, the drive will be set to the closest legal density.

1.4 Terminal Service

1.4.1 Terminal Service changes

Enhancements have been made in terminal service to improve performance. These changes are transparent to user programs.

1.5 DCL

1.5.1 DCL Maintenance Enhancements

Several DCL commands were moved to an auxillary DCL command processor. The commands SORT, MERGE, LINK, MACRO, FORTRAN, COBOL, and DIBOL are now handled by $DCLAUX.TSK.

1.5.2 DCL Command File Format Change

Please be advised that in a future release of RSTS/E, DCL will enforce the documented suggestion of having all DCL command lines begin with dollar signs ($). Each DCL command line or label must have one, and only one, dollar sign and it must be in column one or an error will result. Column one is defined as the character following either a line feed or a form feed character.

This syntax is compatible with the current DCL syntax. DIGITAL recommends that you convert any command files that might be affected to avoid difficulties in a future release.

1.5.3 New DELETE/[NO]WARNINGS Qualifier

A new qualifier /[NO]WARNINGS has been added to the DELETE command. If you specify /NOWARNINGS, no error message will be printed if the file cannot be found.

The default for this qualifier is /WARNINGS.

1.5.4 New LOAD/INDEX and UNLOAD/INDEX Commands

A new feature in the RSTS/E monitor allows you to load and unload the Storage Allocation Table (SAT) of a disk into memory to decrease file processor waits and increase overall FIP performance.

The syntax of the new commands follows:

LOAD/INDEX [device-name[:]]

> This command loads a disk's Storage Allocation Table (SATT.SYS) file into XBUF, eliminating the need for the RSTS/E File Processor to update the SAT on disk whenever new disk space is allocated. If the device is not specified, the SAT from the system disk is loaded.
>
> The system will display the amount of XBUF that was used to load the SAT after it has been successfully loaded. The SHOW CACHE display shows the total amount of XBUF used to load the SAT from all disks that have their SAT file loaded in memory.

Privilege required: SWCTL

Command qualifiers: none

UNLOAD/INDEX [device-name[:]]

> This command unloads a disk's Storage Allocation Table (SATT.SYS) file that was loaded into memory with the LOAD/INDEX command.

Privilege required: SWCTL

Command qualifiers: none

1.6 Commonly Used System Programs (CUSPs)

1.6.1 BACKUP Enhancements

BACKUP now accepts the keywords MINIMUM and MAXIMUM for the /DENSITY qualifier and will also accept any legal density value (e.g. 1600, 6250 etc.).

In addition, BACKUP now prints status messages for each phase and writes a log of accounts written to the list file.

1.7 RT11 Utilities

1.7.1 PIP Enhancements for High Density Magnetic Tapes

The /DENSITY qualifier of PIP has been enhanced to accept 6250 as a legal density. In addition, the /DENSITY qualifier will accept the keywords MINIMUM and MAXIMUM, which will cause the magnetic tape drive to be set to its lowest or highest possible density during a /ZE (zero) operation.

1.7.2 SAVRES Enhancements for 6250 BPI Magnetic Tapes

The SAVRES utility (both offline and online) has been enhanced to support high density magnetic tapes. The /DENSITY: switch will now accept arguments of 6250, 6667 and 8000 in addition to 800 and 1600. As before, high density tape drives which also record at lower densities will be written to at the lowest density by default.

1.8 HELP File Additions

Several on-line help files have been changed for V9.1, to correct known errors or omissions, or document new commands or features in V9.1. The following areas are affected:

- o New /DENSITY arguments. Descriptions of those DCL tape commands or programs that support the /DENSITY qualifier have been changed to reflect the new argument keywords MAXIMUM and MINIMUM, as well as the new density values.

- o LOAD/INDEX and UNLOAD/INDEX commands. On-line help for these new commands are now provided.

- o READ command. A description of the /[NO]DELIMITER qualifier, omitted in V9.0, is now documented.

- SET ENTRY and SUBMIT commands. The descriptions of the /CPU_LIMIT and /TIME_LIMIT incorrectly showed these limits as measured in seconds. They now correctly show the limits measured in minutes.

- SUBMIT command. Descriptions of the qualifiers /LOG_QUEUE, /LOG_DELETE and /LOG_FILE, omitted in V9.0, are now provided.

- SHOW CACHE command. A new example has been included to show the new display format, and to include the new LOAD INDEX size field.

- SHOW DISK command. The example now includes the new LDX (Load Index) field in the disk comment field.

- DELETE command. A description of the /[NO]WARNINGS qualifier is now provided.

- TYPE command. Descriptions of the (now invalid) qualifiers /PROTECTION and /[NO]REPLACE have been removed, and a description of the /[NO]LOG qualifier has been added.

- CREATE/ACCOUNT command. A description of the /[NO]USER qualifier, omitted in V9.0, has been added.

2 Differences Between RSTS/E V9.1 and RSTS/E V9.0

2.1 System Generation (SYSGEN)

2.1.1 MSCP Controllers Question Changed

SYSGEN now prompts for the number of MSCP disks instead of the number of MSCP controllers. In previous releases, SYSGEN would only allow support for a maximum of 2 MSCP class controllers. With RSTS/E V9.1, any number (other than zero) specified at this question will build a monitor that supports configurations with up to 4 MSCP class controllers. Support for MSCP controllers is dynamically determined at system startup. However, the number of units specified for this question determines the total number of units that can be supported across all controllers. In response, type in the number of MSCP class disks that you have (or will have) on your target configuration. The maximum number of MSCP disks that you can have has been raised to 16 in V9.1. As always, RSTS/E will round this number up to a multiple of 4 units.

2.1.2 New question for TMSCP Class Tape Support

RSTS/E V9.1 includes support for the first TMSCP class tape drive, the TK50. There can be up to two TMSCP class tape controllers per system. Type in the number of TMSCP class tapes that you have (or will have) on your target configuration.

2.1.3 Directory Caching Question Removed

All systems will now receive the "data and directory cacher".

The data and directory cacher supports all of the features of the directory cacher, with one exception: it will not use small buffers for caching. Systems which used the directory cacher may find that they need to increase XBUF allocation if they were relying on caching using small buffers.

2.1.4 Usage of Template Monitor has Changed

When using a template monitor, SYSGEN will take all of its defaults from the template monitor, instead of from the monitor and hardware that is currently on the system. The user will be prompted for any new devices (TMSCP tape drives in V9.1).

2.1.5 SYSGEN will now Calculate the Number of RK05's

SYSGEN now calculates the exact number of RK05's configured on a system. Prior to V9.1, if there was an RK05 controller on the system, the default would be 8. Now, the default will be the number of units present.

2.1.6 RK05's no Longer Accepted as the Only Disk

SYSGEN no longer accepts a monitor configuration that only has RK05's for disks. In V9.0, the RK05 could not be the system disk, but SYSGEN would still allow a configuration that just had RK05 disks. This has been corrected in V9.1.

2.1.7 Comments are now Allowed with SYSGEN answers

Comments, beginning with the exclamation point character, can now be added to SYSGEN answers. Use this feature to document the answers for the INSTAL.LOG file.

2.2 Initialization Code (INIT.SYS)

2.2.1 DEFALT Option Changes (Primary Run-Time system)

The concept of a primary run-time system has been removed from RSTS/E V9.1. This will result in a 1K memory saving.

The DEFALT option of INIT.SYS will no longer prompt for primary RTS. In addition, the RTS memory table suboption has been removed.

If you have batch jobs that depend on the RSX.RTS keyboard monitor, you should tailor your start-up command files to install the RSX run-time system.

2.2.2 COPY Option Changes

The COPY option of INIT has been modified to eliminate extraneous text that was printed during an automatic update. When COPY is entered automatically during the course of an update or creation of a new system disk, the message "Copying required system files" will be displayed. Unlike V9.0, a message will not be printed for each file copied; instead, the operation will complete silently, and the new system disk will be booted.

2.2.3 New Bootstrap Format for Magnetic Tapes

The format of the bootstrap for magnetic tapes has been changed for V9.1. Instead of two different bootstrap files (MTBOOT.SYS and MSBOOT.SYS), there is now only one file, called MGBOOT.SYS, which is used on all magnetic tapes regardless of density or drive type. In addition, bootable magnetic tapes now have a non-standard format DOS label record as the first record on the tape; instead of the normal length of 14 bytes, the first label record is 2062 bytes in length. The PIP, DIRECT and DCLUTL utilities have been changed to recognize the new format, and these tapes will be treated just like any other DOS-format tape. Note also that HOOK and SAVRES will write the new-format bootstrap on all bootable magnetic tapes.

Because of the non-standard label length, a new-format bootable tape cannot be copied unless the program copying the tape is capable of reading and writing 2062-byte records. If the COPY program is used to copy a new-format bootable tape, the qualifier /BL:4096 must be specified.

2.3 Monitor

2.3.1 Monitor Size

The memory resident size of a V9.1 monitor will be slightly larger than V9.0. This increase will be less than 1K words. However, since there is no longer a primary run-time system, occupying at least 1K words of memory, it is expected that a V9.1 monitor will be the same size, or slightly smaller than, a V9.0 monitor.

2.3.2 Problem in Crash Dump Module Fixed

Previously, the data around the PC, as reported by ANALYS, was sometimes incorrect. This problem has been fixed.

2.4 File Processor

2.4.1 Quota Checking Problem Corrected

A problem in the logout code, which would incorrectly calculate the number of logged in users for quota checking, has been corrected. This problem occured when jobs logged out on pseudo keyboards with only one user logged in on a physical keyboard.

