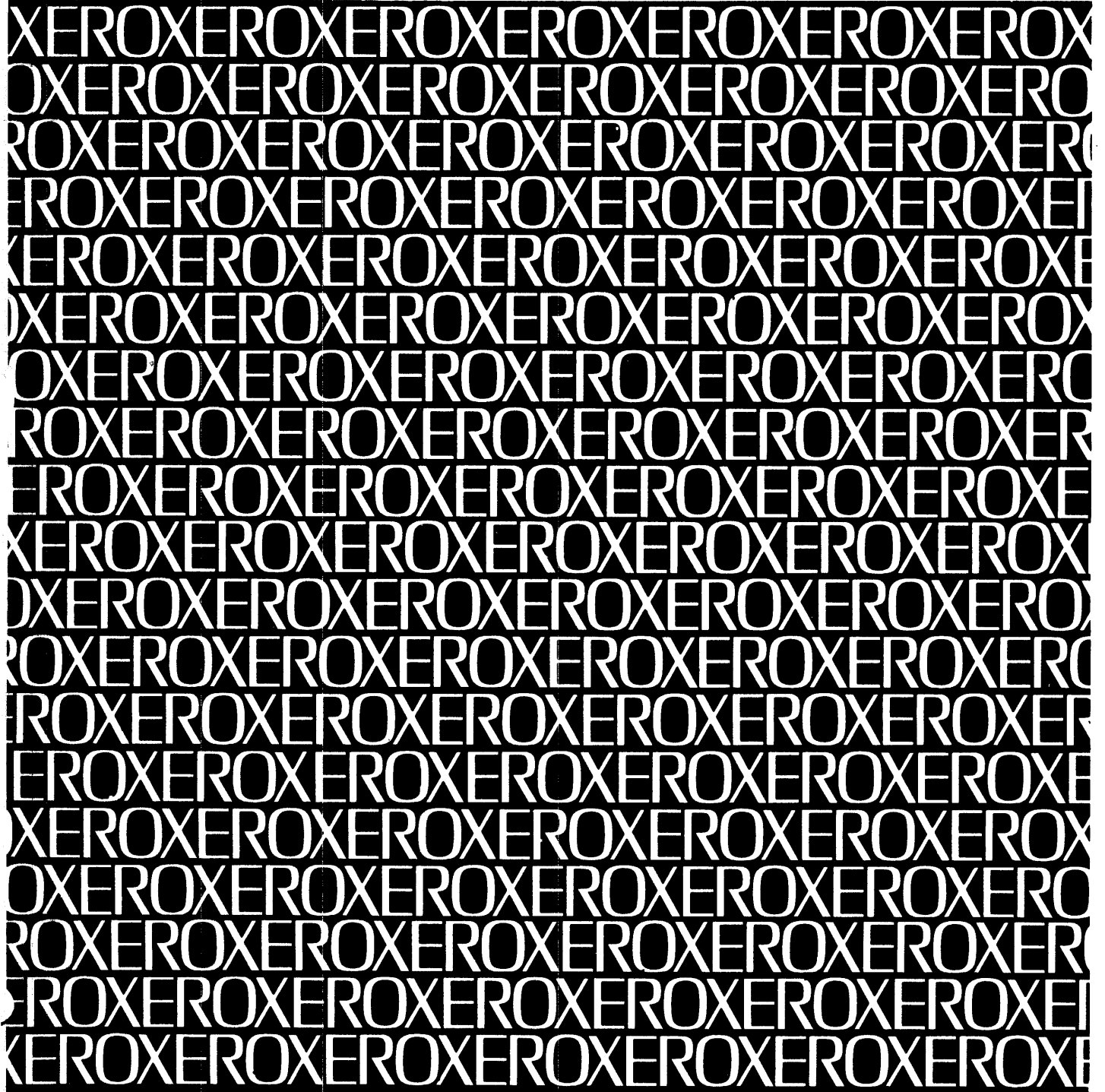


Xerox TEXT

Xerox 560 and Sigma 6/7/9 Computers

Language and Operations
Reference Manual



XEROX

Xerox TEXT

Xerox 560 and Sigma 6/7/9 Computers

Language and Operations Reference Manual

90 18 51C

May 1976

REVISION

This publication documents the B00 version of Xerox TEXT for Xerox 560 and Sigma 6/7/9 computers.

RELATED PUBLICATIONS

<u>Title</u>	<u>Publication No.</u>
Xerox Sigma 6 Computer/Reference Manual	90 17 13
Xerox Sigma 7 Computer/Reference Manual	90 09 50
Xerox Sigma 9 Computer/Reference Manual	90 17 33
Xerox 560 Computer/Reference Manual	90 30 76
Xerox Control Program-Five (CP-V)/TS Reference Manual	90 09 07
Xerox Control Program-Five (CP-V)/TS User's Guide	90 16 92

The specifications of the software system described in this publication are subject to change without notice. The availability or performance of some features may depend on a specific configuration of equipment such as additional tape units or larger memory. Customers should consult their Xerox sales representative for details.

