ANL-80-52

ANL-80-52

TEXT PROCESSING FOR TECHNICAL REPORTS

(Direct Computer-assisted Origination, Editing, and Output of Text)

by

A. De Volpi, M. R. Fenrick, G. S. Stanford,
C. L. Fink, and E. A. Rhodes

ARGONNE NATIONAL LABORATORY, ARGONNE, ILLINOIS

Prepared for the U. S. DEPARTMENT OF ENERGY
under Contract W-31-109-Eng-38

DISCLAIMER

Printed in the United States of America
Available from
National Technical Information Service
U. S. Department of Commerce
5285 Port Royal Road
Springfield, VA 22161

NTIS price codes
Printed copy: A04
Microfiche copy: A01

Distribution Category:
Mathematics and Computers
(UC-32)

ANL-80-52

ARGONNE NATIONAL LABORATORY
9700 South Cass Avenue
Argonne, Illinois 60439

TEXT PROCESSING FOR TECHNICAL REPORTS

(Direct Computer-assisted Origination, Editing, and Output of Text)

by

A. De Volpi, M. R. Fenrick,* G. S. Stanford, C. L. Fink, and E. A. Rhodes

Reactor Analysis and Safety Division

October 1980

*Science Applications, Inc., Schaumburg, Ill. Now at Digital Equipment Corporation, Brookfield, Wis. 53005.

THIS PAGE

WAS INTENTIONALLY

LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

TEXT PROCESSING FOR TECHNICAL REPORTS

by

A. De Volpi, M. R. Fenrick, G. S. Stanford, C. L. Fink, and E. A. Rhodes

ABSTRACT

Documentation often is a primary residual of research and development. Because of this important role and because of the large amount of time consumed in generating technical reports, particularly those containing formulas and graphics, we have adapted an existing data-processing computer system so as to provide text-processing of technical documents. Emphasis has been on accuracy, turnaround time, and time savings for staff and secretaries, for the types of reports normally produced in the reactor development program.

The computer-assisted text-processing system, called TXT, has been implemented to benefit primarily the originator of technical reports. The system is of particular value to professional staff, such as scientists and engineers, who have responsibility for generating much correspondence or lengthy, complex reports or manuscripts--especially if prompt turnaround and high accuracy are required. It can produce text that contains special Greek or mathematical symbols.

Written in FORTRAN and MACRO, the program TXT operates on a PDP-11 minicomputer under the RSX-11M multitask multiuser monitor. Peripheral hardware includes videoterminals, electrostatic printers, and magnetic disks. Either data- or word-processing tasks may be performed at the terminals.

The repertoire of operations has been restricted so as to minimize user training and memory burden. Secretarial staff may be readily trained to make corrections from annotated copy. Some examples of camera-ready copy are provided.

INTRODUCTION

A primary product of research and development is a report; a frequently used tool of technical managers is documentation to specify, communicate, monitor, and control technical activities. It has long been recognized that substantial time consumption and time delay are associated with the necessary documentation. The reactor development program, in common with other disciplined technical activities, has had to adhere to certain reporting standards. At the same time the scientist or engineer involved in the technical activity must concentrate on the substantive experimental or analytical work in order to be able to meet milestones. These goals sometimes clash.

Professional staff are frequently responsible for initiating, generating, and issuing large quantities of paperwork. Traditionally, secretarial assistance is provided to type, correct, and distribute the documentation. Until the introduction of magnetic tape and card-memory typewriter systems, processing of documentary material was potentially error-prone, inefficient, and tardy.

Through the use of modern computer-related devices, significant improvements have been made worldwide in "word processing" to permit more accurate and efficient turnover of paperwork in the office. However, few of these improvements have directly benefited professionals who originate manuscripts; most modernization has been in favor of the officeworker or typesetter. Most authors must still initiate text, correct copy, and arrange details of composition through cumbersome and error-prone processes.

We have developed computer-assisted writing and editing system specifically to benefit professional staff--in particular, technical staff. It is intended to allow easy entry of original text, to simplify and shorten the process of text editing, and to expedite distribution. It is intentionally less sophisticated than programs that lead to general-purpose, publication-quality camera-ready output [1], though it has been used to produce camera-ready reports and a book [2]. Our system is deliberately limited in features, in order to simplify training and reduce memory burden. Nevertheless, it has the virtues of text repeatability, rapid turnaround, and reduced time burden on professional staff.

The technical reports, specifications, and documentation originated by scientists, engineers, and technical writers undergo many transitions before final publication. Numerical accuracy is very important. Lawyers write briefs, contracts, and legal papers in which text accuracy is essential. Other professionals, as well as students, must compile reports generated from original sources or merged from existing material.

Another feature of importance in technical work is the use of special characters, such as Greek or mathematical symbols. Standard typewriters, even those with interchangeable character sets, require more than one step and are prone to error when typing special characters.

Our particular concern has been the preparation of detailed technical reports that have to undergo several stages of review and revision [3]. An extensive burden has always been placed on the staff person to continually change and correct the text, which in various renditions could suffer the inadvertent introduction of new errors. Both magnetic-card and magnetic-tape typewriter-oriented word-processing systems have reduced the compounding of errors, but such systems are cumbersome and primarily of use only to trained personnel.

Improved word-processing practices [4] have eliminated many repetitious aspects of producing letters and forms. Commercial systems [5] are marketed with improvements in this category. However, we found they were generally not well suited for extensive reports. Although a variety of alternative text-processing systems were available at the time this program was initiated, none met our criteria, summarized here:

> Hardware: videoterminal-oriented, rapid memory transfer, convenient storage, fast and good-quality hardcopy; consistent with ongoing data-processing hardware inventory and planned acquisitions; hardware that must not lower the capability for data processing.
>
> Software: simplicity in use, report-compatible, capable of modification and expansion; programs for text processing written in languages normally used for data processing; implemented features to depend upon hardware and software already available; suitable for originating and editing computer programs.

Setting margins is one example of bypassed drudgery: our program automatically sets exact margins on each side of every page; the user need not be conscious of margin boundaries. Perfect copy can be obtained with no strikeovers. A premium has been placed on ease of use, and we have selected from established systems many useful features, especially those that function automatically without operator decision.

Computer-assisted writing and editing systems have been surveyed by P. I. Berman [6]. He presents a detailed evaluation of 16 systems. Although that survey concentrated on the technical author in the aerospace industry, it was author-oriented, and its relevance is general. Some considerations of common interest listed by Berman are:

1. Quantity of documentation
2. Frequency of changes
3. Need for extreme accuracy
4. Use of complex terminology
5. Need to reproduce mathematics
6. Need (in some cases) for security
7. Storage requirements
8. Need for a variety of outputs
9. High percentage of graphics

The system we have developed allows the professional to originate text at a videoterminal. Text can be entered at average speeds faster than on electric typewriters, due in part to automatic carriage-returns. Proofreading and corrections can be made on the videoscreen at a high turnaround rate. All text is stored on magnetic disks. One cartridge disk will easily hold the contents equivalent to a large book. Hardcopy is immediately available at 8 s/page from an electrostatic printer. Trained secretaries may efficiently enter routine (noncreative) changes or corrections in text.

Our text-processing-system hardware needs were met by existing data-processing hardware and careful planning of hardware additions. The only essential elements are a computer (mini- or micro-) and three peripherals: a magnetic-disk drive, a videoterminal, and a printer. With the proliferation of this type of equipment now occurring, it is likely that other professional staffs can likewise obtain the essential hardware at low cost.

Principles of progressive application and learning have been applied so that a user may achieve tangible results from the beginning. Although this report, which thoroughly documents the program, may appear lengthy and complicated, a new user can rapidly adapt to the program. Authors can be originating text in their first session, while many of the embellishments of the program can be learned in subsequent use. Secretaries can make many editing changes at their first keyboard session and incorporate refinements as they make progress.

Figure 1 (created by us at the terminals) outlines the major stages in text processing and illustrates the limited program capability for graphics presentation. Drafting personnel could have created a better-quality rendition--but only after a substantial queuing delay. Graphics of a more finished quality may be pasted on allocated space or appended as separate pages. (Further graphics capabilities are planned for the future.)

Of particular advantage is that the professional person can do the entire task without loss of time in interfacing with secretarial staff. Sometimes the transfer of work to secretaries entails delays, queuing, compounding of inaccuracies, and misunderstandings. The most important advantage to the author accrues where creative editing (rewriting) occurs during correcting sessions. The originator may enter additional thoughts and text changes that spontaneously arise, whereas the secretary is of course restricted to introducing marked corrections only. Deadlines aside, it is efficient to have secretarial personnel finish as much of the task as feasible. The files would be accessible for copy-reading assistance by technical editors.

Control over turnaround time is worth stressing as a significant practical advantage of this type of interactive text-processing system. Any office environment is likely to lead to priorities that are frequently superseded. In addition, a management hierarchy may periodically frustrate access to a typing pool. Thus, the ability to personally avert priority conflicts by a flexible, accessible word-processing system expedites completion of all paper work. This is particularly the case where preliminary drafts are acceptable for interim use after output from a computer terminal.

The system has been fruitful, not only in generating reports, but also in writing and editing computer programs. Line-oriented program text is easily handled, and there are the same advantages as in normal text processing. TXT has been used to write all our current data-processing programs and associated documentation, including the programming and this manual for TXT itself.

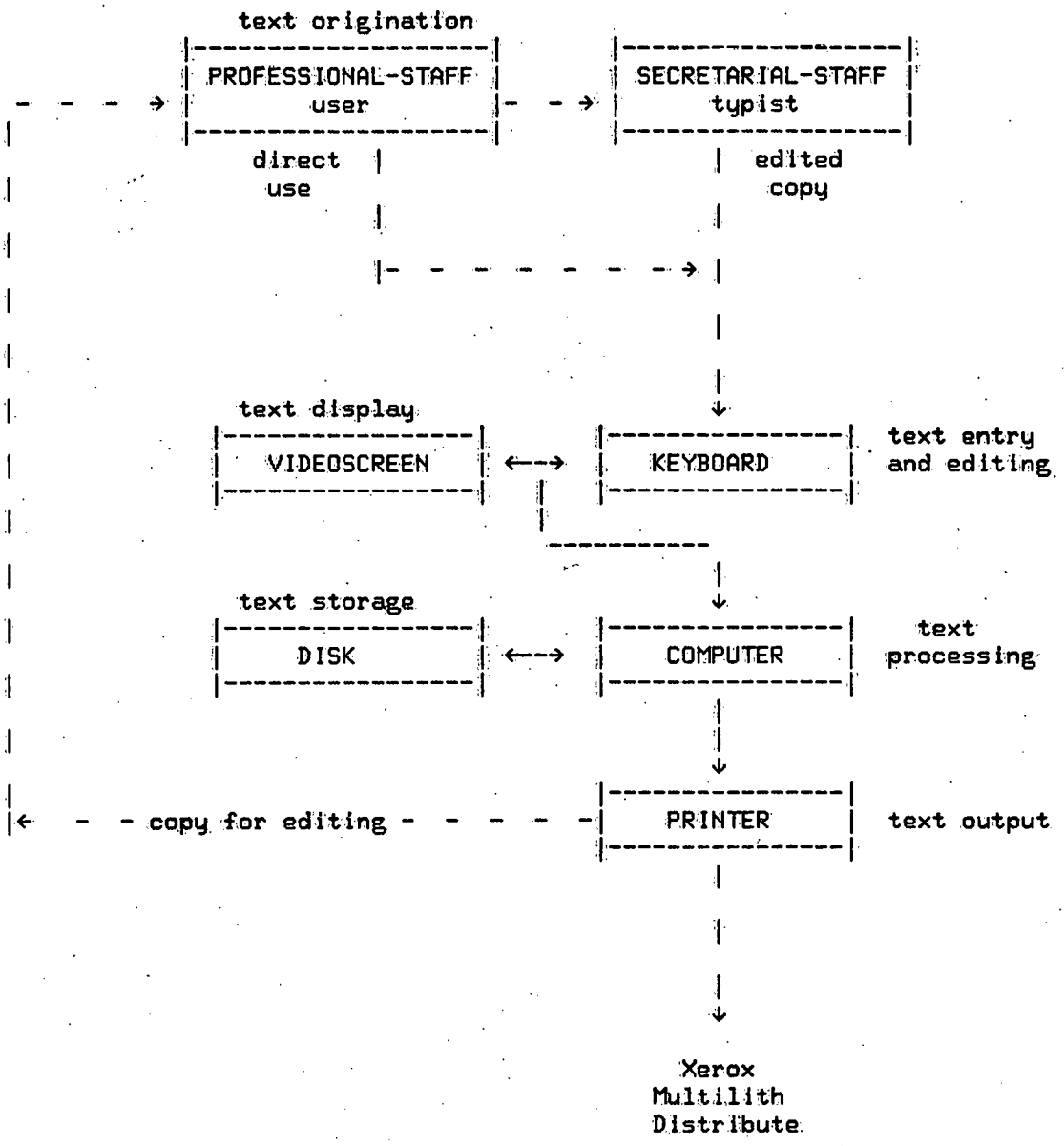


Fig. 1. Diagram of Computer-assisted Text Processing

SOFTWARE

TXT is written in FORTRAN (40%) and PDP-11 macro/assembly language. Originally the program was run under the PDP-11 DOS monitor. It has been converted to an RSX-11M (version 3.1) timesharing environment, allowing multiple users for a single CPU. The program requires about 15,000 words of (16-bit) memory using a resident FORTRAN library, and 21,000 words without resident library. RSX-11M utilities are extensively used in the present timesharing version.