2.5 Terminal Service

2.5.1 Enhancement to the Auto-Baud Feature

The restriction, preventing a change to the current speed setting of a terminal line set to auto-baud, has been removed for RSTS/E V9.1. In V9.1, you can now change the current speed setting of a line set to auto-baud provided that the device is owned. Changing the permanent speed is still *not* permitted for terminal lines set to auto-baud.

2.5.2 Lost Output

A problem in Terminal Service where output could be lost if echo control was not sysgened into the monitor has been fixed.

2.5.3 XON/XOFF Handling and No-Stall Reads

In V9.0 Terminal Service, XON characters were not transmitted after no-stall reads. This problem has been corrected.

2.6 Drivers

2.6.1 Dismounting MSCP Class Drives

When MSCP class disk drives are logically dismounted with the DCL DISMOUNT command, the MSCP driver will make that drive available to other MSCP controllers ported to the drive from other systems. Upon issuing the DISMOUNT command, the (A/B) port light will go out, signaling that the drive is no longer virtually connected to that controller and available for other controllers to gain access to that drive.

This change lifts a restriction with dual-ported MSCP drives across multiple systems that required manual intervention in order to release the drive from the controller.

2.7 DCL

2.7.1 DCL Commands Changed to Support the new Density Format

The DCL commands, INITIALIZE, MOUNT, BACKUP, and SET SYSTEM, now accept any number between 2 and 32766 for the density parameter. These DCL commands will also accept the keywords "MINIMUM" and "MAXIMUM". The density supplied for the INITIALIZE, and MOUNT commands has to be a density supported by the tape drive, otherwise an error will result. For the SET SYSTEM command, any value can be specified for the system default. When a tape drive is initialized or mounted without a density being specified, the system will choose the drive's nearest legal density that is not greater then the system default. If the system default is less than the drive's lowest density, then the density is set to the drives lowest density.

2.7.2 New SHOW CACHE Display

The SHOW CACHE command now displays the total amount of XBUF used to load the SATT.SYS files from all disks that have their SAT loaded in memory.

2.7.3 SHOW TERMINAL change

Users without HWCFG privilege can now display their terminal's permanent characteristics by specifying the /PERMANENT qualifier.

2.7.4 SET JOB qualifiers now Have Defaults

The /PRIORITY and /RUNBURST qualifiers for SET JOB can now be specified without an argument. The /PRIORITY qualifier will default to -8 and the /RUNBURST qualifier will default to 6.

2.7.5 The TYPE command no longer prints Trailing Nulls

The TYPE command will not print nulls when the last block of the file is only partially full. This will increase performance, especially on slow speed terminals.

2.7.6 The CREATE command now Prints a Prompt

When the CREATE command is used to create a file, it will now print a prompt when it is ready to accept input from the keyboard.

2.7.7 Protection Code Range Checking Added

Protection code range checking (0 to 255) has been added to the COPY, CREATE, and RENAME commands.

2.7.8 Cluster Size Range Checking Added

Cluster size range checking (-256 to +256) has been added to the /CLUSTERSIZE qualifier of the COPY and CREATE commands. A cluster size of zero is allowed and, when specified, means that RSTS/E will use the optimal cluster size for the size of the file. The sign of the cluster size is ignored. DCL will always use the pack cluster size if the specified cluster size is less than the pack clustersize. The default is /CLUSTERSIZE=0.

2.7.9 High Density Support for INITIALIZE, MOUNT and SET SYSTEM Commands

The INITIALIZE, MOUNT and SET SYSTEM commands have been enhanced for high density magnetic tape support. [redacted]nt. In addition, the /DENSITY qualifier will accept the keywords MINIMUM and MAXIMUM, which will cause the tape drive to be set to its lowest or highest possible density during an INITIALIZE or MOUNT operation. Refer to the *RSTS/E System Manager's Guide* for more details.

2.7.10 DCL Corrections

Several corrections were made to DCL. The following problems were corrected:

- o The rightmost conflicting qualifier would sometimes cause other non-conflicting qualifiers on the line to be ignored. Now only the conflicting qualifiers are ignored.

- o If $LOGIN.TSK was missing or non-runnable, DCL would loop with "??Program failure in DCL" when you tried to log in. Now DCL will print "?Can't find file or account" and exit leaving you logged out.

- o If a system logical such as SORT$ is not assigned DCL would issue an "?Invalid device" message. Now DCL issues "?Command not available".

- o F$PARSE("",,"FLAGS") returned -1 indicating an illegal file specification. F$PARSE now returns zero.

- o If you issued two consecutive commands which were not installed, DCL would fail with "??Program failure in DCL". This problem has been corrected.

2.8 DCL Command File Processor

2.8.1 Delimiters and Command File Procedures

The *only* line delimiters recognized by the command procedure processing within the monitor are line feed and form feed. All other standard delimiters are treated as data. See section 1.5.2 for further command file information.

2.9 Commonly Used System Programs (CUSPs)

2.9.1 New Density Formats Supported by the COPY and BUILD Programs

The COPY and BUILD programs have been changed to support the keywords MINIMUM and MAXIMUM for the /DENSITY qualifier along with any density the tape drive supports.

2.9.2 Corrections to BACKUP.TSK

Several corrections were made to BACKUP.TSK. The following problems were corrected:

- o BACKUP would occasionally write invalid multi-volume save sets to devices with fewer than 400 free blocks.

- o The BACKUP/QUERY option asked for each selected file twice.

2.9.3 Corrections to BUILD.TSK for $PROMPT RTS command

Previously, the default runtime system displayed for a $PROMPT RTS prompt was the primary runtime system, if that system's executable file type was .TSK or .BAC.

Since RSTS/E V9.1 eliminates the need for a primary runtime system, BUILD now uses as a $PROMPT RTS default the first runtime system encountered in the monitor RTS list that has an executable file type of .TSK or .BAC.

2.9.4 Corrections to MAKSIL

MAKSIL no longer prints "%Unexpected type during $$ALVC phase of symbol resolution" when processing input files containing data auto-load sections.

2.10 Print/Batch Services

2.10.1 Corrections to PBSMGR.TSK

Several corrections were made to PBSMGR.TSK, the user interface portion of PBS. The following problems were corrected:

- The SHOW SERVER command did not display the status "Marked for deletion" for servers that were deleted while still active.
- In some cases, SHOW SERVER error messages displayed KB-type servers using "controller" format (e.g., KBG11:) instead of the standard format.
- The SUBMIT command erroneously accepted /NOAFTER (only the SET ENTRY command allows the NO prefix with the AFTER qualifier).

2.10.2 Corrections to PBS.TSK

Several corrections were made to PBS.TSK. The following problems were corrected:

- Batch commands files with a /CPU_LIMIT of greater than 110 minutes were terminated before the time expired.
- /CPU_LIMIT and /TIME_LIMIT counters were reset upon the execution

of the $LOGIN command within a batch .COM file.

- o The print server incorrectly calculated horizontal position when processing hardware tab characters which caused /TRUNCATE to fail on print devices that supported the hardware tab character. It should be noted that /TRUNCATE may not perform truncation as expected on printers that select to use hardware tabs with tab stops set at other than eight character intervals (since the print server does not read the device's hardware tab stops).

- o The print server incorrectly printed header pages on any form less than 42 lines long.

- o The print server displayed a KB type print device in "controller" (e.g. KBG11:) on header pages.

- o The print server incorrectly printed sequential non-stream format files with no carriage control and no embedded carriage controls.

- o The print server did not honor FORTRAN carriage control on sequential stream files.

- o The print server sometimes printed the ER$LCH (logical channel too large) error message when PBS was moderately busy.

- o User Request Packet (URP) function code 15. (/LOG_QUEUE_FLAG) did not perform as specified.

- o /AFTER jobs did not always start at the proper time if PBS was idle for 18 hours.

2.11 Account Management Utility

2.11.1 ACTMGR Corrections

Several corrections were made to ACTMGR.TSK.

- o In V9.0, using a template account that had a project number greater that 127 yielded incorrect results. This is fixed in V9.1.

- o The privilege check to determine if WACNT or GACNT privilege was needed did not work properly in all cases. This is fixed for V9.1.

- o The message "Account <account> not deleted" has been changed to a warning message.

2.12 TERMGR

2.12.1 TERMGR Corrections

Several corrections were made to TERMGR.TSK. The following problems were corrected:

- If there were any continuation lines on the macros in the $TERMAC.SYS and $TERDFL.SYS files, TERMGR would prompt the user for the next line instead of getting it from the file. This is fixed in V9.1.

- There was a problem with the error handling when setting the speed on terminals set to Autobaud. In V9.1, the appropriate error message will be printed when trying to set the permanent speed on Autobaud terminals and when setting the current speed for logged-out Autobaud terminals.

- The /[NO]INTERACTIVE qualifier and the capability qualifiers (/ANSI, /ADVANCED_VIDEO,...) would be ignored when used with the /DEVICE_TYPE qualifier to set the characteristics of a user-defined macro. This is fixed in V9.1.

- In V9.0, TERMGR would not make the distinction between a terminal that was disabled and a terminal that was in use. In V9.1, the appropriate error message will be printed.

- The /INQUIRE command would not work for LA120 and LA12 devices. This is fixed in V9.1.