CONTENTS

<p>PREFACE AND SUMMARY OF TECHNICAL FEATURES v</p> <p>GLOSSARY vi</p> <p>1. INTRODUCTION 1</p> <p style="padding-left: 20px;">Advantages of TEXT _____ 1</p> <p style="padding-left: 20px;">Applications of TEXT _____ 1</p> <p style="padding-left: 20px;">Organization of Manual _____ 2</p> <p>2. OVERVIEW 3</p> <p style="padding-left: 20px;">TEXT System _____ 3</p> <p style="padding-left: 20px;">Commands _____ 3</p> <p style="padding-left: 40px;">System Commands _____ 3</p> <p style="padding-left: 40px;">Embedded Commands _____ 3</p> <p style="padding-left: 40px;">Command Format _____ 3</p> <p style="padding-left: 20px;">Document Storage _____ 4</p> <p style="padding-left: 40px;">Working Storage _____ 5</p> <p style="padding-left: 40px;">Permanent Storage _____ 5</p> <p style="padding-left: 20px;">How to Use TEXT _____ 5</p> <p style="padding-left: 40px;">Entering a Simple Document _____ 5</p> <p style="padding-left: 40px;">Printing a Simple Document _____ 7</p> <p style="padding-left: 20px;">Summary _____ 8</p> <p>3. TERMINAL OPERATIONS 9</p> <p style="padding-left: 20px;">Description of Terminal _____ 9</p> <p style="padding-left: 20px;">Terminal Start/Shutdown Procedures _____ 9</p> <p style="padding-left: 40px;">Start Procedure _____ 9</p> <p style="padding-left: 40px;">Shutdown Procedure _____ 11</p> <p>4. BASIC TEXT COMMANDS 12</p> <p style="padding-left: 20px;">Document Formatting for Printout _____ 13</p> <p style="padding-left: 40px;">Formatted Mode _____ 13</p> <p style="padding-left: 40px;">Unformatted Mode _____ 14</p> <p style="padding-left: 40px;">Line Width, Page Depth, and Tab Settings _____ 15</p> <p style="padding-left: 40px;">Control of Paragraphs _____ 17</p> <p style="padding-left: 40px;">Widow Suppression _____ 17</p> <p style="padding-left: 20px;">Building a New Document _____ 18</p> <p style="padding-left: 40px;">Clearing Working Storage _____ 18</p> <p style="padding-left: 40px;">Setting Up the Format for Printout _____ 18</p> <p style="padding-left: 40px;">Entering a Document _____ 19</p> <p style="padding-left: 40px;">Corrections While Entering a Line _____ 19</p> <p style="padding-left: 20px;">Printing a Document _____ 20</p> <p style="padding-left: 40px;">Printing Copies with Line Numbers _____ 21</p> <p style="padding-left: 20px;">Editing a Document _____ 23</p> <p style="padding-left: 40px;">Line Corrections _____ 23</p> <p style="padding-left: 40px;">Line Erasures _____ 26</p> <p style="padding-left: 40px;">Inserting/Moving Lines _____ 27</p> <p style="padding-left: 20px;">Storing and Retrieving a Document _____ 31</p> <p style="padding-left: 40px;">Storing a Document _____ 31</p> <p style="padding-left: 40px;">Retrieving a Document _____ 34</p>	<p style="padding-left: 20px;">Deleting a Document _____ 36</p> <p style="padding-left: 20px;">Clearing Working Storage _____ 36</p> <p style="padding-left: 20px;">Ending a Terminal Session _____ 36</p> <p>5. ADVANCED COMMANDS 38</p> <p style="padding-left: 20px;">Embedded Commands _____ 38</p> <p style="padding-left: 40px;">Format-Mode Commands _____ 38</p> <p style="padding-left: 40px;">Miscellaneous Embedded Commands _____ 48</p> <p style="padding-left: 20px;">System Commands _____ 55</p> <p style="padding-left: 40px;">Width Command _____ 55</p> <p style="padding-left: 40px;">Editing Commands _____ 56</p> <p style="padding-left: 40px;">Printout Commands _____ 60</p> <p style="padding-left: 40px;">Storage and Retrieval Commands _____ 64</p> <p style="padding-left: 40px;">Administrative Commands _____ 69</p> <p style="padding-left: 40px;">Index Command _____ 70</p> <p style="text-align: right;">INDEX 101</p> <p style="text-align: center;">APPENDIXES</p> <p>A. SUMMARY OF TEXT COMMANDS 83</p> <p>B. SUMMARY OF TEXT MESSAGES 93</p> <p>C. TERMINAL REQUIREMENTS 100</p> <p style="text-align: center;">FIGURES</p> <p>1. Entering a Simple Document _____ 6</p> <p>2. Printout of a Simple Document _____ 7</p> <p>3. Keyboard of Typical Terminal _____ 9</p> <p>4. As-Entered Printout with Line Numbers _____ 22</p> <p>5. Formatted Printout with Line Numbers _____ 23</p> <p>6. Store Command with Various Options _____ 33</p> <p>7. Entering Lines in Centered Mode _____ 39</p> <p>8. Printout of Lines Entered in Centered Mode _____ 39</p> <p>9. Entry of a Table in Unformatted Mode _____ 43</p> <p>10. Table with Three Rows and Three Columns _____ 43</p> <p>11. Entry of a Table in Table Mode, by Rows _____ 44</p> <p>12. TEXT Printout of Figure 11 _____ 44</p> <p>13. TEXT Printout of Figure 11 with Different Format Specification _____ 45</p> <p>14. Entry of a Table with Unformatted Columns and Blank Entries _____ 46</p> <p>15. TEXT Printout of Figure 14 _____ 46</p> <p>16. Entry of a Table in Table Mode, by Columns _____ 47</p> <p>17. Entry of a Table that is to be Printed on One Page _____ 48</p> <p>18. Entry of Document Containing a Stop Code _____ 51</p> <p>19. TEXT Printout of Figure 18 _____ 52</p> <p>20. Entry and Printout of Special Characters _____ 55</p> <p>21. Printout with Justified Right Margin _____ 61</p>
--	--