Two steps are necessary to initiate an editing session: logging on with your user code and loading the program, TXT. TXT may be terminated through a control-Z ($_{c}$Z) at the program-command level. The program resides on disk; cartridge disks or disk packs are used for data and text storage.

TXT differs from other editing programs in that it is screen-oriented rather than line- or character-oriented. The user has command over the full screen of text: movement of the cursor to any position on the screen allows changes to the text by retyping, by character insertion or deletion, and by line insertion or deletion. Commands entered via special function keys do not directly affect the position of the cursor.

The program was designed around the ADM-2 videoterminal, with the idea of letting the terminal do local editing one screen page at a time, then transmitting this page to the file via the block mode. However, there are two drawbacks in permitting all the editing to be done at the terminal: The terminal does not notify the CPU when it is performing its internal functions, which can cause transfers to be missed; also, without signaling the CPU, the terminal scrolls lines off either the top or bottom of screen, causing the loss of text. These problems have been largely overcome by using the special-function keys to override certain terminal operations and performing these functions via computer software. The screen thus becomes a buffer that can be rapidly and easily modified.

HARDWARE

When we were deciding which hardware configuration to choose, a Digital Equipment Corporation PDP-11/20 minicomputer with Teletype, cartridge disk drive, and electrostatic printer/plotter was already in our possession for data-processing purposes. With burgeoning data and text processing, additional equipment was needed, including terminals, computer memory, and disk memory. (Also Selectric typewriters with MagCard I and II systems were available.) We chose to purchase videoterminals to replace our Teletype units. The existing cartridge disk drive and electrostatic printer/plotter were retained as integral parts of the system. We now have DEC PDP-11/20, /05, /35 and /60 computers. We also have on-line disk drives that combine a fixed disk and a removable cartridge disk in a single chassis, as well as a recently acquired disk-pack drive. Additional characteristics of hardware may be found in Appendix A.

Videoterminal

The videoterminal had to be compatible with both the hardware and software design of the word-processing system. In addition, it had to be suitable for ongoing data-processing operations. The Lear Siegler ADM-2 unit has a 12-in. screen and a full 128 ASCII-character set, and can display twenty-four 80-character lines. Characters are transmitted at 9600 baud. The keyboard is detached and coupled by cable. In addition to the standard alphanumeric keyboard, there are a cursor control pad, a numerical keypad, a set of 16

program-defined function keys, and edit and format keys. Except for the special functions, all keys are repetitive after a fixed delay, repeating at about 15 characters/s. The TAB jumps eight spaces at a time in columns that are an integer multiple of eight. Future terminal acquisitions will be ADM-42's (successor to the ADM-2).

Electrostatic Printers

The Versatec MATRIX 800A electrostatic printer/plotters were originally purchased to satisfy extensive graphic-output needs; they have capability for high-speed printing and plotting. A 100-dot/in. printer with overlapping nibs gives good text quality. To obtain high contrast over a long period of time, regular cleaning, occasional mechanical adjustment, and maintaining of toner supplies is necessary. The primary feature of interest is that a full page can be printed in about 8 s, using fan-folded 8-1/2 x 11-in. paper. The unit is quiet enough for an office environment, and reloading paper is not very difficult.

Impact Printer

A daisy-wheel printer/plotter has recently been leased to determine the viability of compatible output closer in appearance to that of an IBM Selectric typewriter. Although this printer is considerably slower (only 55 characters/s) than our electrostatic printers, it need only be used for final output. All drafts can be produced on the Versatecs.

Minicomputers

The PDP-11 computers have 16-bit words. The PDP-11/20 and /05 have a 28 k-word address space and are basically for single users; the PDP-11/35 and /60 have a 124 k-word address space with memory management, and support multiple users. The aging PDP-11/20 is currently used for experiment control at a remote test site and is to be replaced by the /05. (A videoterminal and disk drive are planned to be added to that installation also, for on-site program editing and experiment documentation using TXT.) The PDP-11/35 and /60 each provide the office with multiterminal text and data processing. The faster 11/60 is used for the more time-consuming data processing; it has magnetic-tape drives for data input/output, and disk backup. The 11/35 and 11/60 are connected by a high-speed link (MBD DA11-BJ), allowing each to access the other's peripheral devices.

Disks

We have three cartridge disk drives (two of which also include a fixed disk on the cartridge spindle) that use IBM 2315-type cartridges of 2.5 M-byte capacity, as well as a 254 M-byte disk-pack drive. The capacity of a disk pack is large enough so that a single one is normally left mounted indefinitely. This procedure increases media and drive integrity and life, and provides automatic disk access for text and data input/output on remote terminals.

More detail is provided in Appendix A on our system hardware configuration and peripheral devices. The use of all equipment in an office environment has been an important factor in selection. Noise, heat generation, air filtration, and size are considerations in this selection.

SYSTEM TRANSPORTABILITY

Program TXT has been designed to work with our specific hardware and monitor configurations and has not been tested to date on any other configuration. No attempt has been made so far to make TXT general enough to operate with other equipment. It cannot be guaranteed to perform properly in other cases. Depending on the variance of the proposed configurations from ours, the required changes could vary from minimal modifications to major rewriting, or could even be infeasible. At the very minimum, however, our text-processing system serves as a model for other systems of similar intent.

TXT should be transportable to later versions of RSX-11M without much difficulty. The program can perhaps be transferred to another PDP-11 monitor, such as RT-11 or DOS, without extensive modifications, although limitations on input/output and file structure must be accounted for. An earlier, more limited version did operate under DOS. The current version of TXT may be too large to fit in PDP-11/20 or /05 address space along with RSX-11M, possibly requiring efforts to reduce both TXT and RSX in size if this must be done. Conversion to a different computer (one not emulating a PDP-11) would be a tedious and extensive project. The entire assembly-language portion of TXT would require rewriting, and the entire TXT program would have to reflect the new input/output and file structure of the selected monitor software.

Revision to handle different peripheral devices on the PDP-11 bus, such as disk drives and printers of other manufacture, should not be difficult, provided their hardware characteristics are adequate. The only requirement on disk drives is that they be supported by the operating-system software and have sufficiently fast access and transfer times so that operator waits are minimized, and such that minimal text is lost when disk access is required during editing. (Relatively slow floppy disks may not be adequate.) Printers should provide legible output of all ASCII characters, preferably on an 8-1/2 x 11-in. page. Printers of other manufacture may require new handler software and modifications to TXT. The availability of both a high-speed printer for draft hardcopies and a high-quality, if slower, printer for final output is useful. Over 99.9% of our output is produced on the electrostatic printer.

Because our text-processing system uses a special terminal handler and is based on operator interaction exclusively through a videoterminal, much of the code depends, explicitly or implicitly, on the characteristics of the chosen ADM-2 terminal. In particular, if a different terminal were chosen (whose basic modes were significantly different from those of the ADM-2), extensive recoding of the input section would be required. The terminal should have modes similar to the "program," "protection," and "page-edit" modes of the ADM-2. The chosen terminal, in conjunction with the computer, must process data and instructions fast enough to allow screen updates to keep up with the operator without losing characters. We have found that the relatively fast ADM-2, when operated at 9600 baud, is adequate for a touch typist at secretarial speed. Some terminals (particularly those based on microprocessors) may not be fast enough. Our text-processing system is probably inadequate at 1200 baud, such as might be needed for a remote terminal operating over an unconditioned dial-up telephone line. In such a case, a more intelligent videoterminal than the ADM-2 may be desired, in order to perform most of the text processing without intervention of the host computer.

Two other terminals, the Lear Siegler ADM-42 and Soroc IQ140, which are similar in operation to the ADM-2, have been tested and found compatible with TXT, with the following caveats. To speed up the ADM-42 enough to avoid losing characters, it was necessary to perform a simple circuit modification which disabled the microprocessor halt line. This modification occasionally caused some minor transient "hash" to appear on the screen. Later ADM-42 models with

the 812 or later electronics board may be fast enough to avoid this modification. Confusing visual attributes (blink, blank, underline, and reverse video), enabled by control characters imbedded in text, appeared on the IQ140 screen. Elimination of these attributes requires reprogramming TXT to replace the imbedded control characters with some other set of one- and two-character codes for formatting. (Such a set would likely be confused with text.)

FILE STRUCTURE

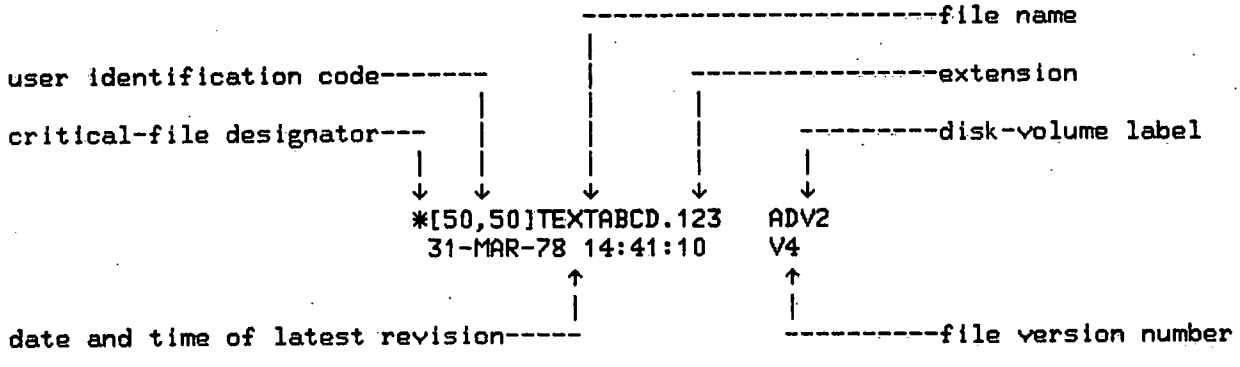
In anticipation of lengthy reports, a well-organized file structure is important. In terms of hardware, our files were originally built around portable cartridge disks. In terms of software, our files are compatible with the DEC RSX-11M operating system.

A single cartridge disk will store 4096 blocks (512 eight-bit bytes/block). This is enough storage of ASCII eight-bit characters to hold a large book or hundreds of small files. Each user is assigned an individual cartridge disk. Each disk constitutes essentially a "volume," and users may have additional disks, if needed. A 300 M-byte disk pack will store more than 100 times as many blocks as a cartridge.

Each disk is identified by a disk-volume label with a four-letter alphanumeric code XXXN, where XXX represents the user's initials, and N is the disk number of that user. The identification is handwritten on the outside of the disk and also recorded magnetically on the disk. The user is prevented from accidentally using another disk by a programmed check on the disk-volume identification label.

The user-identification-code (UIC) corresponds directly to PDP-11 file nomenclature. Brackets set off a pair of (octal) numbers [NNN,NNN] assigned uniquely to each authorized user.

Stored in each cartridge disk are files, each of which has a unique title. Here is a typical file label (printed at the heading of every outputted file):



First line: The asterisk in the example is optional; when used, it identifies a critical file, which is stored not only on the cartridge disk but also on the fixed disk. The user identification code is bracketed. The file name may have nine alphanumeric characters (64-character upper-case ASCII set

only), and the extension following the decimal point may have three more alphanumerics; file names too long for this format are truncated. The last four alphanumerics on the first line are the disk-volume label for disk identification.

Second line: The date and time printed represent the last date the file was accessed for editing. Thus, if additional copies are made of an existing file using the Print or Output commands, the date will remain unchanged. The file version number follows the letter V; normally the latest file version is printed or edited, unless a prior file is specifically addressed. File versions are numbered in octal sequence: 1 through 7, 10 through 17, etc.

FILE PROTECTION

Files are protected against inadvertent destruction by a variety of software and hardware features. Certain features inherent in RSX-11M have been adapted for use in this program.

A user may read but not write on files stored in other user areas. The two most recent versions of any given file in a user area are preserved; the editing of a new version will automatically cause the deletion of all but the last two versions in the user's area. Our practice has been to assign to a secretary a standard UIC [77,77] for use on any disk. If the secretary accesses Version 3 (V3) of a file from the originator's UIC, the file becomes V4 in the secretary's UIC. If the originator returns the file to the original UIC, the file version is incremented to maintain the distinguishing identification, becoming V5 in this example.

To avoid the accumulation of many obsolete file copies, various RSX-11M PIP commands are used to eliminate redundant file versions. These procedures may be invoked only according to a set of prescribed rules

FUNCTIONAL STRUCTURE

The program TXT appears to the user at two levels: Command and Edit; this is illustrated in Fig. 2. At the Command level, the user may choose any of five commands, or program pathways. At the Edit level, reached through two of the five commands, text modifications may take place. A return to Command level from the Edit mode may be effected at any time by pressing the EXIT key (special-function key #18). From any of the other modes (Help, File, Output, Screen, Print), return to the Command level may be accomplished by $_{c}$P (control-P: pressing control and P keys simultaneously). Appendix B contains a glossary of terms.