2.13 System Management Utility

2.13.1 UTLMGR Corrections

Several corrections were made to UTLMGR.TSK. The following problems were corrected:

- The SHOW FILE/SYSTEM/SWAP_FILE display now displays the correct swap max.

- Previously, when executing the INSTALL/RUNTIMESYSTEM and INSTALL/LIBRARY commands, the wrong error message was displayed if insufficient memory was available for the run-time system or the library. On V9.1, the appropriate error message is displayed.

- If a non-existent job was specified for the SHOW JOB/PRIVILEGES command, the error would not be trapped cleanly. Now, an appropriate error message is displayed.

- The command SHOW LOGICAL <logical-name>/SYSTEM will now look up the logical name in the system-wide table instead of the user table.

- Previously, when executing a FORCE command, the message "KB busy" was printed if the destination terminal was set to NOBROADCAST, even though the FORCE would succeed. This is changed in V9.1 so that the BROADCAST setting of the terminal is not checked during execution of a FORCE command.

- The display of the SHOW DEVICES command would not distinguish between a disabled and a non-existent disk. This is fixed in V9.1.

2.14 RT11 Utilities

2.14.1 PIP corrections

- When outputting an ASCII file to a terminal, PIP will no longer print nulls when the last block is only partially full.

- PIP now displays a prompt when creating a file from the user's keyboard.

- When CTRL/C is typed during a copy to a disk file, the output file will be deleted. This ensures that there is no partially-filled file that someone could mistake for a complete file.

- When the default protection code is set, there was a problem mounting the second volume of a multi-volume ANSI tape set. This is fixed in V9.1.

- When appending files from tape to disk, ?Odd address trap errors could result. This is fixed in V9.1.

- The :NOWARN keyword argument to the /RE switch (Rename) will suppress the "?Can't find file or account" error along with the "?Name or account exists" error.

- Before V9.1, PIP would only output a CR/LF terminator for implied <cr> files if the last character of each record was not a terminator. This goes against the meaning of implied <cr> files. In V9.1, every record of an implied <cr> file will be terminated by a CR/LF terminator.

2.14.2 SAVRES Restrictions

If the IDentify option of SAVRES is used on high density magnetic tape saveset volumes, the density of the volume will be incorrectly reported as 800 BPI. The density reported in the dismount messages during SAves and REstores is correct.

Due to a change in RSTS/E V9.0, which allowed non-file-structured write access to the system disk with sufficient privilege, SAVRES will now erroneously allow a SAve or IMage copy of the system disk. This activity should be avoided, since the resulting saveset or copy will be invalid.

These problems will be fixed in a future release of RSTS/E.

2.15 Distribution Media

2.15.1 New TK50 Distribution Media

Beginning with RSTS/E V9.1, the RSTS/E Library and Installation media will be distributed on TK50 Cartridge Tapes. See section 1.1.1 for more information on TK50 Cartridge Tape support.

RSTS/E V9.1 is distributed on the following media:

9-Track Magnetic Tape - 800 BPI (TU10/TE10/TU16/TE16/TU45/TU77)
9-Track Magnetic Tape - 1600 BPI (TS11/TU80/TE16/TU77)
TK50 Cartridge Tape
RK07 Disk Cartridge
RL02 Disk Cartridge

Please refer to Section 3.1 of these Release Notes for more information on the distribution kits.

2.16 Optional Software

Refer to the RSTS/E Optional Software Cross Reference Table of the SPD for a complete list of optional layered products that are supported on RSTS/E V9.1.

2.16.1 IBM Communication Products

This section explains how V9.0 has affected these products.

2.16.1.1 RJ2780

Installation of RJ2780 V3.0 changed slightly for RSTS/E V9.0. To install RJ2780 V3.0, you must first ensure that RJ2780.BAS program is in account [1,2], then enter the BASIC-PLUS run-time system, and run UPDATE$:BUILD using the command file PATCH$:PA2502.CMD. For example:

```
$SW BASIC

Ready

RUN UPDATE$:BUILD
BUILD  V9.0-14  RSTS/E V9.0-14 Released V9.0

Input device <_SY:> ? <CR>
Output device <_SY0:> ? <CR>
Control File is ? PATCH$:PA2502.CMD
```

This will apply the necessary patches to the RJ2780.BAS program, and install it into account [1,2]. The RJ2780 program can not interact with the new Print/Batch Services (PBS). If spooled operation is required OPSER/QUEMAN must still be used. Note, as in previous releases, RJ2780 will not be automatically shutdown by the OPSER program, this must be done manually. If RJ2780 is installed as a privileged program (protection code <232>), any user will be able to use the program, altering their priority to 0, and enabling spooled mode (detaching the program). Note however, that the detached and total job quota of that account will still be enforced by the LOGIN program. If you want the program to run detached, and still be logged into that account, then the job quotas for that account must be set properly. If RJ2780 is made non-privileged, then HWCTL, TUNE, and RDMEM are required. DEVICE privilege is also required if the RJ: device is set as restricted. The detach job quota will be honored by RJ2780 in this case, as will the Send/Receive quotas.

2.16.1.2 HPE 2780/3780 and 3271 PE

These products function the same as in previous releases. For spooled operation, the OPSER/QUEMAN package must be used, since the PBS spooler can not interact with the IBM communication products. The detach and total job quotas must also be set properly to allow detached operation concurrent with other jobs in the same account. In addition, the Send/Receive quotas will be required for spooled operation.

2.17 Support for RSTS/E V9.0

RSTS/E V9.0 will be supported for 6 months after the release of RSTS/E V9.1.

3 RSTS/E V9.1 Distribution Kits

3.1 Distribution Kit Order Numbers

This chapter provides you with the most current order numbers for the RSTS/E distribution kits. For the most current order numbers of other optional software kit(s) see the bill of materials. The *RSTS/E System Installation and Update Guide* tells you to check these order numbers prior to beginning the actual installation.

DIGITAL sends you distribution kits on one of the following:

- 9-track magnetic tape (800bpi)
- 9-track magnetic tape (1600bpi)
- TK50 Cartridge Tape
- RL02 disk cartridge
- RK07 disk cartridge

3.1.1 RSTS/E V9.1 Distribution Kit

The RSTS/E V9.1 distribution kit consists of the tapes or disks that contain the programs and files you need to perform an installation. This section lists the order numbers for the RSTS/E V9.1 distribution kits. You should compare the numbers on the tapes or disks that make up your kit to the numbers shown here, when you are instructed to do so in the *RSTS/E System Installation and Update Guide*.

DEC Order Number	Pack ID	Label Title
9-Track Magnetic Tape (800 bpi)		
AP-2753K-BC	SYS090	RSTS/E V9 LIBRARY 8MT9
AP-C883G-BC	UPD091	RSTS/E V9.1 INSTALLATION 8MT9
9-Track Magnetic Tape (1600 bpi)		
BB-H751M-BC	SYS090	RSTS/E V9 LIBRARY 16MT9
BB-P016I-BC	UPD091	RSTS/E V9.1 INSTALLATION 16MT9
TK50 Cartridge Tape		
AQ-FW57A-BC	SYS090	RSTS/E V9 LIBRARY TK50
AQ-FW58A-BC	UPD091	RSTS/E V9.1 INSTALLATION TK50
RK07 Disk Cartridge		
AY-D526M-BC	SYS090	RSTS/E V9 LIBRARY RK7

AY-P019I-BC	UPD091	RSTS/E V9.1	INSTALLATION RK7
RL02 Disk Cartridge			
BC-J026G-BC	SYS090	RSTS/E V9	LIBRARY RL2 1/2
BC-FR62A-BC	BACKUP	RSTS/E V9	LIBRARY RL2 2/2
BC-P015I-BC	UPD091	RSTS/E V9.1	INSTALLATION RL2

3.1.2 RSTS/E V9 Installation Kits

You will receive this kit approximately three times a year, and the kits are labeled 9.2, 9.3, 9.4 and so forth. Because each kit is cumulative, use the most current one. For example, version 9.3 contains all of the patches and replacement modules that were on 9.1 and 9.2 in addition to some new patches and replacement modules.

3.1.3 UNSUPP Package Components

The following is a list of components and executable tasks which are provided in the UNSUPP (unsupported software) package:

Component	Description
MONEY .BAS	Program to display accounting data
MONEY .TSK	Task file for MONEY.BAS
DSKDMP.BAS	Program for producing formatted dumps of files
DSKDMP.TSK	Task file for DSKDMP.BAS
DSKDMP.HLP	Help file for DSKDMP.BAS
STATUS.B2S	Monitor statistics program
QSTATS.B2S	Monitor statistics program
LQSTAT.BAS	Monitor statistics program
STATUS.TSK	Task file for STATUS.B2S
QSTATS.TSK	Task file for QSTATS.B2S
LQSTAT.TSK	Task file for LQSTAT.BAS
DSKPEK.OBJ	Subroutine for monitor statistics programs
QSTATS.DOC	Documentation file for QSTATS.B2S program
MEMORY.BAS	Program to display current items in memory
MEMORY.TSK	Task file for easy installation of MEMORY program
FLOAT .BAS	Program to determine floating device addresses
FLOAT .TSK	Task file for easy installation of FLOAT program
DIALER.BAS	Auto dialer for some VADIC type hardware