TABLES

1.	Sample Paragraphs in Formatted Mode _____	18
2.	Correcting Lines Via the Backspace Procedure _____	20
3.	Page Codes for Headings/Footings _____	40
4.	Placement of Heading/Footing on Page _____	42
5.	Variations of the Block Callout Command _____	51
6.	Variations of the Skip Command _____	53
7.	Use of Relative Line Numbers in TEXT Commands _____	57
8.	Variations of the Find Command _____	59
A-1.	Embedded Commands _____	84
A-2.	System Commands _____	87
B-1.	TEXT Messages _____	93

PREFACE AND SUMMARY OF TECHNICAL FEATURES

TEXT consists of a central Xerox 560 or Sigma 6, 7, or 9 computer operating under Control Program-Five (CP-V), the TEXT program, and one or more remote typewriter terminals connected to the computer via telephone lines. The terminal used is compatible with the IBM 2741 Communications Terminal, having

- Uppercase/lowercase capability.
- Interchangeable typing elements.
- Up to 150 words per minute printing speed.

Salient technical features of TEXT include the following:

- Input is from a typewriter terminal or equivalent; output is to a typewriter terminal or line printer.
- Documents are created and edited in "working storage". Documents may be saved in "permanent storage" on disk files or on magnetic tape.
- All input is considered as text and entered into working storage unless it is preceded by an ATTN key. Input preceded by ATTN is a command.
- Commands may immediately affect the system status or may be embedded in the text for future action. All embedded commands begin with the letter t.
- Each newly entered line of text is assigned an integer line number reflecting its position in the document and may be referenced at any time by that line number.
- A line of text (including embedded commands) may be edited, erased, or rewritten. A set of lines may be erased or moved, and new lines may be inserted at any point.
- Specific text lines may be located by content as well as by line number.
- The "current line pointer" (CLP) may be set in various ways to point to any line in the document. The CLP normally points to the last line in the document.
- Most of the commands may reference lines of text by using relative line numbers of the form " \pm number". The offset is taken relative to the CLP.
- When lines are erased, their line numbers remain unassigned. When lines are inserted, they are given incremental line numbers distributed between the preceding and following line numbers. The original line numbers do not change unless explicitly renumbered using the "store" command.
- Strings of text may be "tagged" for more efficient access. By tagging all occurrences of a specific reference, it is possible to locate and change them easily.
- Blocks of text, such as figures and tables, may be named and automatically repositioned on output.
- Tables may be created under a special "table" mode with optional columnar formatting.
- Text may be printed under any one of four formatting modes: formatted, unformatted, table, and form letter.
- Special characters are easily handled by defining substitute character sequences.
- Two levels of document security are provided.
- Blank strings may be inserted at any point with an optional "stop" on printout.

GLOSSARY

- { } in format of command items placed one under the other between braces require the selection of one of the items.
- [] in format of command an item between brackets is optional.
- () in format of command an item in parentheses must be replaced by a user-chosen number, word, or phrase.
- account number a unique number assigned by the computer center to each user or department. This number is used at log-on time to gain access to TEXT.
- ATTN key a special key on the typewriter terminal keyboard used to alert TEXT that the user is about to issue a command.
- character any letter, number, or symbol (including a space) that can be typed at the typewriter terminal.
- character string any combination of letters, numbers, or symbols (including spaces). Literally a string (or series) of successive characters. A string may be a few letters, a word, several words, or even one or more sentences.
- command an instruction from the user that tells the computer what to do. TEXT has two types of commands: system commands and embedded commands. System commands are instructions to the computer to perform an immediate operation such as printing a line or storing a document. Embedded commands do not cause immediate computer action, but are embedded in the text for action during subsequent printing or editing of the document. Embedded commands always start with a "t" and usually specify formatting or indexing information.
- computer the Xerox 560 or Sigma 6, 7, or 9 computer on which the TEXT program runs. The TEXT user communicates with the computer via the typewriter terminal.
- computer center the location of the Xerox computer on which TEXT runs. The computer center may be at any distance from the typewriter terminal, since the computer and the terminal are connected by a telephone line.
- computer operator the person who controls and operates the Xerox computer at the computer center, and who performs any functions that are required at the computer (mounting magnetic tapes, etc.). Normally the computer operator has no contact with the TEXT user.
- document any text manipulated or used as a single entity by the TEXT system. A document usually consists of several lines, and may be complete in itself or may be a part of a larger document.
- editing the process of changing a document. Editing may include correcting spelling errors, adding or erasing lines of text or parts of lines, and rearranging lines.
- embedded command see "command".
- entering a document the process of typing the lines of a document at the terminal. Called "entering" because the lines are "entered" into computer storage as the user types them. Also see "typing".
- font an assortment of type, all of one size and style.
- line number a unique number assigned by TEXT to each line of a document as it is entered, so that it may be referred to later for editing purposes.
- log-on procedure the procedure that must be followed by the user to gain access to the TEXT system; it consists of supplying a valid sequence of code words, including an account number and sometimes a password.
- password a keyword that prevents unauthorized persons from accessing a TEXT user's account or documents. The log-on password prevents someone else from logging on in the user's account, and the document password prevents someone else from copying and deleting the user's documents.
- permanent storage one of the two areas of the computer where documents are stored. Documents are saved here for future use; a document can never be worked on while it is in permanent storage. Compare with "working storage".