COMMAND LEVEL

The command options as written on the screen are shown here, and Table I summarizes their role.

```
Name, File, Edit, Help, Print, Screen, Output, Type, Copy
?
```

All commands assume the current file as default. The current file name may be replaced by going to the Name or Output command and typing a new file name.

TABLE I

COMMAND LEVEL OPTIONS

Name:	Designation of source and destination files; entry to editing
File:	File directory and manipulation
Edit:	Direct entry to Edit mode
Print:	Standard hardcopy output
Screen:	Standard videoscreen display
Output:	Nonstandard hardcopy or softcopy output
Help:	User assistance
Copy:	Copy files from one disk to another

By responding to the prompt symbol ? with the first letter of a command, we may excercise the various options. A carriage RETURN causes the response from the keyboard to be transferred to program control. A rubout (RUB) or backspace (←) may be used to correct responses before hitting RETURN. Control-U ($_{C}$U) cancels the entire response line, and $_{C}$P causes return to the command-level string for a new option.

The Name command permits a file name to be introduced through the keyboard. It is usually the first pathway entered for a session using TXT.

The File command provides a utility for directory listing, name changing, and deletion of files.

Edit operates on the current file.

Print provides standard automatically justified hardcopy, and Screen displays the corresponding softcopy.

Output gives various hardcopy and softcopy options.

User assistance may be obtained at the terminal by calling for Help.

The Copy feature allows file transfer between disks.

Name

Below is the sequence that results from responding with N. (The source file is the pre-existing text, which is to be edited.) First, the name of the source file is requested:

```
?N
SOURCE FILE (0=No  Source File)                #ABCD.123
 #TEXT.5
```

In this example, the filename (ABCD.123) from the preceding editing session could have been selected by default (carriage-RETURN), but TEXT.5 is the new source file to be called. If a new file is to be created, then no source file exists and it is necessary to enter the value 0. Carriage returns will preserve whatever characters follow the # mark. Upon response to SOURCE FILE, the following will appear:

```
DESTINATION FILE (0= Use Source File)      #TEXT.5
#
```

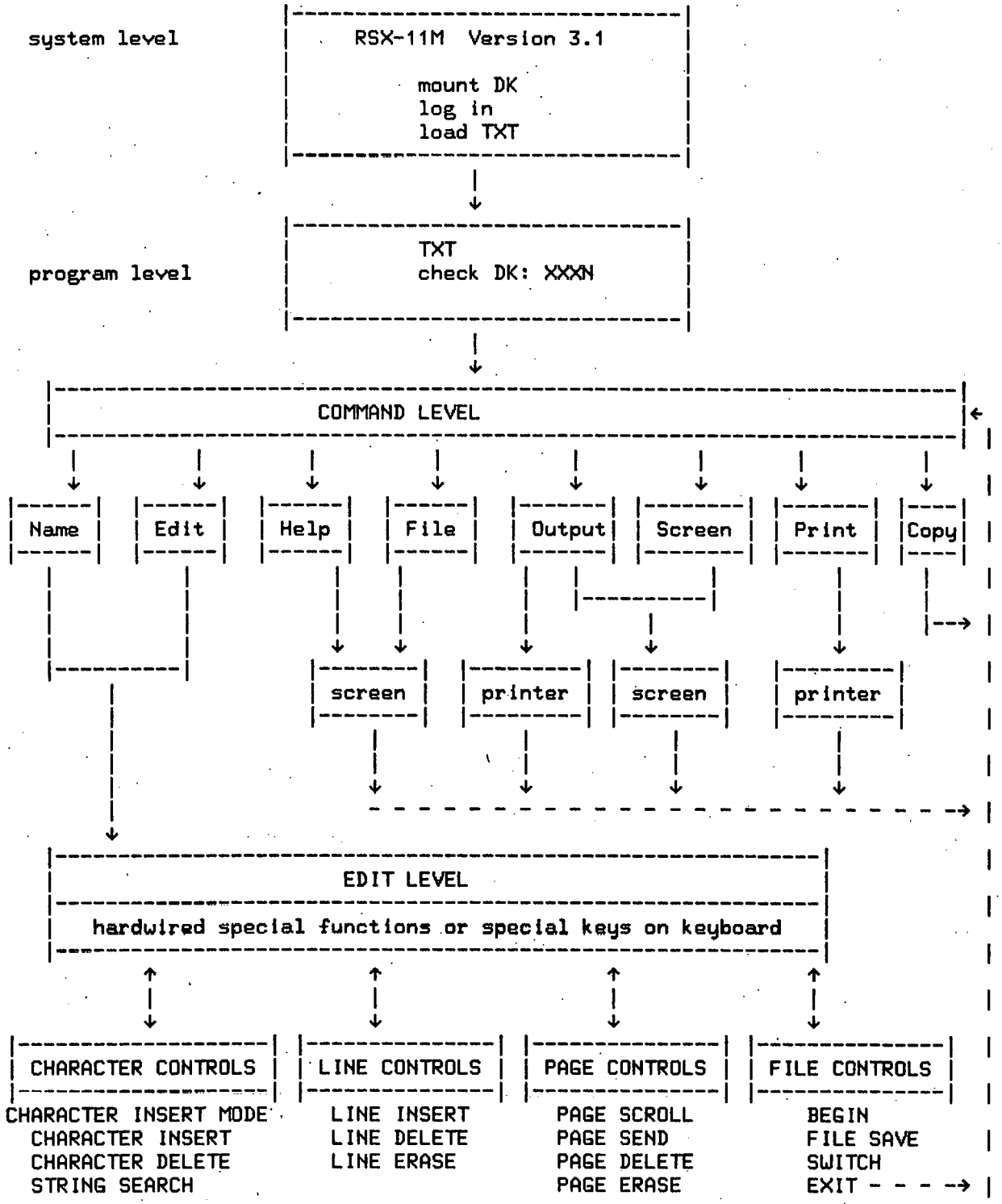


Fig. 2. Block Diagram of Program Structure

If no source-file name had been specified, a destination file would have to be given; this is the procedure for originating a new file. If a source file was listed, it may be used as the destination, or a new destination file may be created. Giving the destination the same name as the source results in a modified version of the source; giving a new destination name preserves the source file without change and allows creation of a new file with material taken from the source file. This latter procedure is convenient for using standard formats or forms.

Suppose the file TEXT.5 is to be edited. A carriage RETURN would now result in the following on the screen:

```
STRING TO BE MATCHED                    #
#
```

A carriage RETURN response opens the file for editing at the beginning of the file. A search may be initiated for any ASCII string that does not begin with a blank space, and which occurs entirely on one line, by typing in the character string followed by a carriage RETURN. The first occurrence of the string will be located and the entire subpage displayed on the screen, with the cursor at the first character in the string.

Edit

The response to an E command is simply:

```
?E
STRING TO BE MATCHED                    #
#
```

This is a shortcut for additional editing of the current file without having to restate the name of the source or destination files.

File

When F is typed in response to the command string, the request

```
DIRECTORY NAME                          #
#
```

appears on the screen, permitting a directory search for the desired file. A $_{c}$P response will cause a return to the command string; a carriage RETURN is equivalent to the response "*.*;*", causing a listing on the screen of the names of all files in the UIC; specific names, extensions, and versions may be substituted for the all-purpose asterisks. For example, the first letter of a file name (or extension, preceded by a period) is sufficient to cause a directory listing of all files (or extensions) beginning with that letter.

The directory scrolls one page at a time; the Page Scroll special function or the RETURN key will advance the page.

We intend to add the capability for deleting and changing names of files, as well as a "DISPOSSESS" function, which will transfer a file from the secretary's area on the user's disk volume to the user's area, with deletion of the file from the secretary's area. The file version number will be incremented if necessary. In deleting and renaming files, all versions of a file are automatically affected unless a specific version is listed; asterisks will not

be used in deleting or renaming. At present this file maintenance must be performed at system level with RSX-11M utilities.

When the file-manipulation capability is implemented, the following sequence will be part of the response to the F command:

```
RENAME FILE (Y,N)                          #N
#
OLD FILE NAME (OLDNAME.EXT)                #
#
NEW FILE NAME (NEWNAME.EXT)                #
#
DELETE FILE (Y,N):                         #N
#
NAME (FILENAME.EXT)                        #;*
#
DISPOSSESS FILE (Y,N)                      #N
#
NAME ([UIC]FILENAME.EXT)                   #[77,77]*.*;*
#
```

Print

The complete form of the Print command is P nn, where nn is the number of copies desired. P followed by a carriage RETURN will print one standard copy, 60 lines per page, 80 characters per line, single-spaced, automatically justified. An example of a response would be:

```
?P 15
```

The corresponding soft copy may be obtained by giving the command S, which causes the videoscreen to display text in approximately the same format as would have been printed by the P command, but left-justified only. With the S command, it is possible to scroll rapidly and nondestructively through a page-numbered text in order to inspect its composition.

Output

The Output command may be used in the form O nn to call for multiple copies. The following sequence occurs when the O command is used:

```
?O
Name of file for output, Justify,Inten,Width,Length,Space,Page,First,Device
     (current file)         Y      0     80     60      N      1     1      V
```

Upon responding with the first letter of one of the output parameters listed, the user is permitted to change the value of that parameter:

N: permits the file name to be changed.

J: permits automatic justification to be bypassed.

I: permits the print intensity (Versatec output) to be increased.

W: permits the number of characters printed per line to be specified, up to a maximum of 99 (Versatec).

L: permits the number of lines per printed page to be specified, up to a maximum of 66 (Versatec).

S: permits double spacing of printed output to be specified.

F: permits the page number assigned to the first page of output to be specified. The first page of any file need not be initiated as page 1; to blend with nine pages of existing text, the offset page number 10 could be given, in which case the first page would be 10, the second 11, and so on. An alphanumeric offset separated by a hyphen may be used, such as A- or II- .

P: permits the user to specify the first page of a file to be printed. The first page actually output may be selected to avoid unnecessary printing or screening of preceding pages. If a file is given an offset page number of 50 and printing is to start on the fifth page of the file, the first page output is given as 55. [Note: currently the page count works imperfectly if superscripts or subscripts are present.]

D: permits specification of the output device (Screen, Versatec, or Typewriter).

Help

User assistance may be obtained through the Help command. An H response will get an index of Help categories. Any category may be called by replying with the number of the Help category. The information will then appear on the videoscreen. The Help structure is as follows (see Appendix G for details):

```
?H
ENTER HELP QUALIFIER                    #
#
```

All commands assume the current file, unless superseded by entry of a new name using the Name or Output command.

Copy

Files may be transferred between disks and the directory of another disk may be addressed using the Copy command (see Appendix K). Because we have an interprocessor disk between two CPU's running under RSX-11 timesharing, we can also transfer files from the disks of one computer system to the other.

EDIT LEVEL

The keyboard is used to enter ASCII characters and to carry out certain control functions during editing functions. In addition, composition-control characters may be embedded in the text.

ASCII characters

Standard ASCII equivalents are transferred to memory and to screen. Either teletype or typewriter keyboard layouts may be used. Generally, 96 ASCII characters are compatible with the full range of equipment, that is, keyboard, screen display, and hardcopy. This allows both upper- and lower-case

characters. Many of the remaining 32 special ASCII characters are used for defined functions.

For Versatec output, the TXT repertoire currently consists of three sets of 96 characters--the standard ASCII set, plus a Greek/math set invoked by prefixing an ASCII character by a backslash (\), and a third set of special characters invoked by prefixing an ASCII character by a control-backslash ($_{c}$\). The third set as yet is only partially implemented, the intention being to design an italic alphabet.

Appendix C contains a cross-reference listing of all currently available characters arranged in order of ASCII number. Users will also find Appendix D useful. It contains a listing of non-Greek special characters, in groupings that should facilitate finding the one desired.

Special Function Keys

Up to 16 special user-defined functions may be called by programming a set of "special function" keys at the top of the LSI keyboard. These keys are labeled F1-F16. Above each of these keys we have marked the keyboard with the operations performed by these keys, as indicated in Table II and described in more detail below. When a special function key is pressed, the operation is directly invoked. For example, the page of text on the screen is scrolled 24 lines (a full page) whenever F1 is pressed.

TABLE II

SPECIAL FUNCTION KEYS

F1	PAGE SCROLL
F2	
F3	CHARACTER INSERT MODE
F4	LINE INSERT
F5	
F6	LINE DELETE
F7	BUFFER/UNSAVE
F8	FILE SWITCH
F9	SAVE
F10	STRING SEARCH
F11	
F12	PAGE SEND
F13	
F14	PAGE DELETE
F15	BEGIN
F16	EXIT

F1 (PAGE SCROLL). Advances a page without entering any corrections that might have been made on the screen; requires that the cursor be in HOME position (upper left corner).

F3 (CHARACTER INSERT MODE). Causes each entered character to be automatically preceded by the entry of a blank, shifting each subsequent character in the paragraph, so that subsequent text is not written over. When in this mode, a square appears in ADM Mode position 3 on the right-hand side of the ADM-2 screen. All other regular Edit functions are enabled during this mode. To return to normal Edit mode, CHAR INSERT MODE is pressed a second time.