DIALER.DOC	Documentation for using DIALER.BAS and CONNEC.BAS
DIALER.TSK	Task file for DIALER.BAS
CONNEC.BAS	Program for use with DIALER
CONNEC.TSK	Task file for CONNEC.BAS
EMTCPY.BAS	Sample program for use with the EMT logger
EMTCPY.TSK	Task file for EMTCPY.BAS
RSTSFL.OBJ	VMS program for reading RSTS/E disks
RSTSFL.OLB	Library used to build RSTSFL on VMS
RSTSFL.DOC	Documentation for VMS RSTSFLX program
RSTSFL.HLP	Help file for VMS RSTSFLX program
B2CREF.TSK	Basic-Plus (+2 also) cross reference generator
B2CREF.HLP	Help file for B2CREF.TSK
B2CREF.DOC	Documentation for B2CREF.TSK
BPDA .BAS	Basic-Plus DUMP analysis program
BPDA .TSK	Task file for BPDA.BAS
RNO .TSK	RSX RNO program
RNO .RNO	RNO documentation in RNO format
RNOBLD.CMD	TKB command file to build RNO
RNOBLD.ODL	TKB ODL file to build RNO
RNPRE .MAC	RNO source file
COMND .MAC	" " "
CMTAB .MAC	" " "
ERMSG .MAC	" " "
FMTCM .MAC	" " "
HYPHEN.MAC	" " "
INDEX .MAC	" " "
PINDX .MAC	" " "
RNCMD .MAC	" " "
RNFIO .MAC	" " "
RNORSX.MAC	" " "
RUNOFF.MAC	" " "
START .MAC	" " "
DUMP .SAV	RT11 program for producing formatted dumps of files
SRCCOM.SAV	RT11 program for doing file comparisons

These files are transferred to your system by specifying the package name UNSUPP in your list of packages to install. The UNSUPP package is included if you specify "ALL."

3.1.4 RSTSFLX

RSTSFLX had several bugs when transferring files from newer disk types and also files that were larger than approximately 34000 blocks. These problems have been fixed in V9.1.

4 Additional System Installation-Related Programs

This section contains an updated version of Appendix E in the *RSTS/E System Installation and Update Guide*. All references to the PBUILD program have been deleted because PBUILD is no longer included in the RSTS/E installation kit.

This section contains programs you may need to use during your installation or update. It also contains programs related to patching RSTS/E code. The BUILD program performs system installation-related tasks. The ONLPAT program allows you to patch binary code. The CPATCH program patches ASCII source code. The DSKCVT program converts a disk to RDS1.2 format.

4.1 Using the BUILD Program

The BUILD program builds new software and updates existing software. The /PATCH qualifier of the BUILD program automatically installs updates as you build new software. (If there are no update files for the software you build, then BUILD does not attempt the update procedure.)

BUILD is a privileged program that is installed with a protection code of <232>. To use the various BUILD functions, you need all privileges except:

- SYSMOD
- EXQTA
- USER1-8

The BUILD program requires that the patches and replacement modules be on disk. If your media is tape, you must first use the layered product installation procedures. See Part III of the *RSTS/E System Installation and Update Guide* for more information.

To run BUILD, type:

```
$ RUN UPDATE$:BUILD
```

BUILD then asks the following questions:

```
BUILD  V9.0 RSTS V9.0
Input device <SY:> ?
Output device <SY0:> ?
Control File is ?
```

After you enter a correct control file name, BUILD prints a message to indicate it is copying the specified control file to the public disk structure. The variable [nnn,nnn] represents the control file account number, and the nn in the PBLDnn.TMP specification is the job number. The format of the message is:

```
*** COPYING FILE dev:[nnn,nnn]controlfile TO SY:PBLDnn.TMP ***
```

The next question asks you to select the function you want BUILD to perform.

You can elect to just build, just patch, or to build and patch a program.

```
Function (Build/Patch, Patch, Build) <Build/Patch> ?
```

If you select a function that includes patching, BUILD then asks you for the location of the patch file.

```
Patch file input location <PATCH$:>
```

BUILD then preprocesses the control file. Normally at this point, during the preprocessing of the first control file, BUILD prints a run-time system question that asks you for the name of the run-time system under which you want the executable programs to run. The format of the question is:

```
Run-Time System <xxxxxx>?
```

BUILD prints (in angle brackets) the name of the first installed run-time system that has an executable file type of .TSK or .BAC. Press the LINE FEED key to accept the default, or enter the name of another run-time system. If you decide not to accept the default, note that the run-time system you specify must include a keyboard monitor and have a default executable file type of .BAC or .TSK. If BUILD does not accept your response, it prints an error message:

- Xxxxxx is not a keyboard monitor
- Run-time system must have a default extension of .BAC or .TSK
- Xxxxxx is not installed
  Attempt to Install xxxxxx <Yes or No>?

After BUILD prints either of the first two messages, it prints the RUN-TIME SYSTEM question again. Enter a new run-time system name that conforms to the run-time system restrictions indicated in the error message. If BUILD prints the xxxxxx is not installed message, it has found the specified run-time system in account [0,1] but has discovered the run-time system is not installed. After printing the message, BUILD asks if you want to install the run-time system:

```
Attempt to Install xxxxxx<No>?
```

BUILD inserts the run-time system name in the xxxxxx field, which represents the run-time system that BUILD will attempt to install. Type NO if you want BUILD to return to the RUN-TIME SYSTEM prompt. One of the following cases occurs if you type YES:

- o BUILD installs the run-time system successfully and prints the message, xxxxxx Installed. BUILD asks the CSPCOM question if you selected RSX or BP2COM as your run-time system.

- o BUILD installs the run-time system, determines whether it includes a keyboard monitor and has the correct file type, and then prints one of the following two error messages:

  - Xxxxxx is not a keyboard monitor

  - Run-time system must have a default extension of .BAC or .TSK.

  After printing either of these messages, BUILD tells you it is removing the previously installed run-time system by printing the xxxxxx Will be Removed message. The program returns to the RUN-TIME SYSTEM question.

- o BUILD may detect errors other than those already described. When it does, it prints:

  ```
  Run-time system operation failed<text>
  ```

  The text can be any valid RSTS/E error message. After printing the error, BUILD returns to the RUN-TIME SYSTEM prompt.

If you selected either the BP2COM or RSX run-time system, BUILD asks the following CSPCOM question:

```
Use CSPCOM<Yes>?
```

Type YES if your run-time system is RSX; type NO if it is BP2COM.

After the preprocessing of the first control file, BUILD may prompt you for the name of another control file by printing:

```
Additional Control File is <None>?
```

Type another control file specification and terminate the response by pressing the LINE FEED key. Like the previous control file procedure, BUILD prints the Copying File message and then preprocesses the file. BUILD continues to print the Additional Control File is question until the default is selected. Press the LINE FEED key to accept the default. BUILD prints the current date and time and begins executing the control files. If you did not attach a hyphen to the original control file specification, BUILD may not ask for additional control files. Therefore, if you need to process more than one control file, attach a hyphen to the logfile=cotrolfile[-] specification to ensure BUILD asks the Additional Control File question. Note that BUILD prints the Copying File message and preprocesses the control files; it does not execute any control file until you specify all files.

During the preprocessing phase, BUILD may ask additional dialogue questions. These questions cannot be listed here; they are initiated by the particular control file that BUILD is processing. There are many control files that might be processed, each with its own unique set of questions. You must therefore see documentation regarding the particular control file to answer these questions.

After BUILD finishes preprocessing the control files, BUILD then runs the ATPK program. When ATPK finishes, it displays two messages.

```
!*** BUILD Complete ***

!*** Processing ended on 31-May-85 at 11:13 AM ***
```

During most installations, BUILD asks additional questions that are specific to the individual layered product. See the installation guide for the individual product for guidelines on answering specific questions.

You can create a log file of the build procedure and also specify that the build is to run detached, in the format shown after the Input device? prompt in the BUILD dialogue:

```
logfile=dev:/det
```

For example:

```
Input device <SY:>: ? <file name>=MM0:/DET
```

Note that the Input device? prompt also accepts a /DOS or /ANSI qualifier.

4.1.1 Error Messages

This section describes the errors that can occur when running the

BUILD program. Errors can occur in BUILD when:

- o You enter information required to run the program but the information is incorrect. In this case, BUILD displays the incorrect input with a text error message and prints the prompt requesting the input again.

- o You enter an incorrect command in a control file. Errors occurring in BUILD while it is running from the control file are always fatal. BUILD prints the error message and returns control to your keyboard monitor.

Table 1 lists the BUILD error messages.

Table 1: BUILD Error Messages

Message and Meaning
?XXXXXX PRIVILEGE REQUIRED
You do not have the listed privilege or privileges needed to run the program.
?FILE NAME ERROR
This error indicates that a bad file name was found in the control file currently executing.
?BAD FILE NAME: <filename>
Filename does not follow proper syntax rules. This error can occur during the dialogue.
?<file string> NOT PERMITTED
File string is valid but contains a feature that is not permitted for the operation, such as wild card characters, or a file name when only a device and PPN was expected.
FILE NOT FOUND: <filename>
The syntax of the file name is correct, but refers to a nonexistent file.
?<file string> ERR=<error number>
The file string is invalid for a reason other than the previous three error messages. The number returned by the message represents the error number produced by the

BASIC-PLUS run-time system.

?CAN'T COPY <file1> TO <file2>
<BASIC-PLUS error message>

Occurs after the message !COPYING TEMP FILE <file1> TO <file2>. BUILD was attempting to copy the source file from the current account to the permanent account. The copy failed for the reason given in the BASIC-PLUS error message. The BASIC-PLUS errors that are most likely are ?No room for user on device and ?Device not available. Others are various magnetic tape errors or hardware I/O errors.