phrase any combination of characters (including spaces). A phrase may be a few letters or one or more words.

printout anything printed by TEXT at the terminal or at the line printer. Compare with "typing".

string see "character string".

system command see "command".

terminal the typewriter used to communicate with TEXT.

text any characters, lines, paragraphs, etc. entered into TEXT as part of a document.

TEXT the TEXT document handling system, which consists of a central Xerox computer, the TEXT program that operates on the computer, and one or more typewriter terminals. The TEXT system is frequently referred to in this manual simply as "TEXT".

typing anything typed at the terminal by the user. Data is "typed" at the terminal by the user (and usually "entered" into storage), and is "printed" at the terminal by TEXT.

Control Program-Five (CP-V) the master program that controls the central Xerox computer and enables it to run programs such as TEXT.

user the person using TEXT at the typewriter terminal.

working storage one of the two areas of the computer where documents are stored. Documents are stored here temporarily while they are being worked on. Compare with "permanent storage".

1. INTRODUCTION

Xerox TEXT is a computer system that facilitates composing, editing, and printing documents at a typewriter terminal. When disconnected from the computer, the terminal can be used like a Selectric® typewriter. Under computer control, it produces formatted, error-free copy at a typing speed of approximately 150 words per minute. The TEXT command language is logical and simple, and anyone who types can quickly learn to use it.

Advantages of TEXT

Major advantages of TEXT over manual methods are

1. Backspacing and overstriking corrects all typographical errors as they are being entered into the computer.
2. When words, lines, paragraphs, or sections are inserted or deleted, the computer automatically reformats the document for output after changes have been made (by consolidating, expanding, page renumbering, etc.).
3. The computer can convert unformatted typing to formatted printout — on the terminal typewriter or on a high-speed printer.
4. Once copy has been proofed and corrected, printout of the corrected copy is totally error-free.
5. Printout on the terminal typewriter is automatic, requiring no attention by the user except to change sheets of paper as necessary.
6. Formatting is simple. Different printout formats can be tried before settling on a final format.
7. Documents can be placed in permanent computer storage (disk or tape) and recalled in seconds to working storage for printout or editing.
8. Text locations are specified by line number or keyword. Line numbers are assigned automatically as lines are entered. Keywords are found by computer search.
9. Automatic formatting capabilities include
 - Margin justification
 - Centering
 - Printout of headings and/or footings
 - Page numbering
 - Tabulating
10. Indexes and tables of contents are generated automatically.
11. Form letters are typed error-free and can be personalized. That is, TEXT will stop at any point in a printout and allow the user to enter a name, address, etc.
12. Typing elements are interchangeable. Printout can be programmed to halt at a preselected spot for an exchange of typing elements if italics or other special fonts are required.

Applications of TEXT

TEXT is more efficient than manual copy preparation in any of the following situations:

1. In general office operations. Because it is so easy to correct mistakes, a secretary can produce error-free final copy at practically rough-draft speed.

® Registered trademark of International Business Machines Corporation.

2. With documents requiring several drafts. Unchanged portions need not be retyped or proofed. Text additions – words, lines, paragraphs – are automatically integrated into the text, providing a clean copy of the latest draft quickly and easily.
3. With documents requiring the integration of boilerplate (material that is used unchanged in many different documents), for example proposals. This can save countless hours of typing and proofreading, relieving deadline pressure.
4. In preparing or copying from lists and catalogs, especially those requiring frequent updates.
5. In document revisions. Without TEXT, extensive revisions are frequently more troublesome to carry out than producing an original document. Because of its ability to reformat and renumber, TEXT makes the job of revising a document easy.
6. In searching for words and phrases that need to be changed or eliminated. TEXT's efficiency cannot be approached by manual searches, either in terms of accuracy or time.

Organization of Manual

Both the beginner and the experienced user will find the appendixes of special value. Appendix A is a list of the complete set of commands and instructions that the user can initiate from the typewriter keyboard. Appendix B lists the messages the computer will generate in response to various occurrences. The information presented in consolidated form in these two appendixes is explained in detail throughout the manual as described below.