F4 (LINE INSERT). Inserts a blank line at the position of the cursor.

F6 (LINE DELETE). Deletes the line at the position of the cursor.

F7 (BUFFER/UNSAVE). The first time pressed, displays contents of the Save Buffer. Pressing a second time returns to the Edit mode. When the Save Buffer is being displayed, there is an indicator on the screen. (The nature of the indication depends on the terminal being used; on the ADM-2, there is a flashing light at the right side of the screen, at mode position 4.)

F8 (FILE SWITCH). Permits switching to a secondary file. Return from secondary file is accomplished by pressing F8 again. When in the secondary file, the indication mentioned for the Save Buffer is visible on the screen.

F9 (SAVE). Stores in the Save Buffer all lines of text on the screen above the line that contains the cursor.

F10 (STRING SEARCH). Allows access to any string of characters in the text, during an editing session, provided the string occurs entirely on one line. The first character must be nonblank. The first search is initiated from the beginning of file; subsequent searches begin with the material that follows what was last displayed on the screen. If the string is not found, the end of the file will be displayed.

F12 (PAGE SEND). Sends a full page of screen text, including any corrections, to memory. The next sequential 23 lines will then appear on the screen.

F14 (PAGE DELETE). Allows deletion of a full screen page from the file in memory (in contrast to PAGE ERASE, which erases the material from the screen only).

F15 (BEGIN). Will revert to the beginning of the file, first offering the option of selecting some other text string, scanned from the beginning of the file. The current screen page will be stored, along with the remainder of the file, but no file exit takes place.

F16 (EXIT). Closes out the Edit file after sending the current page to memory and transferring the remaining text unchanged. Then returns to the Command sequence for the next action.

Screen Controls

Various screen controls may aid in editing (Table III). These are "firmware"-implemented black keys provided by the terminal manufacturer. Some have been disabled in order to avoid conflict with TXT-programmed functions; others are retained for special use. All affect the screen, but not computer memory unless the screen page is sent to the computer. Other keys are also listed in Table III. See Appendix A for more details.

Embedded Controls

To organize text, carry out formatting routines, and assist in composition of output, a variety of control symbols are embedded in the text. They are not acted upon during the text-entry session, but during output they are recognized and operate to carry out certain functions. Table IV summarizes these controls. These control characters appear on the screen as special symbols, each underlined. Although they are visible during file editing, they

do not directly appear on hardcopy output.

A $_{c}$N may be inserted at the beginning and end of a text sequence. It is primarily used when the author wishes to override the automatic justification feature of the normal printout. In a body of automatically justified text, an originator-justified section of text--such as a table--may be included by using a control-N pair. The second $_{c}$N of a pair will terminate the sequence, or a $_{c}$N* may be used in place of the second $_{c}$N.

The $_{c}$X is used to force the program to start a new page at that point, for instance, to leave ample space or full pages for illustrations, tables, or material to be pasted on.

The $_{c}$P acts as a carriage-return/line-feed. It can be used to terminate a line of printing, since it causes whatever follows it to start on a new line.

The $_{c}$W is for automatically centering the nonblank characters that follow on the line--as for titles. It must be placed at the beginning of the line to be centered, followed immediately by the character string to be centered. Initial blanks (following the $_{c}$W) are counted as characters in doing the centering.

The $_{c}$T.Bnn is intended for allowing a number (nn) of blank lines to be set aside (for equations and short tables to be pasted on). If space is not left on the page for the required number of contiguous lines, TXT will start the next page with them. During double-space printout of single-space copy, all material reserved in contiguous lines will be output as single-space copy.

The $_{c}$T commands must be on otherwise-blank lines directly preceding the material affected. The first character of the line must be $_{c}$T. The $_{c}$T lines do not appear on the hardcopy. More than one $_{c}$T command can be put on the same line; e.g.,

$_{c}$T.Cnn.Lmm

will cause the subsequent material to be output at nn characters per line, mm lines per page.

The _ (low dash--used for underlining on a standard typewriter: the lower-case symbol on the ADM-2 RUBOUT key) may be used to reserve blank spaces. While it appears on the screen as an underlined blank, it is suppressed in printing, leaving a blank space for each occurrence. It may be used (1) to prevent space-separated letters or words from being separated at the end of a line by automatic word justification, (2) to reserve space for writing special symbols or equations in text or numbers on forms, and (3) as an aid in aligning character strings of differing length that are to be centered by means of the $_{c}$W.

TABLE III

KEYBOARD CONTROLS FOR SCREEN DATA

CLEAR SPACES**	Erases page, putting spaces in all positions.
CLEAR NUL**	Erases page and puts null code in all positions.
PAGE EDIT*	Enables character wrap-around (Normally used mode for TXT program)
CHAR INSERT	Inserts blank space at the cursor position.
CHAR DELETE	Deletes the character under the cursor.
LINE INSERT*	Disabled in PROTECT mode.
LINE DELETE*	Disabled in PROTECT mode.
LINE ERASE**	Erases remainder of line.
PAGE ERASE**	Erases remainder of screen page.
WRITE PROT*	Used to protect fields.
PROT MODE*	Normally used mode for TXT program.
SEND PAGE*	Not used by TXT user.
SEND LINE*	Not used by TXT user.
SEND MESSAGE*	Not used by TXT user.
PRINT*	Not used by TXT user.
PROG MODE*	Normally not used (sometimes used for debugging TXT).
CONV MODE*	Normally used mode for TXT program.
PAGE BACK*	Nonfunctional.
PAGE FWD*	Nonfunctional.
RESET	Unlocks cursor.
CTRL	Key that alters meaning of any other key.
SHIFT LOCK	Locks upper-case letters into use.
BREAK*	Nonfunctional in edit mode.
ESC*	Not used.
NEW LINE	Moves cursor to start of new line. If cursor is at bottom of screen, scrolls top line off page and brings in blank line at bottom.
LF	Line feed: moves cursor down one line.
RETURN	Carriage return: moves cursor to left margin.
TAB	Advances cursor eight positions in a modulus of eight.
BACK TAB*	Not useful.
RUB	Deletes character preceding cursor and moves cursor to left.
TAB SET**	Sets nulls (tabs) in a vertical line on the screen.
SOM/EOM**	Not used.
HOME	Returns cursor to upper left corner of screen.
CR	Moves cursor to left margin (carriage return).
←	Moves cursor to left.
→	Moves cursor to right.
↑	Moves cursor up; also scrolls text downward when cursor is on top line.
↓	Moves cursor down; also scrolls text upward when cursor is on bottom line, inserting a blank line at the bottom.

---

* Keys not normally used by TXT operator.
** Dangerous--causes loss of fresh text.

TABLE IV

EMBEDDED COMPOSITION CONTROLS

$_c$F	Footnote (not yet implemented).
$_c$N	Justification control.
$_c$N*	Terminates justification.
$_c$P	Line termination.
$_c$U	Underline.
$_c$W	Center text.
$_c$X	Form feed (start new page).
$_c$Y	The next character becomes a subscript.
^	The next character becomes a superscript.
$_c$Tnn	Reserves nn contiguous lines.
$_c$T.Bnn	Inserts nn blank lines.
$_c$T.Cnn	Causes output line to be nn characters long; cancelled by $_c$T.C0
$_c$T.C-nn	Causes output line to be nn characters shorter than specified at output time; cancelled by $_c$T.C0.
$_c$T.C+nn	Causes output line to be nn characters longer than specified at output time; cancelled by $_c$T.C0.
$_c$T.C0	Cancels preceding line-width specification.
$_c$T.Lnn	Causes output number of lines per page to be nn.
$_c$T.L0	Cancels preceding page-length specification.
_ (low dash)	Reserves contiguous blank spaces.
` (grave)	Conditional hyphen; space eater.
\ (backslash)	Invokes first special character set.
$_c$\	Invokes second special character set.

The grave--which looks on the screen like an apostrophe slanted upward to the left (`)--can be used in justified work to insert a conditional hyphen within a word: if the word is too long for the remaining space on a line, it will be hyphenated at the `; otherwise, the hyphen will be omitted. This feature permits tentative manual hyphenation without having to delete hyphens because of subsequent changes in the text. In nonjustified mode, the ` acts as a "space eater"; on printout, it and its space are ignored.

Special Control Characters

Certain characters that are addressable on the keyboard are reserved for control functions: i.e., they are not part of the normal alphanumeric set. Some of the special control characters have already been spoken for at the RSX-11M system level, as indicated in Table V. Others have been assigned by us for use in either the command or edit levels of TXT. The next section describes operational use of the characters in Table V.

TABLE V

SPECIAL CONTROL CHARACTERS

RSX-11M SYSTEM LEVEL:

$_{c}$U	Deletes entire response on that line.
RUB	Deletes preceding character.
>	Ready for response at System Level (e.g., >HELLO or >TXT).
$_{c}$Z	Exit from program and retun to System Level.
$_{c}$Q	Restarts output to terminal.
$_{c}$S	Stops output to terminal.
$_{c}$O	Discards output to terminal.
$_{c}$C	Returns to command prompt.

TXT COMMAND LEVEL:

?	Awaiting response at Command Level (i.e., N or P).
#	Awaiting parameter needed to execute command.
RUB	Deletes preceding character.
RETURN	Returns cursor to beginning of line (left-hand margin).
$_{c}$U	Deletes entire response on that line.
$_{c}$P	Exit from program and return to Command Level.
$_{c}$C	Returns to System Level, but program is still alive.
$_{c}$Z	Exit from program and return to System Level.

TXT EDIT LEVEL:

?	Awaiting response to chosen edit-level function.
#	Awaiting parameter needed to execute edit-level function.
$_{c}$D	Deletes character at cursor.
$_{c}$E	Enters blank space at cursor.
$_{c}$Q	Scrolls text one line at a time, sending that line to memory.
$_{c}$V	Draws a vertical column.
^	Cursor up; backward scrolling.

TXT OPERATIONS

Major operations in the program, previously illustrated in Figs. 1 and 2, and features listed in Tables I-V, are described below.

Input

Entry of text is through the keyboard. Original text is typed in at the keyboard using the CRT screen to monitor up to 23 lines at a time. The original may be prepared from an existing draft, or it may be composed at the terminal. Because text may be entered without attending to carriage returns or pages, it is possible to compose at a high uninterrupted rate. This is perhaps one of the largest advantages to the originator--the ability to transfer thought processes to typed form rapidly. Moreover, as errors are made and recognized, or as changes are deemed worthwhile, the typist may return to any portion of the text and make immediate corrections. Thus, the first hardcopy of a text can frequently be a final copy, or at least one free of many distracting errors. Error-prone or slow typists benefit by being able to easily correct the text at the keyboard session.

Since the TXT program uses the same general file structure as PDP-11 ASCII data files, such data sets are also accessible to the word-processing program. Computer programs for data processing may be edited with ease at the terminal. TXT provides a more convenient program-creation and -correction facility than the PDP-11 edit programs that were provided by DEC. A burdensome repertoire of memorized instructions has not been necessary, and the most frequent editing steps may be performed in TXT by striking over characters designated with the cursor.

Editing

When an existing file is accessed for editing purposes, a variety of objectives may be met. Corrections from marked copy may be desired: In this case, the pages or subpages of interest are accessed and the particular passages deleted or replaced, or new text inserted. The cursor may be directed to a position of interest, and lines may be inserted or deleted, or remainders of lines or subpages may be erased.

Line Structure

Text is presented on the screen as 23 lines of 80 characters. This is defined as a screenpage. (A printed page is approximately three screenpages of unjustified text, depending on format. In this section all references to "page" mean "screenpage" unless otherwise noted.) For editing, portions of text may be addressed by using the "string-search" feature. It is possible to go back to the beginning (of the current file in the Edit level) by using the special function, BEGIN. Pages may be advanced by a PAGE SCROLL feature. A programmed TXT buffer allows the entire videoscreen file to be scrolled either up (forward) or down (backward).

When text entry reaches the bottom line of the page, the line feed control (NEW LINE) automatically scrolls the top line off the page into memory and readies a blank line at the bottom to receive more free-form text.

Two ways in which lines of text may be transmitted from screen to memory are by scrolling with the control-Q (cursor at any position) or by using NEW LINE (cursor at bottom of screen). Both move lines off the top of the page.

Control-Q brings up replacement lines at the bottom of the page; NEW LINE leaves blank lines at the bottom.

Lines may be scrolled or deleted, or blank lines inserted, through special function keys; in addition, the remainder of a line following the cursor may be erased by the LINE ERASE. The remainder of the page is also erasable, by the PAGE ERASE key.

Character Manipulation

Characters may be inserted in text by any of three means. The control-E ($_{c}$E) command causes insertion of blank spaces where new characters can be entered. The $_{c}$E character entry wraps text around lines up to the paragraph (blank-line) boundary. Control-D deletes characters with wraparound. The keys CHAR (BLANK) INSERT and CHAR DELETE may be used to insert blanks and delete characters at the cursor; however, there is no wraparound at the ends of lines (overflow characters are lost). These keys are useful for line-oriented text, such as computer programs and tables. Care must be taken that entries at the end of the line are not inadvertently dropped. The third method of character manipulation is through special function CHAR INSERT MODE. A blank space is automatically inserted before accepting any character from the keyboard, so that typeovers do not occur. The text is advanced up to paragraph boundaries. The mode is terminated by pressing the CHAR INSERT MODE key again.