?FATAL ERROR NO. <error number> ON LINE <error line> LEAVING PROGRAM

The program was terminated by an unexpected error, which may indicate a system error or a hardware failure. The error returns the BASIC-PLUS error number and line number.

4.2 Using the ONLPAT Program to Patch RSTS/E Binary Code

The need for manual file patching has greatly decreased. However, you may still have to use ONLPAT to make emergency corrections.

The ONLPAT.SAV program allows you to patch binary code. You can run ONLPAT in either the keyboard mode or the command file mode. In the keyboard mode, you enter patch information on your terminal. In the command file mode, the patch information is read from a command file.

4.2.1 Using ONLPAT in Keyboard and Command File Mode

ONLPAT gives you a choice. You can make patches to the software by entering the corrections manually at your keyboard or you can use a previously created command file to apply the patches for you. You make the choice at the beginning of the ONLPAT dialogue when the program asks for the name of the command file you want to use (if one exists). Your response determines how ONLPAT gets the patch information.

If you enter a command file name, ONLPAT uses the command file to make the corrections. When you press the RETURN or LINE FEED key, ONLPAT assumes the patch information comes from the keyboard. ONLPAT expects keyboard input when you enter a file name followed by an equal sign (=) symbol. This indicates you want ONLPAT to create a log file of the patch. The remainder of this section describes how to patch software using either of these methods.

When you enter patches at the keyboard, follow procedures similar to those you use for the PATCH option of INIT.SYS. Unlike the PATCH option, ONLPAT does not install a patch until you enter a circumflex/C (^C) at the end of the last line in the patch. Do not press CTRL/C unless you want to abort the patching operation. Because the program waits while you enter all the patch information, you can review the entire patch and make changes before installing the patch.

The ONLPAT program can apply patches to the monitor or to programs written in MACRO, such as PIP.SAV. To install a published patch, you must first look in the *RSTS/E Maintenance Notebook* or the *RSTS/E Software Dispatch* for the article containing the patch you want to install. After reading the article thoroughly, you can create a command file or install the patch manually.

An ONLPAT example follows:

```
RUN $ONLPAT
Command file name? <RET>
File to patch? [0,1]BASIC.RTS
File found in account [0,1]
Base address? ..XTN
Offset address? 0
 Base   Offset  Old     New?
??????  000000  020000  ? 0
??????  000002  ??????  ? ^C
```

Table 2 lists and describes the types of responses ONLPAT accepts in the program dialogue.

Table 2: Responses to ONLPAT Questions

Response	Meaning
Number	Enter an octal number from 0 to 177777; leading zeros are optional. Or, enter a decimal number in the range 0. to 65535., making sure to include a trailing decimal point to distinguish it from an octal number. The octal or decimal number you enter becomes the new contents of the current location.
Symbol	Substitute a global symbol name for an octal number as the new contents of the current location. The symbol table for the module being patched is part of the SIL file. It contains that

	module's global symbol names and their values. A global symbol name must be one- to six-alphanumeric characters and must be defined in the symbol table for the current module.
Expression	Use an expression as a substitute for an octal number. An expression consists of one or more numbers or global symbols, separated by arithmetic operators (+,-, and *). You can use parentheses to group portions of an expression.
RETURN or LINE FEED key	Advance to the next location without altering the contents of the current location.
circumflex	Return to the previous location without altering the contents of the current location.
CTRL/Z	Enter CTRL/Z to return to the previous question.
CTRL/C	Enter CTRL/C to abort all patching and return to the program prompt.

Table 3 describes the questions asked by ONLPAT.

Table 3: ONLPAT Dialogue Questions

Question and Response
COMMAND FILE NAME? o Enter a command file name to have ONLPAT install patches using commands in the command file. You can have ONLPAT make a record of the patches as well as install them if you include a log file name with the command file name in the command line: Command file name? log file = command file name If you do not include a file type for the command file, ONLPAT selects .CMD. The default file type for the log file is .LOG. o Press the RETURN or LINE FEED key if you do not have a command file to apply the patches. ONLPAT assumes you want to enter the patches manually from the keyboard. If you want a record of the patches that you apply manually from the keyboard, include only the log file name in the command

line:

```
Command file name? log file =
```

You must place an equal sign after the log file name to make sure ONLPAT does not interpret the file you specify as being a command file. Again, ONLPAT uses .LOG as the file type if you do not specify one with the file name.

FILE TO PATCH?

Press the RETURN or LINE FEED key if you are applying a patch to the installed monitor SIL. However, another response may be necessary. If so, enter the response suggested either by the RSTS/E Maintenance Notebook or the RSTS/E Software Dispatch article.

MODULE NAME?

Enter the module name included with the patch described either in the RSTS/E Maintenance Notebook or the RSTS/E Software Dispatch article.

BASE ADDRESS?

Type the base address included with the patch described either in the RSTS/E Maintenance Notebook or in the RSTS/E Software Dispatch article.

OFFSET ADDRESS?

Type the offset address included with the patch described either in the RSTS/E Maintenance Notebook article or the RSTS/E Software Dispatch article. The offset address is the first location to be patched relative to the specified base. The response suggested in the patch article will be a number, symbol, or expression as described in Table 2.

Use the information in Table 3 and the following procedures to apply patches manually:

1. Gather the articles that contain the patches you need. You can find the articles either in the *RSTS/E Maintenance Notebook* or the *RSTS/E Software Dispatch*.

2. Run the ONLPAT program by typing RUN $ONLPAT. The program begins by displaying the first of a set of dialogue questions:

```
Command file name?
```

At this point you:

1. Press the RETURN key to enter the patch manually from your terminal.

2. Type the name of a file, followed by an equal sign (=) symbol. This instructs ONLPAT to create a log file of the patch. The patch is also displayed on your terminal.

3. You then answer the remaining ONLPAT dialogue questions, following the instructions in the *RSTS/E Maintenance Notebook* or the *RSTS/E Software Dispatch*. Remember that ONLPAT does not install the patch until you type ^C (not CTRL/C) to end the program. Typing ^C is entirely different than entering CTRL/C: ^C ends the patch successfully; CTRL/C stops execution of ONLPAT without making any patches.

4. After installing the patch, ONLPAT displays a message informing you that the patch has been installed:

```
PATCH COMPLETE
1 PATCH INSTALLED
```

ONLPAT then returns to the command file question. You can then either exit the program or enter another patch.

To install a patch with a command file:

1. Create a command file. See the section, Building ONLPAT Command Files.

2. Run the ONLPAT program by typing RUN $ONLPAT, and then press the RETURN key. The program displays a question asking for the name of the command file that contains the patches. Enter a command file in the following format:

   ```
   log file=command file
   ```

   If you want a record of the patch, type the name of a log file. The default file type for a log file is .LOG. ONLPAT does not create a log file by default.

3. Type the name of the file that contains the commands ONLPAT uses to install the patch. The default file type for a command file is .CMD. ONLPAT then executes the command file. If you did not include in the command file the name of the

file you want to patch, ONLPAT pauses, letting you enter the file name.

4. When ONLPAT finishes processing, the program displays a summary of the patches installed and any patches skipped, and then returns to your keyboard monitor.

4.2.2 Patching a Running Monitor with ONLPAT

Usually you can patch the installed monitor Save Image Library (SIL). The monitor is modified on disk, and only after you shut down, and then restart the system, do the patches become effective. However, patches to monitor overlay code (OVR) are a special case. If you use patched code from OVR along with an unpatched monitor in memory, your system may crash, or not work properly. This is because monitor overlays are read in from the monitor SIL (from disk) as they are needed, and changes to the monitor do not take place until the system is brought down and then restarted.

When ONLPAT detects that the file being patched is the installed monitor SIL, the program imposes certain restrictions:

- Installs the patch if the patch does not involve OVR. That is, ONLPAT applies the patch to the monitor image on disk. The patch takes affect only after you shut down the system and then restart it.

- If only one block of OVR is involved in the patch, ONLPAT installs the patch. Because the patch affects only one block, ONLPAT can make all monitor modifications with a single disk write.

- If the patch involves more than one OVR block, ONLPAT does not modify the monitor, and displays the following error message:

  ```
  ?Patch to module ovr too long for installed sil
  ```

- If the patch involves both OVR and some other module of the monitor, ONLPAT displays the following error message:

  ```
  ?Patch to installed sil spans ovr and other module(s)
  ```

  The way to install a patch of this type is to copy your installed SIL, patch the copy, shut down the system, start the patched copy, and delete the old unpatched SIL.

4.2.3 Building ONLPAT Command Files

Create ONLPAT command files as follows:

1. Run ONLPAT from any logged-in terminal by typing RUN $ONLPAT. The program then displays the COMMAND FILE NAME? question.

2. Respond to the command file question in the following format:

   Patchfile = KB:

   First type the file name of the patch file you want to create and then KB: to indicate you are creating the patch file from the keyboard.

3. Press the RETURN key after you finish entering the command string. ONLPAT then asks:

   FILE TO PATCH?

   Type the name of the file you want to patch, press the RETURN key, and enter the text of the patch described in the *RSTS/E Maintenance Notebook* or the *RSTS/E Software Dispatch* article.

After creating the command file, you may want to modify it. If modifications are necessary, apply the following rules:

- End each individual command in the patch by pressing RETURN.

- When ONLPAT executes the command file, the program changes all lowercase characters to uppercase and reduces all tabs and spaces to a single space.