A functional approach to TEXT is given in Chapter 2. This chapter will serve the beginner as an introduction to TEXT and as a review to the more experienced user. A description of the terminal with instructions on how to use it is given in Chapter 3. Chapters 4 and 5 contain the available command repertoire, divided into two chapters to separate the basic from the more advanced operations. Only 20 simple commands are required to use TEXT efficiently in most applications. These are explained in detail in Chapter 4. Chapter 5 explains the more advanced commands which should not be attempted until the user feels comfortable with the basic ones of Chapter 4.

2. OVERVIEW

This chapter describes some of the basic terms used in TEXT and presents an overview of how TEXT is used, with a simple example. Succeeding chapters will show actual procedures to be followed by the terminal user in handling a document.

TEXT System

From the point of view of the terminal user, TEXT consists of a typewriter terminal connected by telephone to an unseen Xerox computer. The typewriter terminals are used to communicate to the computer the documents and commands typed by the user. When not connected to the computer, the terminal typewriter may be used as an ordinary electric typewriter.

Commands

Commands are instructions that tell TEXT what to do; all commands are typed by the user at the terminal. To alert TEXT that a command is about to be issued, the user strikes the ATTN key located in the upper right corner of the terminal keyboard (see Chapter 3 for a description of the keyboard). The user then enters the command by striking one or more keys, and completes the command by striking a RETURN key or an ATTN key, depending on the command. It is important to remember that all commands to TEXT must be preceded by the ATTN key; if this keystroke is omitted, TEXT assumes that the command is part of the document being entered, not a command. There are two types of TEXT commands: system commands and embedded commands.

System Commands

System commands are instructions to the TEXT system to perform an immediate operation such as print a line or store or retrieve a document.

Embedded Commands

Embedded commands do not require immediate action, but are stored with a document for action during subsequent printout. Embedded commands always start with a "t" and usually specify formatting information. Some specific uses of embedded commands include changing the format of a document as it is printed; tagging words or phrases so they can be easily located; defining blocks of words for insertion in the document; defining text that must be kept together on one page; skipping lines; etc.

Command Format

In most examples in this manual two forms of a command are shown. The first form shows the actual keyboard keys that the user is to type, and the second form shows what these keys print at the terminal. These two forms will often be different, since some keys cause different characters to be printed, and some keys cause nothing to be printed; for example, the ATTN key prints as an underline, and no character is printed for the RETURN or BACKSPACE key.

The following nomenclature is used in showing the first form, the command as the user should type it:

1. The $\text{\textcircled{ATTN}}$ symbol denotes striking the ATTN key.
2. The $\text{\textcircled{RET}}$ symbol denotes striking the RETURN key.
3. The $\text{\textcircled{BKSP}}$ symbol denotes striking the BACKSPACE key.
4. Parenthesized items indicate data that must be supplied by the user. The parentheses are not part of the actual command; that is, the parentheses should not be typed as part of the command.

Thus, for example, the command shown as

ATTNw;(lines)RET

means

1. Strike the ATTN key.
2. Type the letter w and a semicolon.
3. Substitute a value for "(lines)". (In this case a number — say, 70 — will be substituted; other commands will require the substitution of a name or phrase.)
4. Strike the RETURN key.

TEXT immediately responds to the ATTN key by typing an underline. The typing element backspaces one character after the underline is typed, so that the next character typed will appear directly above the underline. The RETURN key merely ends the current line, returning the typing element to the left margin of the next line. Thus this command will appear at the terminal as

w;70

As another example, consider a line typed as

abcdefATTNBACKSPACEBACKSPACEBACKSPACExyzRET

This represents the following actions on the part of the user:

1. Type the letters a, b, c, d, e, and f.
2. Strike the ATTN key.
3. Strike the BACKSPACE key three times.
4. Type the letters x, y, and z.
5. Strike the RETURN key.

This example replaces the characters d, e, and f with x, y, and z. The line will appear at the terminal as

abcde_f_

even though it will be stored and later printed as abcxyz. Notice that the underline appears after the end of the character string since the typing element was positioned there when the ATTN key was struck. If the user manually advances the platen before typing the replacement characters x, y, and z, this line will appear at the terminal as

abcdef_
xyz_

Notice that manually advancing the platen does not cause a new line to be entered into the document in the way that hitting the RETURN key does. Advancing the platen merely prevents overstrikes and thus makes copy easier to read as the user makes a correction.

Most examples in this manual will show both forms of a command: first the command as the user types it, using the symbols ATTN, RET, and BACKSPACE for the ATTN, RETURN, and BACKSPACE keys; and then the same command as it appears at the terminal. Some examples may show only one form, however, especially when an example is extremely long or toward the end of the manual when the reader's familiarity with TEXT will make both forms unnecessary.

Document Storage

There are two areas of the computer where documents are stored: "working storage" and "permanent storage".

Working Storage

Working storage is an area in which TEXT stores a document that is currently being worked on. It can be thought of as a temporary storage area used by the user to work on a document before transferring it to permanent storage.

When the user enters a new document at the terminal or copies an existing document from permanent storage, the document is automatically entered into working storage. When the user is finished with a document, he should transfer it to permanent storage and then clear working storage so that another document can be entered. If he does not do this, any other document will merely be added to the one already in working storage. In other words, a new document entered into working storage does not automatically replace the document already there. In addition, a document is not automatically cleared from working storage at the end of each terminal session; it is carried over from one terminal session to another unless cleared by the user.