Paragraphs

Some of the most useful features of TXT are paragraph-oriented. Paragraphs are separated by blank lines. Character manipulations involving the control-E and control-D keys will not propagate beyond paragraph boundaries. A disconnect can occur at the interface between screen and TXT buffer, unless the paragraph ends on the screen page. It is useful to keep paragraph length under two dozen lines.

Tabulations

To assist in forming vertical lines, particularly for tables, a special character (control-V) has been reserved to make vertical lines for tabulation, as shown here:

When control-V is pressed, the following manual operations are replaced: vertical line character, line feed, and backspace.

Searching for Character String

When additional changes are desired early in the current file, the special function BEGIN may be pressed. This will send the current page of text and then return to the beginning of the file. The user will then find the following on the screen:

```
STRING TO BE MATCHED                    #
```

If the first page of the file is to be addressed, one can press RETURN. If a specific ASCII text string is sought, the response should begin with the first nonblank character in the desired string. If the exact string is not located, the pointer will go to the end of the file, and the screen will show the last part of the file, or perhaps a blank page. The BEGIN button could again be pressed to renew the search for the text desired. A file is transmitted at about 1200 characters/s, or less than 1 s per page.

A current limitation of the string-search capability is the inability to accommodate carryovers; i.e., the string to be sought must occur entirely on one line of stored text. Thus, if the string is selected from a copy of justified text, the safest policy is to specify a string that appears just after a blank line.

When a page has been edited, the special function PAGE SEND may be pressed to transfer the text on screen to memory. When the entire editing session is complete, the EXIT key is used to send the last screenpage to memory and close the file, and the program returns to the command level.

Composition

With conventional typewriters, a major portion of a typist's work involves text composition. At the end of each line on a mechanical or electric typewriter, text must be terminated with either a full or hyphenated word. When inputting text to our electronic terminal, carriage returns are unnecessary: flush right-hand word justification can be performed automatically by the program. Words are not automatically hyphenated in the current version of TXT. However, hyphenation is rarely necessary when automatic right-hand justification is involved, as in this sample of text. These features result in considerable reduction in typing effort. In particular, whenever the number of characters on a line is changed, it is not necessary to rejustify all subsequent text; the program carries that out automatically. Alternatively, the operator may choose to control the word justification. Below is an example:

> The automatic justification can be overridden by $_{c}$N, as was done for this paragraph. For this example, the material has been indented by three spaces. In using this type of format, the typist must perform the text justification, including the hyphenation of words if desired. Use of tab, blank-character-insert ($_{c}$E), and character delete ($_{c}$D) features makes such a process rather easy. At the end of this paragraph, justification will be restored by a second $_{c}$N.
>
> > The automatic <u>left</u> justification can be overridden by indenting each line by eight or more spaces, as was done for this paragraph. The TAB key is convenient for indenting by eight spaces. If five or more blank spaces remain at the end of a line on the screen, as there were in the second line of this paragraph, that line will not be <u>right</u> justified by the program if the line following begins with at least eight blanks. It was necessary to type this paragraph with no word wraparound. Further, since this paragraph was typed on the assumption of an 80-character line width, it would not be satisfactory to try to print it with fewer than 80 characters per line: that would produce extra lines with carried-over words.

Line Composition

> A number of composition-control features have been implemented that are invoked by using an instruction of the form $_{c}$T.yx, where y specifies the quantity to be controlled (e.g., C for number of characters per line, L for number of lines per page, JL for left justification only, B for blank lines), and x is an appropriate integer. For instance, this paragraph was formatted as you see it by preceding it with $_{c}$T.C-22. This caused the output to be centered and justified with a line length that is 22 characters shorter than the line length specified in the output instruction--a feature that is useful for lengthy quotations. To restore full-width justification, $_{c}$T.C0 will follow this line.

When single lines are to be terminated, the line terminator ($_{c}$P) may be used. This defines the intended end of line so that the automatic process will not concatenate the two text strings. As mentioned above, if the line that follows does not begin sooner than eight spaces from the left-hand margin (one tab), the text strings will not be concatenated, even without the $_{c}$P. If two lines are separated by a blank line, the two lines are never concatenated, and the blank line is always preserved.

These features are subject to modification and expansion as our exerience with the program leads to ideas for further improvement.

Page Composition

All references to "page" in this section mean "printed page," rather than "screenpage." In order to start a table or figure at the top of a page, a formfeed ($_{c}$X) is used. A second $_{c}$X at the bottom of the table or figure ensures that the entire page is dedicated to that table or figure.

Contiguous lines may be set aside for tables, figures, or headings that do not require a full page to be set aside. The $_{c}$T embedded command followed by a number nn will reserve nn lines on the page; if the room left on the page will not allow all of them to be printed on that page, the remainder of the page will be left blank and the full set of contiguous lines will be printed starting at the top of the next page. Blank lines are not printed at the top of any page unless automatic justification is overridden by a $_{c}$N, or unless a dummy character (low dash [_] or grave [`]) is used to define the top of the page. The $_{c}$T command must appear as the first character on a line (which will not be printed) directly above the contiguous block.

Paragraphs may be defined by separation with a blank line. Not only is this convenient for easy reading, but it helps in the editing process. If a given paragraph boundary may be seen on the screen, then all changes within the paragraph will be satisfactorily entered. If the paragraph spills over to the (unseen) next screenpage, some blank spaces will be introduced at the rollover point. After scrolling up additional lines, the invalid blanks may be erased. Keeping on the screen the end of the paragraph being edited avoids this housekeeping operation.

Sentences are customarily separated by two blank spaces and words by one. Paragraphs may be indented. TXT will preserve two spaces that follow a period. Lines will be terminated (a) at a space, (b) following a hyphen that is

not followed immediately by another hyphen, and (c) following a conditional hyphen. In case (c), a hyphen is printed at the end of the line.

Embedded control symbols and linked encoding are suppressed on output. In the normal run of text, control symbols such as $_cN$ do not affect line spacing; on the other hand, a $_cT$ requires the allocation of a separate screen line, which is ignored on formatted output. A special character, such as the Greek letter α is coded with a backslash followed by the letter a. Both of these characters appear on the screen, but in output only the Greek character is printed. Notice also that the in-line coding automatically eliminates the space occupied by the backslash. However, in tables in which the $_cN$ format control is used, the special character is positioned at the point where the linked code begins; this allows tables to be set up without automatic justification overriding the originator's positioning of the characters. A sequence of special characters such as αβδ is assembled at the starting point of the coding to avoid spaces between the symbols. Superscripts and subscripts are handled in-line by printing such lines with half spacing, as done above for the control symbols demonstrated.

Titles, headings, etc., may be centered by use of the $_cW$ embedded command. The $_cW$ is placed as the first character on a line; the entire character string (including leading blanks) that follows the $_cW$ will be centered on the line during output.

Blank lines may be reserved for figures or tables to be pasted on before duplication. The embedded command $_cT.Bnn$ reserves nn blank lines. The command $_cTmm.Bnn$ reserves mm contiguous lines, of which the first nn are blank. An example of contiguous allocation for a 10-line table is as follows:

```
cT11
cW TITLE
cT.B05
```

Space for a figure may be allocated in the following manner:

```
cT11.B05
cW FIGURE TITLE
```

Equations

Equations or formulas may be handled in several ways, thereby giving the user considerable flexibility. If no characters other than the standard ASCII set are employed, then straight-line equations may be written in either the automatically justified or unjustified ($_cN$) modes. Superscripts and subscripts can be placed on separate full lines or half-line controls may be invoked. The program has provision for a superscript or subscript line shift for the full line; however, there is a known gliche in its operation at present. The superscripts and subscripts may also be generated in line with special linked coding. Such linked coding is necessary with our hardware to produce the Greek letters and mathematical symbols usually used in equations. Some examples are given below:

Eq. 1a: $$P(x) = 1 - e^{-x^2 NkT/2}$$

Eq. 2a: $$Y = 238 * k^3 * (s/L)^2$$

With the recently added ability to adjust superscripts and subscripts and use special characters, the equations may be written as follows:

Eq. 1b: $$P(x) = 1 - e^{-x^2 NkT/2}$$

Eq. 2b: $$Y = 238 * k^3 * (s/L)^2$$

Equations 1b and 2b are written with linked coding; on the screen appears the coding that corresponds to the key strokes neccessary to create the special characters and the superscripting/subscripting. For example, the exponent is coded in-line with carets to elevate the argument. The powers of 2 and 3 are generated by superscripting the special characters (backslash 2 and backslash 3) needed to print smaller numbers.

A variation on the above method may be employed by a half-line spacing capability that permits a full line of superscripts or subscripts to be designated by placing either a caret (for superscript) or $_cY$ (for subscript) as the first character of the line. The example produced below would thus appear on the screen in proper juxtiposition without any linked coding:

Eq. 2c: $$Y = 238 * k^3 * (s/L)^2$$

Re-edit

Text may be re-edited at the same or a subsequent keyboard session. Between sessions, one can mark corrections on hardcopy and then regain access to the file via the Name command. If it is necessary to return to an earlier portion of the text, the BEGIN special function may be used before exiting from the current file, or the Edit command may be used after EXIT from the current file. The Edit command allows direct access to any character string that appears entirely on one line, and does not require reidentification of the current file. The file version number is incremented for each EXIT. Only the current and immediately preceding versions are retained on disk, in order to avoid tying up extensive disk space. However, the program can be modified to permit all prior versions to be retained until they are purged. In addition, changing file names can protect an earlier version indefinitely.

Merging

It is often expedient to use an existing file or files as source material for a new file. To start with an existing file, one simply designates the old file as source and assigns a new name for the destination. The old file will be preserved; the new file will consist of all material conveyed from the old file plus any new text added at the keyboard.

To merge two existing files, the "secondary-file" feature may be used. Access to a secondary file during any editing session is obtained by pressing the FILE SWITCH key. The response at the bottom of the text page will be:

```
SECONDARY FILE                          #SOURCE
#
```

The source file is the default. When the file name is designated, the program asks for an ASCII string in the secondary file:

```
STRING TO BE MATCHED                    #
#
```

Recovery from the secondary file may be achieved at any time by pressing the FILE SWITCH key. When in the secondary mode, the ADM-2 Mode light #1 blinks repetitively.

Before switching to a secondary file, all text that is intended to precede the new text should have been sent off the page (by either page send or line scroll). The new text from the secondary file will be positioned immediately afterward, starting on a new line that appears at the top of the videoscreen.

Text in the secondary file may be skipped over by addressing a particular text string, or by using PAGE DELETE or LINE DELETE. Text is sent by LINE SCROLL ($_{c}$Q) or PAGE SEND. When all intended text has been sent, FILE SWITCH is pressed to return to the previous position in primary file. Normal editing of the screenpage of secondary-file material may be performed before sending the page to the primary file.

To ensure that the new text has been properly merged into the primary file, the user may reverse-scroll, or press BEGIN to return to the beginning of the primary file. Often some minor editing is needed to integrate the text transferred from the secondary file.

Output

All text is stored on disk when the EXIT key is pressed. Two commmands are available for printing hardcopy. The Print command writes (up to 99) copies of the current file in justified mode, single-line spacing, 80 characters/line, 60 lines/page, on the MATRIX electrostatic printer.

All pages are sequentially numbered, from (1) to (999). If a file contains only one printed page, the page number (1) is suppressed. Under the Print command, the page numbers are automatically positioned at the bottom of the page, centered and surrounded by parentheses.

Options are available under the Output command, as explained in the section entitled COMMAND LEVEL. The file may be displayed (softcopy) on the screen or printed (hardcopy) on the electrostatic or daisy-wheel printers. Justified or unjustified mode may be selected. The number of lines per page may be extended from the standard 60 lines up to 66 lines or reduced to 1. The number of columns of text (characters per line) may be reduced from the standard 80 down to an arbitrarily small number, or increased to 99. If nonstandard column widths or lines per page are designated, the notation Cnn or Lnn is printed with the file label.

Double spacing between lines may also be chosen. The automatic conversion from single to double line spacing creates some anomalies in

appearance. If text is intended ultimately for double-spaced output, some special care may be useful in the original text generation (see Appendix E). Lines reserved by $_{c}$T are not now double spaced, but there are plans to modify this feature.

Each page of a file on output is sequentially numbered. All page numbers may be accompanied by an alphabetical prefix. Examples: A-1, I-5. Or the first page of a file can be output with a numerical offset; example: Start the first page with number 70. Moreover, printing of a file may start at a preselected page of the text (although currently the page count is imperfect if superscripts or subscripts are present).

Files may be printed in contiguous fashion, with sequential pagination. This is carried out by storing in a file XXX.CMD a list of files YYY.ZZZ. The @XXX.CMD response to file Name request will cause the sequential printing of the list. (See Appendix E.) Output on the printers or screen may be aborted at any time by using the $_{c}$P.

The normal automatic page-numbering scheme can be overriden (as it was done in this report) to produce page numbers elsewhere on the page (see Appendix E). Reports and documents may be produced with running heads, dates, and boilerplate.