- Blanks are significant. Use them exactly as shown for individual command lines.

- You can include comment lines at the beginning of a patch. A comment line must begin with an exclamation mark (!).

- You must include a line that specifies the file you want to patch. The format of this line is:

  FILE TO PATCH? filename.type

  Type the name of the file that you want ONLPAT to patch. If you do not include a file name, ONLPAT prompts you for the file name during program execution. ONLPAT uses the file name to specify any subsequent patches in the command file that do not specify the file name.

- Subsequent command lines in a patch are identical to the formats used in the *RSTS/E Software Dispatch* or the *RSTS/E Maintenance Notebook*. The following command line is optional, but must appear with spacing exactly as shown in the *RSTS/E Software Dispatch* or the *RSTS/E Maintenance Notebook* articles:

  ```
  BASE OFFSET OLD NEW?
  ```

  Also, do not use any parenthetical comments.

- You must use the following notation:

  ^Z   is a circumflex, then Z; or CTRL/Z

  ^C   is a circumflex, then C; not CTRL/C

  <LF>   is the LINE FEED key

4.3 Using the CPATCH Program to Patch ASCII Source Code

With the advent of prebuilt tasks and other replacement modules, the need for manual patching has greatly decreased.

Use the CPATCH program to install patches to the source code of the BASIC-PLUS library programs. CPATCH can patch individual programs, as described in this section or build patching command files (see the section, Building CPATCH Command Files).

4.3.1 General Operating Procedures

The general procedures to patch a BASIC-PLUS source file are:

1. Run the program by typing RUN UPDATE$:CPATCH. After printing a header line, CPATCH prints:

   ```
   FILE TO PATCH-
   ```

2. Respond in the format:

   ```
   out-filespec=in-filespec
   ```

   Enter the file name for the unpatched source file on the input side of the command line and any file specification you want on the output side. The default file type for both files is .BAK. If you do not include an output file, CPATCH assumes the file has the same name as the input file and gives it a .BAK file type.

3. Press the RETURN key in response to the number sign (#) prompt. The prompt printed by CPATCH indicates that it is ready to patch the source program.

4. Edit the source program, using the instructions in the *RSTS/E Maintenance Notebook* or the *RSTS/E Software Dispatch*. CPATCH prints an * prompt to show when to begin editing the source.

5. Type EXIT in response to the * prompt when you finish editing. CPATCH prints a message indicating the patch is installed:

   ```
   PATCH FROM file specification COMPLETE.
   ```

   CPATCH returns to the # prompt. If there are other patches to the current file, do them now, or enter CTRL/Z. If you enter CTRL/Z, CPATCH returns to the prompt:

   ```
   FILE TO PATCH-
   ```

6. Return to Step 2 in this procedure if there are other files to patch.

7. Enter CTRL/Z to exit the program if there are no more files to patch.

8. Follow instructions in either the *RSTS/E Maintenance Notebook* or the *RSTS/E Software Dispatch* for including the patched source program in the system library.

4.3.2 Building CPATCH Command Files

This section describes how to create new files used by CPATCH to perform the actual edit of your source programs.

4.3.3 Building the Patching Command File

To make a new patch, you must create a CPATCH command file by running the CPATCH program. Enter the commands at your terminal and use a file as the log. The output log becomes the new CPATCH command file.

4.3.3.1 File Naming Convention

Patch files are much easier to manage if you adopt a consistent file naming convention. The convention used for all patches to the Commonly Used System Program (CUSP) Library is presented here as an example.

A typical example might be the program QUMRUN. Assume the first article in the *RSTS/E Software Dispatch* concerning QUMRUN describes a patch. The article's sequence number is 14.6.1, where the number 14 is the component number (Spooler and Operator Services Package), 6 is the subcomponent number, and 1 indicates it is the first article published about QUMRUN. The corresponding patch file name would be PA1406.001.

4.3.3.2 Creating the CPATCH Command File

Run the CPATCH program to create the CPATCH command file. The file for which the patch file is being created is specified at the FILE TO PATCH prompt, and the keyboard is specified as the command file with the CPATCH command file being created as the log file. The patch is then manually applied. Note, however, that CPATCH is a line editor which means its text buffer contains only a single line of text at any one time. When the patch is completed, the log file contains a CPATCH command file that performs the same patch when applied using CPATCH. The commands to create the CPATCH command file to perform the first patch to QUMRUN is an example.

You first start the program by typing:

```
RUN UPDATE$:CPATCH
```

After printing a header line, CPATCH prints:

```
FILE TO PATCH-
```

The response to this prompt would be:

```
out:QUMRUN.BAS=in:QUMRUN.BAS
```

You place on the input side of the command line the device name and PPN that contains the unpatched source file QUMRUN.BAS. On the output side, specify any device and PPN. Use a different device name and PPN from the one used for the input specification. Then you can test the patched program before replacing the original source.

CPATCH prints a # prompt to which the correct response for the example patch would be:

```
PA1406.001=
```

With PA1406.001 on the output side of the command line, CPATCH uses it as the log file. There is no specification on the input side, thus, CPATCH uses the terminal (KB:) as the default command file. Note, however, that you may want to check or determine the patch checksum by including the /CS switch (see the section Verifying the Patch).

CPATCH prints the * prompt again. At this point, use CPATCH editing commands to perform the patch on the file. The following sections introduce you to the editing capabilities of the CPATCH program.

4.3.4 Editing with CPATCH

After CPATCH prints the * prompt, you are ready to use the CPATCH line editor to perform the patch. When the patch is complete, use the EX command to return to the # prompt. A CTRL/Z returns the FILE TO PATCH prompt, and another CTRL/Z exits the program. The file PA1406.001 (from the example in the previous section) then becomes the CPATCH command file required to perform the patch.

The CPATCH editor is a character-oriented text editing program written in BASIC-PLUS for use on the RSTS/E operating system. It reads ASCII files from any input device, makes specified changes, and writes on any output device. You operate the editor through the use of commands typed at your terminal. The basic editing process can be divided into three parts:

1. Reading the input text into an internal buffer

2. Changing the text stored in the buffer

3. Transferring the revised text to a new file

The following sections list terms and definitions and describe the commands used with the CPATCH editor.

4.3.4.1 CPATCH Editor Terms and Definitions

The CPATCH editor enables you to perform editing operations on ASCII text. The program stores the text to be edited in an intermediate area of memory called a buffer. It does not alter text in the input files.

The editor refers to text in the buffer by using a character location pointer called a cursor. The cursor is considered to reside between any two characters. At the start of editing operations, the cursor precedes the first character in the buffer. You move the cursor during editing operations according to the type of editing being performed. You can refer to text in the buffer as so many characters or lines preceding or following the cursor.

To edit text, you must specify a command or series of commands in response to the * prompt printed by the CPATCH program. The commands are classed according to the type of operation they perform:

Cursor Manipulation — Moves the cursor without altering text within the buffer.

Character Search — Finds a specified occurrence of text within the buffer to facilitate editing.

Character Manipulation — Adds to and removes characters or a line of text from the buffer.

Many editor commands are character-oriented. That is, they affect a specified number of characters preceding or following the cursor. The argument of these commands specifies the number of characters in the buffer on which to operate. The number of characters you specify with the argument n is the same forward (n) as backward (-n). The LF, CR, and NUL characters, although not printed, are embedded in text lines, counted as characters in character-oriented commands, and treated as any other text characters.

Some commands are line-oriented. The argument of these commands specifies the number of lines on which to operate. Because the editor counts the line-terminating characters to determine the number of lines on which to operate, an argument n does not affect the same number of lines forward (positive number n) as it affects backward (negative number -n).

For example, the argument -1 applies to the line beginning with the first character following the second previous end-of-line and ending with the character preceding the cursor. The argument 1 in a line-oriented command, however, applies to the text beginning with the first character following the cursor and ending at the first end-of-line. Thus, if the cursor is at the center of the line, the argument -1 affects one and one-half lines backwards from the cursor and the argument 1 affects one-half line beyond the cursor.

Character search and manipulation commands operate in command mode. The text used to search or being manipulated can be a single character or a group of characters (called a string). If the text does not contain a CR or LF character and is small enough to fit on a single typed line, command mode can be used to specify the text. In command mode, the editor expects the first character following a search or manipulation command to be a delimiting character for the desired text. The editor uses as the text the characters between delimiters. The delimiting character, therefore, cannot appear in the text, nor can the characters CR and LF.

To specify the word INPUT as the text in command mode, you can use the following string with the search or manipulation commands:

```
/INPUT/
```

The editor uses the characters between the slash delimiters (/) as the text. In command mode, you can use any printable character that does not appear in the text as a delimiter. If the delimiting character appears in the text, the editor attempts to interpret the remaining characters as commands.

To prevent ambiguities when a file is used for command input, CPATCH requires all "invisible" characters, (ESCAPE, LINE FEED, and so forth) to be translated to a visible form. This translation is always done on log file output to permit log files to be used for command input with no modification. Because the output is used for patch verification, it must be typed exactly. An error causes a rejection of the patch.