Note that transferring a document to permanent storage does not clear it from working storage; a copy of the document still remains in working storage and may still be altered even though the changes will not be made to the one in permanent storage. The commands used to clear working storage and to transfer data to permanent storage are described in Chapter 4 under "Storing and Retrieving a Document".

Permanent Storage

When a terminal user wants to retain a document for future use, he must transfer the document from working storage to an area called permanent storage. To alter or print a document that is in permanent storage, the user must first transfer the document back to working storage; he can never work on a document while it is in permanent storage. The commands for transferring documents from one storage area to another are described in Chapter 4 under "Storing and Retrieving a Document".

How to Use TEXT

To use TEXT, the terminal user first establishes a connection between the terminal and the computer, normally via a telephone line. This procedure is described in Chapter 3. Once the connection with the computer and the TEXT system is established, TEXT accepts input from the terminal and stores the input in the computer. This is called "entering" the document into computer storage, and requires no user action other than typing the document. Special user actions -- called commands -- are necessary only for TEXT functions such as printing or changing parts of the document. Fewer than 20 commands are sufficient for handling most ordinary, small or medium-size documents; these basic commands are given in Chapter 4.

TEXT automatically assigns a line number to each line of a document as it is entered, starting with the first line as line 1, the second as line 2, etc. These line numbers are not printed at the terminal as a document is entered, but are stored along with each line in working storage. Line numbers are used in editing functions like correcting spelling errors, inserting and deleting words or phrases in existing lines, erasing lines, and inserting new lines. All these editing functions are performed by typing commands.

Before the user can do any editing, he must know the line numbers of the lines to be edited. (The exception is editing with the "find" command, discussed in Chapter 5). Normally he finds the line number by requesting a printout with the actual line numbers, although a document may also be edited by relative line numbers, counting back from the end of the document (see "Editing Commands", Chapter 5). After editing the document, the user may instruct TEXT to print the final draft at the terminal.

Once a document has been entered into TEXT, edited, and a copy printed, the terminal user may permanently store the document before disconnecting the terminal from the computer. The stored document may then be retrieved at any future time for updating or correcting. As previously mentioned, documents are saved in the "permanent storage" area of the computer. The commands used to store and retrieve documents are described in Chapter 4.

Entering a Simple Document

Figure 1 illustrates how a simple document is entered into TEXT. It is provided merely as an introduction to the types of things the user must do to use TEXT; the reader is not expected to understand every line at this point. The line numbers in the right margin are for illustrative purposes only and represent those automatically assigned by TEXT; they are not printed at the terminal as the user enters a document.

The ATTN key is struck in the middle of lines 9 and 30 to correct typing mistakes on those lines. On line 9 the user types "hte" instead of "the", notices the error, strikes the ATTN key (thereby producing the underline), backspaces to the first erroneous character ("h"), manually advances the platen one line, and retypes the line from there on. Failure to advance the platen will result in the corrected text being typed directly over the original text, making the copy difficult to read (even though it is correct internally). The typing correction on line 30 is of the same nature as that on line 9. In both cases TEXT does not assign a line number to the incomplete part of the corrected line because the user has not struck the RETURN key yet.

Notice that only three commands are used to enter the entire document into the system; in fact, many documents can be entered using only these three commands.

Printing a Simple Document

Now that the document shown in Figure 1 has been entered into working storage, it is ready to be printed on a command from the user. This command involves striking the ATTN key, typing the letter p, and then striking the RETURN key. TEXT responds with the message SIGNAL WHEN READY and pauses, giving the user a chance to insert a clean sheet of paper before printout begins. This sequence is shown at the terminal as

```
  p
SIGNAL WHEN READY
```

To begin actual printout the user must slowly strike the ATTN key twice. (When printing a document an underline is not typed in response to the ATTN key since the underline will show on the printout.) TEXT then takes control of the terminal and prints the entire document, as shown in Figure 2, according to format instructions previously given by the user. (These format instructions are discussed in Chapter 4.) Notice that the two lines which were entered in the unformatted mode are printed exactly as they were entered; and that the remaining lines, entered in the formatted mode, are rearranged by TEXT to comply with the user's format instructions — which in this case were for a line width of 80 characters per line.