PROCEDURES

Informal procedures are used as much as possible. Enough terminals are available to avoid queuing or scheduling: to encourage origination of text by the professional staff, terminals should be readily accessible. Under a timesharing mode, it could be adequate to provide keyboards and terminals for each user station, but printers only for clusters of terminals. In our case, one printer is assigned to each terminal. Simplified access to disks and drives also facilitates system use. On the other hand, secretarial support requires access to each designated disk, as well as a printer. Our secretaries have been trained primarily on one terminal and are aware of storage locations for disks.

Normally, the originator initiates text at a terminal, making some corrections as proceeding. The hardcopy output with corrections marked is placed in a priority-ordered basket for secretarial action. The secretary obtains the disk in accordance with the label in the upper-right-hand corner of the first page of the handcopy, and logs into our standard secretary user's area (UIC [77,77]). Although the secretary may access source files in the staff user's file space, the new file is generated in the secretary's area; this gives added protection against inadvertent destruction of files.

The secretary completes all corrections and produces a copy that is returned to the originator; location of the corrected file is now indicated in the file label on the first page. Further iterations by either the secretary or professional staff member may occur as needed. If in a hurry, the originator has the option of short-circuiting the entire process by producing final output. This is often convenient if last-minute changes must be introduced in the text at the keyboard. The secretary may also separate the fan-folded pages. If a page tears, the operator can quickly print another copy of the file or of the page. Experience indicates that, for handling convenience, files should not be longer than 10 or so hardcopy pages.

HUMAN ADAPTABILITY

The critical feature of this system is its compatibility with direct use by authors. This compatibility is achieved by

1. providing an adequate selection of features;

2. not overtaxing human memory requirements;

3. choosing convenient keyboard and screen arrangements;

4. using relatively fast-access, large-capacity disk storage;

5. supplementing the system with a full range of secretarial services;

6. allowing for special-purpose programmed additions;

7. having computer-support personnel carry out those periodic file-management procedures that are best accomplished by a specialist;

8. structuring the program to permit the user to learn while producing;

9. minimizing the necessity for embedded and special-purpose codes;

10. adapting the software to accommodate existing hardware limitations, keeping in mind multipurpose (data, graphics, and text) processing;

11. devising a number of ways of treating technical notation, ranging from direct one-to-one symbol correspondence to more elaborate linked encoding.

A person who can type rapidly has a big advantage. Learning to use the system is not difficult: in less than one hour a professional can be originating text, or a typewriter-qualified secretary can be productively editing existing text or entering new text.

All special-function keys have adjoining markings with simple mnemonics. These keys have been ordered in a functional mannner, with attention to risk of inadvertent striking of adjacent keys. The other special functions and embedded controls are few enough to be typed onto gummed labels attached to free space on the keyboard. All commands and responses are identified by mnemonics, and they are displayed on the screen for visual reminder. It is convenient to post code conversion tables adjacent to the typing area to simplify the entry of ASCII code combinations for special symbols.

The Help command provides convenient access to information files that give increasingly detailed explanations of various functions. (Many portions of this report are accessible under the Help processor.)

Because special characters must be encoded by striking more than one key and because our video screen will not render the special characters directly, we provide the next best capability: the key-stroke images are displayed on the screen in direct relationship to the sequence of strokes.

FILE MANAGEMENT

Under RSX-11M, all versions of a file may be preserved on a cartridge disk; however, we choose to save only the latest two versions to avoid clutter and overload of storage capacity. Also, a Purge command is available which deletes all but the most recent version.

To protect files against loss due to disk crash and to avoid exceeding disk capacity, computer-support personnel are assigned to manage disks. The contents of each disk are periodically backed up by writing to magnetic tape. The period depends upon frequency of use and may be as short as one week, or more typically one month. Tapes are retained until several backup cycles are complete. If a disk crashes, the most recent backup tape may be used to reconstruct the disk. It is wise to save hardcopies of corrected text so as to be able to fully reconstruct files that were changed since backup.

In addition, it is possible for critical files to be stored on the fixed disk. The user achieves this by placing an asterisk in front of the file name. When the EXIT function is activated, the file is then stored on both the cartridge and the fixed disk. To avoid overcrowding, only the latest version of a file is saved on the fixed disk. When a disk is backed up, all critical files from that cartridge may be purged from the fixed disk. If the cartridge crashes, the critical files can be recovered from the fixed disk through either TXT or the PIP utility program.

Upon backup of cartridge disks onto magnetic tape, all files but the latest version are deleted. This is done by the PURGE command of PIP. Unncessary file copies are thus not retained on the user's disk.

In transferring the latest file version, original file dates (i.e., date the file was last edited) are retained.

LIMITATIONS

Some limitations of the system are deliberate, some are subject to future remedy, and some are a result of the hardware configuration. Possible hardware and software extensions of the system are described in the next section. The order in which the limitations are noted below is only roughly in order of their significance.

Text draft output quality is limited by the capabilities (and maintenance) of the MATRIX electrostatic printer; this choice is still the best available compromise on speed of output, cost, quietness, and quality. The output can be copied easily for offset reproduction or Xeroxing. However, it does not provide the character quality of fully formed character impact printers (such as an electric typewriter). The electrostatic printer/plotters have proven reliable and easy to use. Two undesirable characteristics have been noted: If paper movement is very slow or at maximum speed, contrast can be relatively low. Also, output is not immediately visible, but can be viewed only after about 20 lines have been printed. The print-contrast limitation can be overcome by intensifying the output at a sacrifice in print speed, and, sometimes, at the cost of visible background on the copy.

Sufficient experience has not yet been obtained with the daisy-wheel character-impact printer/plotter to evaluate its limitations fully. Output quality appears excellent (provided the ribbon is changed periodically and the print head is adjusted occasionally), and camera-ready quality is possible with single- or multistrike film ribbon. Speed is relatively slow (55 characters/s), and ribbon and print-wheel interchanges (required to change character set) are somewhat clumsy.

The ADM-2 terminals have proven relatively reliable and easy to use. However, they have several undesirable characteristics. Correction with the rubout key, which is often used, required two-step operation (SHIFT-RUB). The left arrow has been programmed as a substitute at the Command level. Also, the keyboard tiers are offset differently from those of a standard typewriter,

making touch-typing more error prone (corrected on the ADM-42).

The ADM-2 keyboard has two-key rollover protection, but a character can be lost when typing rapidly. This omission is frequently caught by the typist and immediately corrected.

Some limitations have recently been overcome by software development within the confines of hardware restrictions, as described in the following section. Half-spaced superscripts and subscripts may now be used in single-spaced text. To avoid extensive paper waste, we tend to produce as much material as possible with single-line spacing. Text can be underlined with a one-fifteenth of a line space difference. Exponents and powers of ten can be written directly in single-spaced text.

Because the electrostatic printer we use has only the standard ASCII 96-character alphanumeric set, Italic and Greek letters are not directly available. Before recent software improvements, equations and special characters were entered by hand or by paste-over. An equation with Greek symbols, for example, could be typed on an electric typewriter and then pasted over space left in the text. This was also done for illustrations. Full (numbered) pages may be set aside for figures. Limited to single-spaced text, many equations can be handled adequately.

A comparative analysis [8] of some dedicated and multipurpose systems using technical text processing has recently been brought to our attention. In the space below we will illustrate some of the ways that TXT accommodates the examples or lacks capability.

For both super- and subscripts tied to a single character, we can accomplish this either in the equivalent half-line mode or by linked coding:

Eq. 3a: $x_i^2 = 5$

Eq. 3b: $x_i^2 = 5$

The most challenging test equation in the Seybld Report [8] is written as follows. At the moment, we do not have normal need for h-bar or much use for a radical; however, they can and probably will be added to our symbol set.

Eq. 4:

$$\Psi_\omega(y,z) = \frac{1}{\pi\omega}\int_{-1}^{1}\frac{d\xi}{\xi}\sin(p\omega\xi/2h)e^{ipz[(1-\xi^2)^{1/2}]/h}\,e^{ipy\xi/h}$$

Setting up Eq. 4 is by no means a simple task with TXT, as it required several iterations--mostly due to lack of practice with the special notation involved. Because the $_cN$ format for equations is not operating as it should now, a trial-and-error routine was necessary to establish the correct arrangement. However, the reproducibility of the equations often outweighs the initial time investment and any necessary compromise in form, particularly when measured in terms of accuracy and professional staff-time expenditure. With any rapid hardcopy device, such as an electrostatic printer, equations can be

readily proofed, thereby decreasing concern over exact visual matching of display and hardcopy. In any event, all the methods provide substantial improvements over manual typewriters.

In reviewing alternative technical text-processing systems [8], TXT appears to satisfy all the key criteria, with some tradeoffs and remedial omissions. Our system has not progressed to the photocomposition capabilities of Stanford University's TEX, which appears to be similar in hardware, but which differs in programming a higher-resolution, wider printer with a larger character set for photographic reduction. Although TXT is used for final camera-ready output, the main production is in proofs and internal documents.

IMPROVEMENTS

Improvements are continuously being introduced. he most recent version of TXT produces text on electrostatic printers in plot mode, with appropriate interfaces to the videoterminal and daisy-wheel printer. Most of the early software-related limitations have been eliminated, since software-generated characters can be arbitrarily positioned. Currently, flush right-justified hardcopy consisting of standard and nonstandard (Greek and mathematics) characters is being produced at approximately the same output rate as before in print mode. This report is itself an example of copy generated entirely by the TXT program and hardware in camera ready form. Except for the first five pages, where the output was directed to a daisy-wheel impact printer, the remainder of the report was produced on the electrostatic printer.

Superscripts, subscripts, and underlines are now provided. Since the daisy-wheel printer has similar capabilities, interfacing it for this hardcopy output mode did not present a major difficulty. Because the videoterminal has no capability for display of fractional-line or -character spacing or software-generated characters, the interface here is somewhat clumsy, involving extra lines, embedded control characters, and two-character codes for nonstandard characters. This clumsiness may be alleviated to some extent by additional videoterminal hardware character sets. Such sets can be installed on ADM-42 terminals, which are on order as system additions compatible with the ADM-2.

A SAVE buffer has recently been provided. When the SAVE (F9) key is pressed, all lines on the screen above the cursor are copied to a buffer. The original lines are not erased. Additional lines from other portions of the file may be appended to the buffer in a manner helpful in reorganizing passages. The SAVE buffer may be entered and edited before inserting some or all of its contents back in the main file. Buffer contents are transferred back by line scroll or page-send instructions. The contents of the SAVE buffer are not deleted during transfer. The buffer is erased only upon EXIT from the session.

Future modifications will allow the addressing of justified output text. This would permit calling in a page that corresponds exactly to the hardcopy output. Other improvements planned include protected fields and global changes. (An example of a "global change" would be an instruction to change "May 31" to "July 15" whenever the former appears in the text.) In general, a variety of program features may be accessed by the output ports provided. Special user-developed programs for hyphenation, reference numbering, automatic footnotes, and indexing could be implemented using RUNOFF, a PDP-11 DECUS Library program.

If a large amount of text already exists in printed form, and it is desired to enter that text for merging or corrections, an optical character reader (OCR) could be secured. Depending on the type-face versatility of the OCR, text is read and stored as ASCII characters in a file. This can be made accessible for manipulation via the TXT program.

Through modems and appropriate interfaces, more sophisticated computer systems and programs are usually accessible from time-sharing networks. For example, at ANL the SCRIPT [7] facilities could be connected. Because of the greater complexity, however, this would best be done only in specialized situations with a single trained person. In principle, files created under TXT can be rendered suitable for producing high-quality typeset or camera-ready output.

CONCLUDING REMARKS

Before implementation of TXT, we suffered delays, unnecessary effort, and aggravation in word processing. Now we are able expeditiously, efficiently, and smoothly to generate and consummate our written communications. Substantial value may be credited simply to computer memories that hold at least several pages of information and to editing schemes that allow direct character substitution; the improvement in the next order of importance has been in the use of videoterminals to facilitate the process of text visualization and modification. Our own contribution consists of adapting existing or planned data-processing systems to a software program tailored for a user who is not a specialist in word processing. Our hardware centers about rigid disks to emphasize high-speed file transfer and electrostatic printers for high throughput. As the system has evolved, refinements have been introduced to improve the end product in terms of appearance and compatibility.

We are able to produce camera-ready copy suitable for almost any duplicating system. Future improvement in file compatibility and standardization should extend this capability to match automated composition systems. At the moment, we have achieved an effective adaptation for a multipurpose (data, graphics, and text), multiuser environment.

In the six or so years of development of TXT, other word-processing systems--some with dedicated hardware--have appeared. Our system is appropriate for a user who already has or is acquiring general-purpose data-processing hardware but has no software for a satisfactory system of text processing. Adaptation to different computers and peripherals will require knowledgeable reprogramming, though we would anticipate the investment to be worthwhile. TXT fulfills most technical word-processing goals--but not to the extent of some dedicated systems.

Improvements that could be made to TXT include keyboards with multiple character markings, more file-handling features, more sophisticated automatic page-composition features, better treatment of complex equations, automatic indexing, and additional special characters. TXT will also benefit from integration with image-producing graphics programs.

We will continue to adopt improvements, yet keeping in mind our objective of providing a text-processing system primarily for use by professional staff.