The translation is:

Command File Format	Control Characters
<TAB>	ASCII 9 (horizontal tab)
<FF>	ASCII 12 (form feed)
<ESC>	ASCII 27 (escape)
<LF>	ASCII 10, 13, 0 (line feed sequence)
<CR>	ASCII 13, 10 (carriage return sequence)
<null>	ASCII 0 (isolated null character)
<13>	ASCII 13 (isolated carriage return sequence)
<10>	ASCII 10 (isolated line feed character)

4.3.5 CPATCH Editor Commands

The commands you can use with the CPATCH editor are:

Advance Command (nA)

The Advance command is a line-oriented command that moves the cursor to a point preceding the first character of a line. depending on the size of the argument used. If you do not include an argument, the editor selects one line forward as the default. That is, the A command moves the cursor to the beginning of the next line. Arguments and their effect on the Advance command are listed as follows:

nA Advances the cursor n lines and positions the cursor at the beginning of the line.

0A Moves the cursor to the beginning of the current line.

Change Command (nC/xxxx/)

The Change command changes a specified number of characters following the location of the cursor. The text </xxxx/> must be set off by delimiters. The C command is equivalent to an Insert command followed by a Delete command. Arguments and their effect on the C command are described as follows:

nC Replaces n characters following the cursor with the specified text. The cursor is placed after the inserted text. The C command does not affect text beyond the current line.

-nC Replaces n characters preceding the cursor with the specified text. The cursor is placed after the inserted text. The C command allows you to make changes to the current line only.

0C Replaces the current line up to the cursor with the specified text.

Delete Command (nD)

The Delete command is a character-oriented command that deletes n characters in the page buffer beginning at the cursor. If you do not include a value for n, the editor deletes the character immediately following the cursor. After executing the command, the editor places the cursor at the first character following the deleted text. The following list describes each argument and its effect on the Delete command:

nD Deletes n characters following the cursor. The D command does not allow you to delete text beyond the current line.

-nD Deletes n characters preceding the cursor. You cannot use the D command to delete text other than on the current line.

0D Deletes the current line up to the cursor. The editor positions the cursor at the first character following the deleted text.

Get Command (nG/xxxx/)

The Get command is a search command that allows you to search for the nth occurrence of the specified text (/xxxx/) starting at the current cursor location. If you do not include a value for n, the editor searches for the first occurrence of the text. The search ends when the editor either finds the nth occurrence or encounters the end of the buffer. If the search is successful, the editor places the cursor at the end of the searched text. The editor prints the G command followed by a question mark (?) to indicate an unsuccessful search. In that case, the cursor follows the last character in the buffer.

Whole Command (nH)

The Whole command reads each page of the primary input file into the buffer until the nth occurrence of the specified text object is found. The editor begins at the cursor and searches the current buffer until the nth occurrence of the text is found or the end of the buffer is reached. If the search is successful, the editor positions the cursor immediately following it. If the editor does not find the nth occurrence in the current buffer, it writes the buffer to the primary output file, clears the buffer, reads the next page of the primary input file into the buffer, and continues the search. The search is unsuccessful when the nth occurrence is not found and the end of the primary input file is reached. The H command followed by a question mark indicates an unsuccessful search. When this occurs, the editor copies the entire contents of the primary input file to the primary output file and places the cursor at the beginning of an empty buffer. The H command operates in a forward direction only; negative arguments are not allowed.

Insert Command (I/xxxx/)

The Insert command allows you to insert text (/xxxx/). The editor inserts the text and places the cursor after the last character of the inserted text. You cannot use an argument with the Insert command. Up to 80 characters per line can be specified by typing the letter I on one line followed by the RETURN key and the text to be inserted on the following line(s). Execution of the command occurs when you press the LINE FEED key. If the text does not contain a carriage return or a line feed character, it can be typed on the same line as the I command but must be set off by delimiters.

Jump Command (nJ)

The Jump command moves the cursor over a specified number of character locations. If you do not include an argument, the J command moves the cursor one character position forward. Arguments and their effects upon the Jump command are described as follows:

nJ	Moves the cursor forward n characters. The J command does not affect any text beyond the current line; that is, a command such as 400J only moves the cursor to the end of the current line.

-nJ	Moves the cursor backward n characters. The J command does not affect any text beyond the current line; that is, you cannot move the cursor further than the beginning of the
current
line.

0J	Moves the cursor to the beginning of the current line.

Kill Command (nK)

The Kill command removes n lines of text (including the carriage return and line feed characters) from the page buffer beginning at the cursor and ending with the nth end-of-line. The editor places the cursor at the beginning of the line following the deleted text. The following list describes each argument and its effect on the Kill command:

nK	Removes the character string (including the CR and LF sequence) beginning at the cursor and ending at the nth end-of-line.

0K	Removes the current line up to the cursor location.

List Command (nL)

The List command prints at your terminal lines of text as they appear in the buffer. An argument preceding the L command indicates the portion of the text to print. For example, the command 2L prints on your terminal the text beginning at the cursor and ending with the second end-of-line character. The cursor is unaffected by the L command. Arguments and their effect on the List command are described as follows:

nL	Prints one line of text at the terminal. The L command accepts only zero and one as arguments.

0L	Prints the current line up to the cursor.

Verify Command (V)

The Verify command (V) prints at your terminal the entire line in which the cursor is located. It provides an easy means of determining the location of the cursor after a search is completed and before any editing commands are given. You can also use the V command after you have typed an editing command to allow you to check your input. The V command does not take arguments.

EX Command (EX)

To end an editing session, you use the EX command. It writes the buffer to the primary output file, transfers the remainder of the primary input file to the output file, closes all open files, and renames the temporary file as the edited, primary output file. The following dialogue shows the procedure:

```
*EX

#^Z
```

The editor prints the # prompt (in response to which you can enter more input and output specifications). Entering CTRL/Z ends the editing session and control returns to your keyboard monitor.

You should periodically use the EX command to end the editing session. The primary input file remains intact while you are editing. The temporary file retains the revised text as edits are made. If the system crashes during the editing session, the primary input file is not disturbed, but the edits you are making are lost. Therefore, it is recommended that you frequently exit to update the primary output file. In the event of a system crash, the amount of editing lost is limited to those edits made since the beginning of the latest session.

4.3.6 Verifying the Patch

The CPATCH program provides two switches to verify the accuracy of the patching operation. To make sure CPATCH applies a patch correctly, you must specify the /CS switch while creating a patch file and the /CS:n switch when you are applying the patch.

To create a patch file, run CPATCH and answer the FILE TO PATCH prompt. After you enter the name of the file to patch and press the RETURN key, CPATCH prints a # prompt. Because you are creating a file to store the patches, enter its file specification followed by an equal sign, attach the /CS switch, and press the RETURN key. CPATCH prints an * prompt to indicate its readiness to accept the patch.

After you enter the patch and terminate your input by typing EX, CPATCH calculates a number, called a checksum, and prints it on your terminal. CPATCH uses this number while applying the patch to verify the patch was correctly incorporated. CPATCH prints the patch complete message to mark the end of the operation and returns to the # prompt. You can then enter the name of a new patch file or enter CTRL/Z to return to the FILE TO PATCH prompt. If you also enter CTRL/Z in response to the FILE TO PATCH prompt, CPATCH terminates and returns control to your keyboard monitor. The following terminal output illustrates the use of the /CS switch and the creation of a checksum number:

```
RUN UPDATE$:CPATCH
<CPATCH's header line>
File to patch - PATCHA.BAS=PATCHA.BAS
#PA2325.001=/CS
*G/-10/-1D
*EX
Checksum = 53437
Patch from _KB:[1,247]CPATCH.CMD complete.
#^Z
File to patch - ^Z
```

After creating the patch file, you can use CPATCH again to apply the patch. As before, run CPATCH, specify the file to patch, and press the RETURN key to receive the # prompt. In response, attach the /CS:n switch, replacing n with the checksum number that CPATCH calculated previously (53437). Press the RETURN key, enter the patch, and exit by typing EX. If you enter the patch correctly, CPATCH calculates the same checksum number, applies the patch, and prints the patch complete message. Otherwise, it prints an error message:

```
?Actual checksum of 53437 does not match 53537
%Patch from _KB:[1,247]CPATCH.CMD skipped.
```

You must retrace your steps and try again. An example of a successful attempt to apply a patch follows:

```
RUN UPDATE$:CPATCH
<CPATCH's header line>
File to patch - PATCHA.BAS=PATCHA.BAS
#PA2325.001/CS:53437
*G/-10/-1D
*EX
Patch from _SY:[1,247]PA2325.001 complete.
#^Z
File to patch - ^Z
```

The /CS:n switch, n representing a number from 0 to 65535, causes CPATCH to recalculate the checksum as a result of your new input and to compare it to the original calculation. In this case, you entered the patch accurately; CPATCH was able to calculate the same checksum and then apply the patch.

4.3.7 Building the BUILD Control File

After building a command file of patches for CPATCH, you may want to build a control file that contains the instructions to BUILD for installing the patches.

The BUILD control file must reference the CPATCH file created in the section "Creating the CPATCH Command File." You can create a new BUILD file with the required BUILD commands, or you can insert the commands in an existing file. In either case, use an editor to edit the file.

In the example, the CPATCH file for the first patch to QUMRUN was created. Now the commands to reference this file are inserted in the BUILD file (either new or existing). The commands required are:

```
$PATCH QUMRUN
PA1406.001
$END
```

Because this is the first patch to QUMRUN, all three of the commands must be added. If there were already existing patches to QUMRUN, you must add the reference to the new patch before the $END statement of any previous patches to QUMRUN. For example, if the second article concerning QUMRUN is also a patch article, the CPATCH command file for that patch would be PA1406.002. A patch already exists for this program (PA1406.001), therefore you insert PA1406.002 immediately before the $END statement in the sequence of commands that apply the previous patch. The resulting command sequence is:

```
$PATCH QUMRUN
PA1406.001
PA1406.002
$END
```

These commands cause PA1406.001 to be applied to QUMRUN. The file PA1406.002 would be applied to the patched file resulting from first applying PA1406.001.