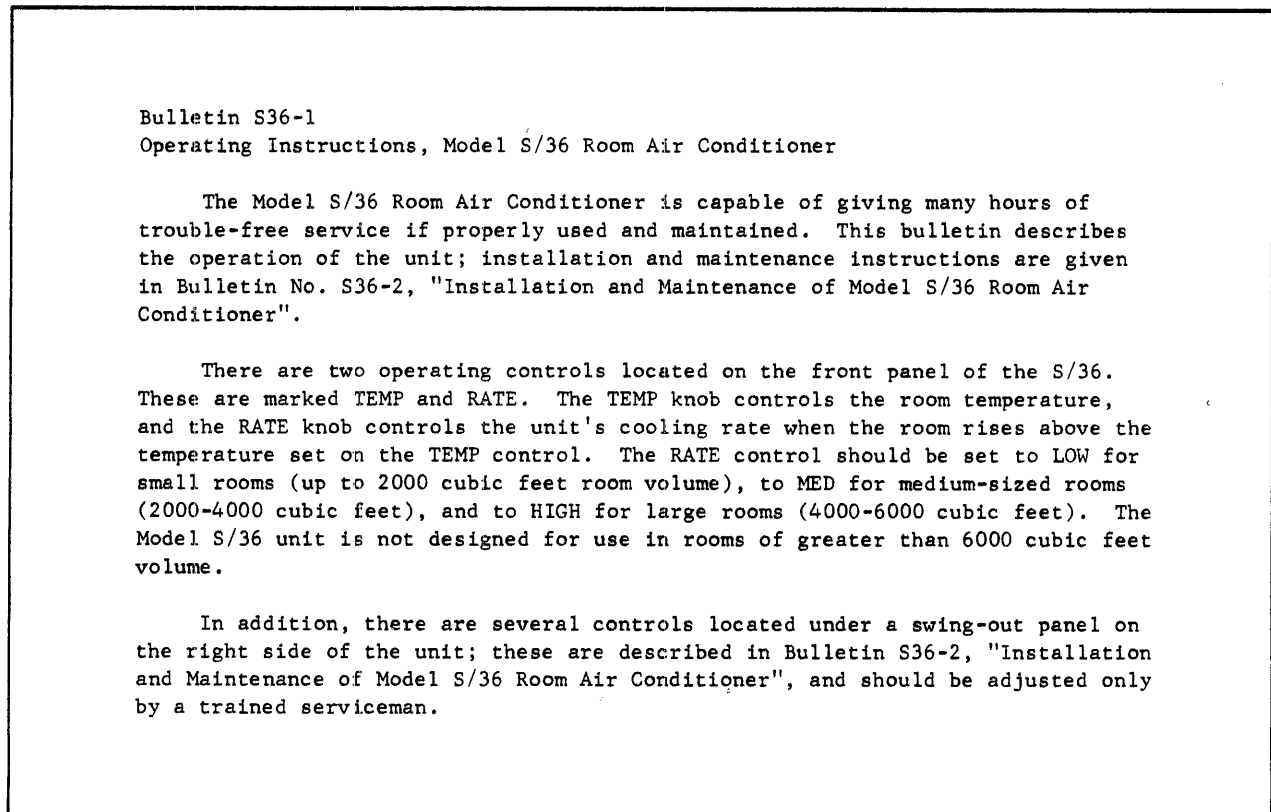


Figure 2. Printout of a Simple Document

Summary

The terminal is the basic link between the user and TEXT. The user enters the document into the TEXT system, and controls TEXT via the terminal. TEXT merely records, edits, and prints the document according to user commands.

The user first establishes a connection between the terminal and TEXT, normally by telephone, and then begins to type the document. TEXT automatically records each line entered, and assigns a line number to it so that the line may be referred to later in editing. For anything other than recording the document being entered, the user must strike the ATTN key and then type a command. Everything the user types at the terminal is either a command to TEXT or part of the document itself. Commands are preceded by the ATTN key; if this key stroke is omitted, TEXT assumes that the line is part of the document and not a command.

Commands to TEXT fall into one of two classes: system commands and embedded commands. System commands call for immediate action on the part of TEXT. Embedded commands do not require immediate action, but are stored with the document for action during subsequent printing.

Finally, information entered at the terminal is entered into working storage, and any editing is performed on information in working storage. Documents needed for future use are saved in permanent storage. Information is transferred between working storage and permanent storage via commands from the user. Any document entered at the terminal or any document copied into working storage from permanent storage is added to whatever is already in working storage, unless the user first clears the storage area. Thus it is important to remember to clear working storage before entering a new document or copying an old one from permanent storage.

3. TERMINAL OPERATIONS

This chapter describes the terminal, its associated equipment, and procedures for establishing a connection between the terminal and TEXT.

Description of Terminal

The terminal is basically a typewriter that can be connected to a Xerox Sigma computer. It has some switches that must be set to put it into operation and a keyboard that is used to enter and print text and commands. The keyboard is similar to an ordinary typewriter keyboard except for the ATTN key in the upper right corner (see Figure 3). This is the previously mentioned ATTN key that is used to get the attention of TEXT when a command is about to be issued. When the terminal is not being used in conjunction with TEXT, it can be used as an ordinary typewriter.

There are many different models of terminals and each has slight variations, especially in the names and locations of switches. This chapter describes a typical terminal.

Terminal Start/Shutdown Procedures

Start Procedure

Preparing Terminal for Use

The user must first prepare the terminal for use before actually establishing connection with TEXT; that is, he must open a communications line between the terminal and the computer. For a typical terminal he does this as follows:

1. Position the keyboard ON-OFF switch at ON. This switch is located in the lower right corner of the keyboard (see Figure 3).
2. Position the LOCAL-REMOTE switch at REMOTE. This opens the communications line between the terminal and the computer. When this switch is positioned at LOCAL, the terminal is not connected to the computer and functions as an ordinary typewriter.