APPENDIX A

SELECTED HARDWARE CHARACTERISTICS

VIDEOTERMINAL: Lear-Siegler model ADM-2 (or ADM-42)

Display

12-in. diagonal CRT TV raster-scan screen, P4 phosphor
dual intensity (fainter for protected fields)
128-character ASCII character set, 5x9 dot matrix
80 characters/line, 24 lines
blinking fields
reverse image cursor
status display

Keyboard

detached
standard typewriter keyboard
cursor control keyboard
numerical keyboard
16 program-defined function keys and space for labeling
lighted format keys
transmission mode keys
edit keys
15 cps repeat

Edit Keys

page and line edit
character insert and delete
line insert and delete
erase to end of line and page

Format Keys

tab, backtab, and tab set
program mode (display control characters)
protection modes
page edit mode (character wraparound)

Transmission

20-mA current loop serial controller (DL11)
up to 9600 baud (switch selectable, operated at 9600 baud full duplex)
conversation mode (full- and half-duplex character transmission)
block mode (line, page, or message transmission)
absolute and incremental cursor addressing

ELECTROSTATIC PRINTER/PLOTTER: Versatec MATRIX model 800A

resolution 100 dots/in.
plot speed 1.2 in./s
print speed 500 lines/min
character font 7x9-dot matrix
100 print columns
96 ASCII character set (TXT uses large software character set in DMA plot mode)
simultaneous print/plot
1 line (print/plot) buffer
direct-memory-access parallel controller (model 121, 1 microsec/byte)
accepts fan-fold 8-1/2-in. x 11-in. electrographic paper
paper-out sensor

CHARACTER-IMPACT PRINTER/PLOTTER: Agile A1 Data Terminal (option Q96A)

resolution 1/120 in. horizontal and 1/48 in. vertical
print speed 55 characters/s
changeable daisy wheel fonts, 158 print columns (12 pitch)
variable pitch and line spacing, margins, tabs, and form length
modes: underscore, overstrike, justification, word wrap, proportional space, center, program, plot
256 ASCII character buffer with X-on and X-off control
serial 20-mA current loop controller (DL11, 1200 baud full duplex)
accepts up to 14-7/8-in.-wide fan-fold paper (friction or pin feed)
paper/ribbon-out sensors
full ASCII keyboard with numeric pad and function/control keys

MAGNETIC DISK DRIVE SYSTEMS (flying heads, one/surface)

Model	DEC RK 05	Plessey PM-DS/11B	Plessey PM-DS 11/300C
medium	IBM 2315 cartridge	IBM 2315 cartridge + fixed disk	CDC 9883 disk pack
capacity/drive (megabytes formatted)	2.46	4.92	253.6
average access time (ms)	70	70	38
transfer time (microsec/16-bit word)	11.1	11.1	1.7
data surfaces/drive	2	4 (2 fixed)	19
tracks/surface	200	200	815
sectors/track	12	12	32
bytes/sector	512	512	512
bit density/in. (inside track)	2040	2040	6036
rotation speed (rev/s)	25	25	60
DEC controller (or equivalent)	RK-11D	RK-11D	RH-11
buffer size (words)	6	6	128

MINICOMPUTERS (DEC 16-bit word PDP-11's)

Model	PDP-11/05	PDP-11/35	PDP-11/60
memory size (K-words)	28	124	124
memory type	core	core	MOS
memory cycle time (nanosec)	950	950	700
cache memory cycle time (nanosec)	no cache	340	340
branch instruction time, cache hit (nanosec)	2500	1100	850
integer arithmetic instruction set	software	EIS	EIS
floating-point arithmetic instruction set	software	FIS	FP-11C (integral)
memory-management program modes	no mem. mgmt.	kernel/user	kernel/user

All the above computers have over 400 basic instructions (including eight address modes), eight general registers, vectored interrupts, console, bootstrap, line clock, and battery-powered calendar clock.

APPENDIX B

GLOSSARY

ALPHANUMERIC. From the standard set of English upper- and lower-case alphabetic characters and arabic numerals.

ASCII. American Standard Code for Information Interchange.

BEGIN. A special function that causes the TXT program to return to the beginning of a file and display the first page.

BUFFER. A temporary file-storage area in memory.

CHARACTER. Any member of the set of ASCII characters, of which there are 128, including alphanumeric, punctuation, and control characters.

CHARACTER STRING. A sequence of characters from the standard 96-character ASCII set normally found on the keyboard, excluding two-key control characters.

COMMAND. A program instruction invoked by typing one or more keyboard characters, when program is in command level.

COMPOSITION. The construction or layout of a page of text.

DATA. In TXT, any collection of ASCII characters.

DELETE. (a) To remove data from the screen (and thus from computer memory, if screen page is transmitted).

(b) To remove a file name from the disk directory listing, thus freeing the file space.

DIRECTORY. A listing of files contained on a designated permanent storage device.

DISK. Any of the permanent storage media for files. Program TXT is invariant with respect to the particular disk or mode of operation.

EDIT. Text modification performed at the terminal by the operator.

EMBEDDED COMMANDS. Special characters, placed in a text file, that act as non-printing formatting controls. They appear on the screen in a form that follows the key strokes, but upon hardcopy printout they perform their control function without appearing directly on the copy.

ERASE. An operation that eliminates the characters that follow the cursor on the given line or page.

EXIT. Termination of a text-editing session, which sends all current screen text to computer memory, and all text in computer memory to a disk file.

FILE. A data set on a disk, accessed by its specific name. For TXT, a named listing of the sequence of ASCII characters generated by keystroke on the keyboard during the Edit mode of TXT program operation. The DESTINATION file name will be assigned to the created or edited text. The SOURCE file name designates the file containing text to be edited and is nonexistent if a new text file is being created.

FILE SWITCH. Changeover on the screen from presentation of one file to presentation of another file during an editing session.

INPUT. The process of generating or changing a file from the keyboard.

INSERT. To place a character on the screen (and thus in computer memory) at the location of the cursor.

I/O. Input/Output of data to/from designated device (terminal, computer, disk, printer).

JUSTIFICATION (LEFT AND RIGHT). The process of composing output in such a manner as to form uniform vertical margins. In left justification, all words start on the left margin, with the lines terminated at word or hyphen boundaries. In flush-right justification, the last characters on all lines are vertically aligned at the right margin.

MERGING. The process of combining the contents of two files.

MONITOR (SOFTWARE). The program that controls the applications software used in the computer and directs file and program traffic for multiple users.

NAME. The designation that distinguishes a file from other files.

OUTPUT. Presentation of the results of an editing session, in soft (screen) or hard (printer) copy.

PAGE. On the screen, the maximum number of visible lines--24 on the ADM videoterminals.

PROPORTIONAL SPACING. Adjustment of the spacing between characters to permit output with higher density, more uniform appearance, and/or flush right-hand justification.

SCROLL. To move the screen page forward or backward through the file (up or down on the video screen).

SYMBOL. A visual representation of a keyboard stroke (on the screen or printer), not necessarily in the form illustrated on the keyboard.

TERMINAL. A device used in I/O for the program, particularly the video unit with keyboard.

TEXT. An organized arrangement of any printable ASCII characters or other special symbols.

TEXT PROCESSING. The I/O and editing operations used on any printable characters or symbols.

VERSION. An (octal) number that is incremented for each file upon EXIT from an editing session. The largest number represents the latest edited version.

VOLUME. A memory medium (usually a disk) in which a collection of files is stored.

WORD PROCESSING. Same as text processing, except that word processing has historically been confined primarily to standard ASCII codes (i.e., not applied to text in which ASCII control codes are used to format and to designate non-standard characters).

APPENDIX C

ASCII CODE LISTING OF ALL CHARACTERS

Control Characters

(See footnotes for explanation of column headings and special symbols)

ASCII code	KB2 & KB42	VT2	VT42	Function in PDP-11 TXT program Edit Mode	TTY Char.	Comment
1	c^A	A	S_H	Special function	SOH	
2	c^B	SP1	S_X		STX	
3	c^C	SP2	E_X		ETX	
4	c^D	D	E_T	Character delete	EOT	
5	c^E	E	E_Q	Blank entry	ENQ	
6	c^F	F	A_K		ACK	
7	c^G	BEL	B_L		BEL	Bell
8	c^H	←	B_S	Cursor left	BS	
9	c^I	→	H_T	Tab	HT	Tab
10	c^J	SP3	L_F	Cursor down	LF	Line feed
11	c^K	↑	V_T	Cursor up	VT	Vertical
12	c^L	→	F_F	Cursor right	FF	Form feed
13	c^M	SP4	C_R	Return	CR	Carriage return
14	c^N	SP5	S_O	Format control	SO	
15	c^O	SP6	S_I	[Reserved]	SI	
16	c^P	0	D_L	Line terminator	DLE	
17	c^Q	1	D_1	Line scroll	DC1	
18	c^R	2	D_2		DC2	
19	c^S	3	D_3		DC3	
20	c^T	4	D_4	Format control	DC4	
21	c^U	U	N_K	Underline	NAK	
22	c^V	V	S_Y	Vertical column	SYN	
23	c^W	W	E_B	Center text	ETB	
24	c^X	X	C_N	Page feed	CAN	
25	c^Y	Y	E_M	Half-line down	EM	
26	c^Z	Z	S_B	[Reserved]	SUB	
27	c^[	SP7	E_C	[Escape]	ESC	Escape
28	c^\	SP8	F_S	3rd char. set	FS	
29	c^↑	↑	G_S		GS	
30	c^]	]	R_S		RS	
31	HOME	SP9	U_S	Cursor home	US	

SP1 solid triangle facing right
SP2 solid triangle facing left
SP3 underlined down arrow
SP4 shaded triangle facing left
SP5 underlined double up arrow
SP6 underlined double down arrow
SP7 shaded E
SP8 underlined backslash
SP9 underlined left-slanted arrow

KB2 Keyboard, ADM2
KB42 Keyboard, ADM42
VT2 Visual display, ADM2
VT42 Visual display, ADM42
BEL Audible tone
TTY Teletype notation

Special Characters

	ASCII code	KB2 & KB42	VT2 & VT42	ASCII V [1]	ASCII A A6 WP	A-G/M [2]	A-G/M Cv [3]	V-G/M [4] Sym-bol	V-G/M [4] Comment	V-G/M [5] Sym-bol	V-G/M [5] Comment
	32	Sp	Sp	Sp	¢	≈		Sp	space		
	33	!	!	!	!	π		°	up. left °	é	e acute
	34	"	"	"	"			°	degree	¨	umlaut **
	35	#	#	#	#	{		■	solid ■	¶	paragraph
	36	$	$	$	$	}		°	up.right °	§	section
	37	%	%	%	%	Å		Å	Ångstrom	ø	sl'd zero
	38	&	&	&	&	∫		∫		a	closed a
	39	'	'	'	'	÷		÷			
	40	(	(	(	(	(		(			
	41	)	)	)	)	µ		)			
	42	*	*	*	*			∘	mid.left ∘	©	copyright
	43	+	+	+	+	±		±			
	44	,	,	,	,	'		∘	mid. rt. ∘		
	45	-	-	-	-	/		∓		¯	o'score **
	46	.	.	.	.	~		∘	low left ∘	·	ov.dot **
	47	/	/	/	/	/		/			
*	48	0	0	0	0	0		0	Superscript		
	49	1	1	1	1	1		1	or		
	50	2	2	2	2	2		2	subscript		
	51	3	3	3	3	3		3	·		
	52	4	4	4	4	4		4	·		
	53	5	5	5	5	5		5	·		
	54	6	6	6	6	6		6	·		
	55	7	7	7	7	7		7	·		
	56	8	8	8	8	8		8	·		
	57	9	9	9	9	9		9	·		
	58	:	:	:	:			∘	low rt. ∘		
	59	;	;	;	;			‡	dagger	®	registered
	60	<	<	<	<	∏		←			
	61	=	=	=	=	≡		≡			
	62	>	>	>	>	→		→			
	63	?	?	?	?	∫		∫			
	64	@	@	@	@	∑		∑			
	65	A	A	A	A	∇		∇	del		
	66	B	B	B	B	∞		∞	infinity		
	67	C	C	C	C	Ψ		Φ			
	68	D	D	D	D	Φ	Δ	Δ	DELTA		
	69	E	E	E	E		▲	▲			
	70	F	F	F	F	≤	Φ	Φ	PHI		
	71	G	G	G	G	Λ	Γ	Γ	GAMMA		
	72	H	H	H	H	↔	⊛	⊛	THETA		
	73	I	I	I	I		‖	‖			
	74	J	J	J	J	≥		≥			
	75	K	K	K	K	×	≤	≤			
	76	L	L	L	L	≅	Λ	Λ	LAMBDA		
	77	M	M	M	M	≮		≮			
	78	N	N	N	N	≯		≯			
	79	O	O	O	O	ϕ	Ω	Ω	OMEGA		
	80	P	P	P	P	µ	∏	∏	PI		
	81	Q	Q	Q	Q	Γ	≅	≅			
	82	R	R	R	R	⊛	↔	↔			
	83	S	S	S	S	Σ		Σ	SIGMA		
	84	T	T	T	T	∝		∝	prop. to		
	85	U	U	U	U	Ξ	ϒ	ϒ	UPSILON		
	86	V	V	V	V		×	×	times		
	87	W	W	W	W	Δ	Ψ	Ψ	PSI		
	88	X	X	X	X	Ω	Ξ	Ξ	XI		
	89	Y	Y	Y	Y	ϒ	—	—	dash		
	90	Z	Z	Z	Z	✓		✓	check		