If you want to include checksums, add the /CS:n switch after each patch file reference. For example:

```
$PATCH QUMRUN
PA1406.001/CS:57315
PA1406.002/CS:12345
$END
```

4.3.8 Error Messages

This section describes the errors that can occur when running the CPATCH program. Errors can occur in CPATCH when CPATCH, which is run by BUILD, attempts to apply an invalid CPATCH command file as a patch to a program. The program will continue to run.

CPATCH error messages appear at the user's terminal. If they occur while running under BUILD, BUILD continues to run. If they occur running CPATCH standalone, they may cause CPATCH to terminate execution. Table 4 lists the CPATCH error messages.

Table 4: CPATCH Error Messages

Message and Meaning

?MISMATCH

Occurs when editing a version of the file that was not correct, for example, attempting to patch a file that was already patched. When this happens, the patch is ignored, and the file is restored to the state before attempting to apply the patch. The log file (or KB: if that is the log) shows the cause of the mismatch by printing:

```
 =cmd=> string
 =log=> string!
 ?mismatch?
```

String is the contents of the command file that is the correct program response to the previous command in the command file; string! is the actual program response produced. It can be a verification message on a line that has been modified or a CPATCH error message, such as ?Srch fail when a string that was expected has been deleted.

This is a nonfatal error when running under BUILD. BUILD continues to run normally but the source file used for any commands to follow will be the original, unmodified source.

?SRCH FAIL

The string searched for with the G command was not found.

?EOF

If the error occurs in response to the H command, it indicates that the string was not found, and the source input file has been read to the end of file. If it occurs while running from a command file, it indicates that an end of file occurred in the command file before an EX command was encountered.

?BAD ARG FOR <command>

The argument provided with command is invalid for that command. For example, the L command only permits 0 and 1 as valid arguments.

?BAD COM: <command>

Command is not a valid command string.

?CAN'T BE THE SAME <log>=<command>

Unless the log file and the command file are both the keyboard, they must be different files. If they are not, this message is printed:

```
FATAL ERROR NO. <err> ON
LINE <error line>
LEAVING PROGRAM
```

The same as in BUILD. However, err=10 is possible in normal operation. That error number indicates a protection violation, which can occur because CPATCH does not have to be run in a privileged account.

?FATAL ERROR NO.<error number>ON LINE <error line> IN AUTOED

The same as above, however, the error occurred in the program AUTOED which is chained to by CPATCH. This error does not terminate CPATCH, however. It has the same effect as a ?Mismatch error, but it may still indicate the need to send DIGITAL an SPR.

?Please 'RUN CPATCH'

An attempt was made to run AUTOED as a stand-alone program.

?<file string> NOT PERMITTED
?BAD FILE NAME: <file string>
?<file string> ERR=<error number>
?FILE NOT FOUND
?CAN'T COPY <file1> TO <file2>

These messages have the same meaning as they do for BUILD: see Table 1.

4.4 Using the DSKCVT Program

The DSKCVT (Disk Convert) utility converts V8.0 and older disks to the new V9.0 RDS1.2 format. Specifically, DSKCVT performs the following functions:

- Convert RDS0.0 disks to RDS1.2 format

- Convert RDS1.1 disks to RDS1.2 format

- Convert accounts with "LOOKUP" passwords to "NOLOOKUP" at any time, on any RDS1.2 disk (optional)

To run DSKCVT, you must start with a disk that does not need rebuilding. (Use the MOUNT command if necessary to rebuild the disk first.) As a precaution, DIGITAL recommends that you back up any disk before conversion.

DSKCVT requires some work space if the pack being converted is RDS0.0 format. Therefore, you must make sure that there are 32 blocks times the number of different group numbers used on the pack, in order to allow conversion. (For example, if there are ten group numbers, then the pack should have 320 blocks of free space for conversion.)

DSKCVT accepts either logically mounted or logically dismounted packs. However, the pack should not be in use when you run DSKCVT. To verify that no one is using the pack, DSKCVT dismounts and remounts it at the start of processing.

The following is an example of the DSKCVT program:

```
$ RUN $DSKCVT
DSKCVT  V9.0  RSTS  V9.0  SYSTEM

Disk to convert? DM0:
Is this a System Disk <NO>? YES
Generate NOLOOKUP passwords <YES>? <RET>
Display accounts <YES>? <RET>

 ** CURRENT DISK ATTRIBUTES **

      Disk:            _DM0:
      Label:           SYSDSK
      Level:           1.1

Proceed (Y or N)? Y

Converting _DM0:SYSDSK from level 1.1 to 1.2 ...
[0,1] Converted
```

```
[1,1] Converted
[1,2] Converted
      .
      .
      .

_DM0: conversion to 1.2 completed 31-May-85 11:05 AM

Converting _DM0: passwords to NOLOOKUP
[0,1] Converted
[1,1] Converted
[1,2] Converted
      .
      .
      .

_DM0: Password conversion completed 31-May-85 11:06 AM
```

In the example, DM0: is the disk pack to convert to RDS1.2. When you type DM0: in response to the prompt, DSKCVT displays the pack ID DM0:SYSDSK. This lets you verify that you are converting the proper pack. If the pack was already converted, DSKCVT displays an error message and then repeats the Disk to convert: prompt. Otherwise, DSKCVT continues with the dialogue.

When you answer Y to the Proceed (Y or N)? prompt, DSKCVT begins the conversion. DSKCVT processes the accounts in ascending order. When the pack is converted, DSKCVT dismounts it and updates the pack label. The pack is now in RDS1.2 format. However, the storage allocation table (SAT) does not yet reflect the free disk space on the pack. Therefore, the pack is marked "dirty" until DSKCVT updates the SAT. Then the pack label is updated to show that the pack is now rebuilt. If the pack was mounted when DSKCVT started, it will be mounted once again, with the pack ID as its logical name.

DSKCVT will not lose data even if the program aborts because of lack of work space, a system crash, or a disk hardware problem. If there is a problem during the conversion, you need only correct the problem and rerun the conversion. In such a case, it is not necessary to restore the pack from a backup pack.

If the conversion proceeded far enough that DSKCVT was able to convert the pack, it is then necessary only to mount it. Otherwise, DSKCVT restarts the conversion from the beginning, after deleting the work files that were left behind from the first attempt at conversion.

When you convert an RDS0.0 (pre-V8.0) disk, DSKCVT makes two passes. In the first pass, DSKCVT converts the RDS0.0 pack to RDS1.1. The second pass converts the RDS1.1 pack to RDS1.2.

In converting RDS1.1 disks to RDS1.2 format, DSKCVT adds several new attribute blocks to each account. Since some of the new attributes are only relevant on system disks, DSKCVT asks you whether the disk being converted is a system disk. DSKCVT adds the following new attribute blocks:

- o Disk quota/usage block
- o Privilege block, for system disks
- o Non-disk quota block, for system disks

In creating the privilege attribute block, DSKCVT uses the following rules:

- o For account [1,1], it assigns all privileges, except EXQTA
- o For all other [1,*] accounts, it assigns all privileges, except SYSMOD and EXQTA
- o For accounts not in the [1,*] group, it assigns no privileges.

This scheme assigns privileges that are compatible with previous versions of RSTS/E.

HOW TO ORDER ADDITIONAL DOCUMENTATION

DIRECT TELEPHONE ORDERS

In Continental USA
and Puerto Rico
call **800–258–1710**

In Canada
call **800–267–6146**

In New Hampshire,
Alaska or Hawaii
call **603–884–6660**

DIRECT MAIL ORDERS (U.S. and Puerto Rico*)

DIGITAL EQUIPMENT CORPORATION
P.O. Box CS2008
Nashua, New Hampshire 03061

DIRECT MAIL ORDERS (Canada)

DIGITAL EQUIPMENT OF CANADA LTD.
940 Belfast Road
Ottawa, Ontario, Canada K1G 4C2
Attn: P&SG Business Manager

INTERNATIONAL

DIGITAL EQUIPMENT CORPORATION
P&SG Business Manager
c/o Digital's local subsidiary
or approved distributor

Internal orders should be placed through the Software Distribution Center (SDC), Digital Equipment Corporation, Northboro, Massachusetts 01532

*Any prepaid order from Puerto Rico must be placed
with the Local Digital Subsidiary:
809–754–7575

RSTS/E
Release Notes
AA-GK62A-TC

Reader's Comments

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

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

____

____

____

____

____

____

Did you find errors in this manual? If so, specify the error and the page number. ______________________________

____

____

____

____

____

____

Please indicate the type of user/reader that you most nearly represent.

□ Assembly language programmer
□ Higher-level language programmer
□ Occasional programmer (experienced)
□ User with little programming experience
□ Student programmer
□ Other (please specify) ______________________________

Name ______________________________ Date ______________________________

Organization ______________________________

Street ______________________________

City ______________________________ State ______________________________ Zip Code or Country ____________

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

digital

No Postage
Necessary
if Mailed in the
United States

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

POSTAGE WILL BE PAID BY ADDRESSEE

ATTN: Office Systems Documentation MK01-2/E02
DIGITAL EQUIPMENT CORPORATION
Continental Boulevard
Merrimack N.H. 03054

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

Cut Along Dotted Line