ASCII code	KB2 & KB42	VT2 & VT42	ASCII V [1]	ASCII A A6 WP	A-G/M [2]	A-G/M Cv [3]	V-G/M [4] Sym-bol	V-G/M [4] Comment	V-G/M [5] Sym-bol	V-G/M [5] Comment
91	[	[	[	[	[		[			
92	\	\	\	®	\		\			
93	]	]	]	]	]		]			
94	^	^	↑	^ ©	↑		↑	up arrow	^	c'flex **
95	_	_	←	_	NC	NC	NC	null char		
* 96	`	`	`	` °			`	grave	`	grave **
97	a	a	a	a	α		α	alpha		
98	b	b	b	b	β		β	beta		
99	c	c	c	c	ψ	χ	χ	chi		
100	d	d	d	d	φ	δ	δ	delta		
101	e	e	e	e	∈		∈	epsilon		
102	f	f	f	f	~	φ	φ	phi		
103	g	g	g	g	λ	γ	γ	gamma		
104	h	h	h	h	η	θ	θ	theta		
105	i	i	i	i	∫	ℓ	ℓ	script l		
106	j	j	j	j	≠		≠			
107	k	k	k	k	κ		κ	kappa		
108	l	l	l	l	ω	λ	λ	lambda		
109	m	m	m	m	)	μ	μ	mu		
110	n	n	n	n	ν		ν	nu		
111	o	o	o	o	o	ω	ω	omega		
112	p	p	p	p	ρ	π	π	pi		
113	q	q	q	q	γ		∂	partial der.		
114	r	r	r	r	θ	ρ	ρ	rho		
115	s	s	s	s	σ		σ	sigma		
116	t	t	t	t	τ		τ	tau		
117	u	u	u	u	ξ	υ	υ	upsilon		
118	v	v	v	v	↓		↓			
119	w	w	w	w	δ	ψ	ψ	psi		
120	x	x	x	x	χ	ξ	ξ	xi		
121	y	y	y	y	υ	η	η	eta		
122	z	z	z	z	ζ		ζ	zeta		
123	{	{	{	{ §	{		{			
124	|	|	|	| ¶	|		|		|	caret
125	}	}	}	} ‡	}		}			
126	~	~	~	~ TM	≈		≈		~	hi ~ **
127	RUB			- dbl						

KB2 Keyboard, ADM2
KB42 Keyboard, ADM42
VT2 Vis.display, ADM2
VT42 Vis.display, ADM42

A Agile typewriter
V Versatec line printer
G/M Greek/math
Cv Converted

Sp Space
dbl Double underline
TM Superscript TM
NC Null character

A6 Agile ASCII 96 PRESTIGE ELITE 12-character wheel
WP Agile SEQUENCE WP ELITE 12-character wheel

* ASCII code 48 is a zero with a slash through it on KB2.
* ASCII code 96 appears as a shoe facing right on V-ASCII.
** To appear above the succeeding character. Overprint not yet implemented.

[1] Standard ASCII character set for Versatec line printer.
[2] Standard for Agile G/M character wheel.
[3] Converted to make A-G/M compatible with V-G/M. Invoked by \ followed by keyboard character. [Not yet implemented.]
[4] First special character set for Versatec (plotted). Invoked by \ followed by keyboard character.
[5] Second special character set for Versatec (plotted). Invoked by $_c$\ followed by keyboard character.

APPENDIX D

CROSS-REFERENCE TABLE OF SPECIAL CHARACTERS
(Greek alphabet excluded)

Sym.	Key	Comment
(	(	
)	)	
(	\(	
)	\)	
[	[	
[	\[	
]	]	
]	\]	
{	{	
}	}	
{	\{	
}	\}	
<	<	
>	>	
≤	\K	
≥	\J	
‹	\M	
›	\N	
^	\L	
^	c\	key is c\ followed by a caret
|	c\	c\ followed by vertical line should print a caret
↑	\	backslash followed by caret should print up-arrow
↓	\v	
↔	\R	
←	\<	
→	\>	
|	|	
|	\	
‖	\I	
/	/	
/	\/	
\	\\	two backslashes needed to print a backslash

c\ Control-backslash.

Sym.	Key	Comment
—	\Y	dash
-	-	hyphen; minus
CH	`	cond. hyphen
+	+	
±	\+	
∓	\-	
×	\V	times
÷	\'	
=	=	
≡	\=	
≢	\j	
≅	\Q	
≈	\~	
~	~	
∝	\T	proportional to
∫	\?	
∫	\&	
Σ	\S	
Σ	\@	
∇	\A	del
∂	\q	partial derivative
∞	\B	infinity
▲	\E	solid delta
°	\"	degree
°	\!	upper left °
°	\$	upper right °
°	*	mid. left °
°	\,	mid. right °
°	\.	lower left °
°	\:	lower right °

CH Conditional hyphen. The grave can serve as a "space eater" in that it is ignored on printout. Exception: If embedded in a word, on justified output it will print as a hyphen if the word breaks at the end of a line.

Sym.	Key	Comment
$	$	
¢	\C	
#	#	
%	%	
&	&	
‡	\;	dagger
Ø	c\%	zero
'	'	apostrophe
`	\`	grave
	c\`	hi grave
˙	c\.	overdot
¨	c\"	umlaut
¯	c\-	overscore
^	c\	circumflex (key is c\ followed by caret)
~	c\~	hi tilde
✓	\Z	check
▒	\#	solid block
[blank]	_	dummy character
¶	c\#	paragraph
§	c\$	section
@	@	at
©	c*	copyright
®	c\;	registered
ø	c\%	slashed zero
ɑ	c\&	closed a
Å	\%	Ångstrom
ç	c\,	c cedilla
é	c\!	e acute
è	c\(	e grave
ê	c\)	e circumflex
ℓ	\i	script l

APPENDIX E

SEQUENTIAL FILE PRINTING AND HEADERS

COMMAND FILES

TXT has a feature that permits one or a sequence of files to be printed by a single command. Its main use is to produce a single document that has been broken for convenience into a number of files. To use this facility, one first creates a "command file" that contains the names of the file or files to be printed. The command file can have any name, but it is convenient to use the extension .CMD as a reminder that this is a command file; e.g., the command file could have the name DOCS.CMD.

To do the printing, the OUTPUT mode is entered (by typing O in response to the main command string), and the command file, with its name prefixed by the symbol "@" (e.g., @DOCS.CMD), is named as the one to be printed. The pages will automatically be numbered sequentially.

Example of a Simple TXT command file

```
INTRO.P1
FILE.1
FILE.2
CHAP.3
APPEN.A1
APPEN.A2
```

See Appendix D for notes on managing the electrostatic printer for sequential output.

HEADER OPTIONS

Several formatting options can be used in the command files: Page headers can be specified, with or without the date and page numbers, and the number of lines per page and characters per line can be specified. These features are invoked by incorporating in the command file the desired instructions from the following list:

.H	(Header)	Flags the following text to be used as a header for next files.
.E	(Even)	Flags the header to be used on even-numbered pages only.
.O	(Odd)	Flags the header to be used on odd-numbered pages only.
.Bn		Will end the header with n blank lines.
.Ln		Sets the number of lines per page to n.
.Cn		Sets the number of characters per line to n.
__-___-__		Date field (e.g., 29-JAN-80).
___		Page field (e.g., 27).

Headers can be applied with a command file, even if there is only one file to be named.

Example of a TXT Command File	Note
.H	1
PAGE HEADER	2
.B4	3
FILE.1	4
FILE.2	5
.H	
	6
.B0	7
FILE.3	8
.H	
NEW HEADER, for File 4 __-___-__	9
SubHeading Page __.	10
.B2	
.C62.L55	11
FILE.4	
.H.E	12
HEADER FOR EVEN-NUMBERED PAGES	
For File 5	13
.B3	
.H.O	14
HEADER FOR ODD-NUMBERED PAGES ____	15
For File 5	
.B4	
FILE.5	16

Explanation

1. The .H specifies the next line to be a header.

2. Each page will be headed "PAGE HEADER" until superseded.

3. The .B4 calls for four blank lines under the header.

4. The first file in the list is named "FILE.1."

5. The second file, FILE.2, will get the same page headings as FILE.1.

6. The blank line following .H will cancel the headers for FILE.3.

7. A .Bn entry is needed after each header. In this case, zero blank lines are desired.

8. The final entry for each file is the name of the file.

9. A two-line header is specified for FILE.4. The date will appear on the first line, called out by the 2-3-2 pattern of low dashes (underline key on a conventional typewriter) and hyphens.

10. In this file, the page number is wanted as part of the header. It will appear on the second line. (The page numbers for the first three files will appear at the bottoms of the pages.) The sequence of two low dashes causes the page number to be printed. The units digit of the page number will appear at the position of the rightmost low dash (of which there must be at least 2). If the number of digits exceeds the number of dashes, characters to the left of the dashes will be overwritten as necessary.

11. FILE.4 is to be printed with 62 characters per line, 55 lines per page. This will remain in effect for subsequent files until superseded.

12. The .H.E entry specifies that the header that follows is for even-numbered pages only.

13. A two-line header for FILE.5. With no page number requested for even pages, but with the request for odd pages (below), no page number will appear on the even-numbered pages.

14. The .H.O entry indicates that the header that follows is for odd-numbered pages.

15. The odd page numbers for FILE.5 will appear on the first line of the header, at the right end of the string of low dashes.

16. The final entry in the list is the name of the final file.

```
TXTG.BEG
.H.E
      TEXT PROCESSING
.B2
.L63
.H.0
                                                    INTRODUCTION
.B2
TXTG.0
.H.0
                                                    SW/HW SYSTEM
.B2
TXTG.1
.H.0
                                                    PROGRAM OPERATIONS
.B2
TXTG.2
.H.0
                                                    HARDWARE CHARACTERISTICS
.B2
TXTG.A
.H.0
                                                    GLOSSARY
.B2
TXTG.B
.L66
.H

.B0
TXTG.C
.L63
.H.E
      TEXT PROCESSING
.B2
.H.0
                                                    SPECIAL CHARACTERS
.B2
TXTG.D
.H.0
                                                    SEQUENTIAL FILES AND HEADERS
.B2
TXTG.E
.H.0
                                                    COMMAND FILE
.B2
TXTG.CMD
.H.0
                                                    REFERENCES
.B2
TXTG.REF
```

REFERENCES

[1] For example: M. Leavitt and C. M. Lederer, "Overview of IRATE--Interactive Retrieval and Text Editor," Lawrence Berkeley Laboratory report LBL-4607 (1975).

[2] A. De Volpi, "Proliferation, Plutonium, and Policy," Pergamon Press, NYC (May 1979).

[3] A. De Volpi, Intra-Laboratory Memo, (8 Nov. 1974).

[4] M. Walker, "Word Processing Transforms Office Paperwork Routine," Electronics, p. 89 (12 June 1975).

[5] See: Minicomputer News, 4(21):2 (9 Nov. 1978).

[6] P.I. Berman, "Survey of Computer-associated Writing and Editing Systems," North Atlantic Treaty Organization, Advisory Group for Aerospace Research and Development, report AGARD-AG-229 (1977).

[7] SCRIPT, an ANL word-processing program (Applied Mathematics Division, Argonne National Laboratory).

[8] J.H. Siegman, "Tools for Typing Technical Text," reprinted in The Seybold Report on Word Processing, Vol. 2, No. 11 (Dec. 1979).

Distribution for ANL-80-52

Internal:

W. E. Massey
E. S. Beckjord
R. Avery
L. Burris
D. W. Cissel
S. A. Davis
B. R. T. Frost
L. G. LeSage
R. A. Lewis
R. J. Royston
R. J. Teunis
C. E. Till
R. S. Zeno
J. F. Marchaterre
M. R. Fenrick (2)
C. L. Fink (2)
G. E. Marsh
J. P. Regis
E. A. Rhodes (2)
G. S. Stanford (2)
R. R. Stewart
G. Ridges
A. De Volpi (65)
A. B. Krisciunas
ANL Contract File
ANL Libraries
TIS Files (6)

External:

DOE-TIC, for distribution per UC-32 (183)
Manager, Chicago Operations and Regional Office, DOE
Chief, Office of Patent Counsel, DOE-CORO
Director, Technology Management, DOE-CORO
Director, DOE-RRT
Acting Chief, Safety and Physics Branch, DOE-RRT
Acting Chief, Safety Section, DOE-RRT
President, Argonne Universities Association
Reactor Analysis and Safety Division Review Committee:
S. Baron, Burns and Roe, Inc.
J. R. Dietrich, West Hartford, Conn.
L. C. Hebel, Xerox Corp., Palo Alto
W. Kerr, U. Michigan
S. Levine, NUS Corp., Rockville
S. Levy, S. Levy, Inc., Campbell, Calif.
D. Okrent, U. California, Los Angeles