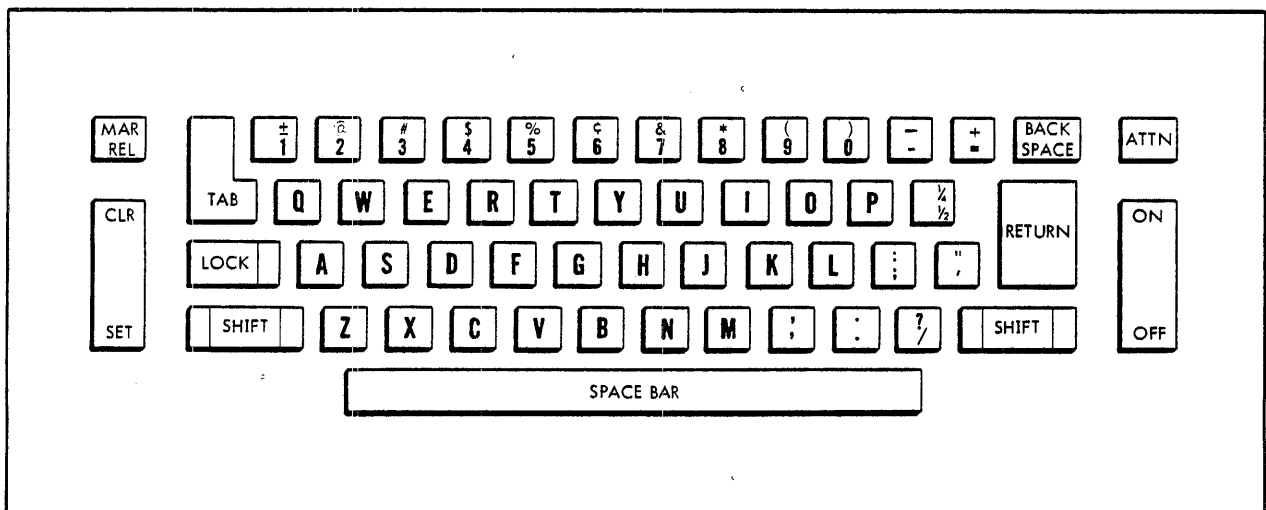


Figure 3. Keyboard of Typical Terminal

3. Establish communications with the computer via a telephone line connected to the terminal as follows:
 - a. If the telephone has an ON-OFF switch, position this at ON.
 - b. Pick up the telephone receiver and listen for a normal dial tone. (Some telephones may have a button labeled TALK; for these telephones, press the TALK button before picking up the receiver.)
 - c. Dial the telephone number for the computer.
 - d. Listen for a high-pitched tone. When this tone is heard, place the receiver in the receptacle provided. (Some telephones may have a button labeled DATA; for these telephones, press the DATA button and hang up the receiver.)

Remember that these operating procedures apply to a typical terminal. Operating procedures for other terminal models may differ slightly, and so may the names and locations of the switches described above. In addition, some terminal models do not have telephone equipment attached, and require only steps 1 and 2 above to establish a direct-line communication between the terminal and the computer. For most terminal models, however, connection with the computer is established via a telephone line as described above.

TEXT Log-On Procedure

Once the communications line with the computer has been established, the next step is to establish a connection with TEXT. The user does this by "logging on" as follows:

1. Type an asterisk (*) and strike the RETURN key. The computer responds by typing the following message at the terminal:

```
CP-V AT YOUR SERVICE
ON AT      time and date
LOGON PLEASE:
```

where "time and date" are the actual time and date, and the last line is a prompt for the user to key in identifying code words (on the same line as the message).

2. Type in identifying code words: a user account number, a user identification word, and an optional password (in that order), separated by commas.[†] The account number and password (if present) each consist of from one to eight alphanumeric characters, and the user identification word consists of from one to twelve alphanumeric characters. (Alphanumeric characters are any combination of the letters a-z or A-Z and the numbers 0-9.) A sample log-on entry is

```
LOGON PLEASE: 7361,ALINCOLN,SECRETWD
```

In this example 7631 is the user account number, ALINCOLN is the identification word, and SECRETWD is the password. (In most installations only uppercase letters are allowed in code words.) The code words must be the same as in the computer's list of authorized users. If they are not the same, the computer will request that the user reenter the identifying code words. If the user's log-on code is accepted, the computer types an exclamation mark (!) at the terminal (with typing elements that have no exclamation mark, this will print as whatever character is in place of the exclamation mark).

3. Type the word TEXT in uppercase letters and then strike the RETURN key to complete the log-on procedure. This input will be shown at the terminal as

```
ITEXT
```

[†]Several users may be logged on at the same time with the same account number, but each user must have a different user identification word.

When the terminal is successfully connected to TEXT, the following messages appear at the terminal:

```
TEXT -- Version nnn
name SIGNED-ON time date

WIDTH 60   DEPTH 54   CONTINUOUS-FORM DEPTH 66
-----t-----t-----t
      t       t                               t

UNFORMATTED MODE
NEXT LINE NUMBER   1.000
```

These messages relate system status conditions associated with the document being entered or retrieved. The first line shows the version of TEXT being used. The second line shows the user name or identification number, a confirmation of signing on, and the date and time of signing on.

The WIDTH and DEPTH values on the third line are the line width (number of characters per line) and page depth (number of lines per page) that a document will have when it is printed; and the CONTINUOUS-FORM DEPTH value is the number of lines per page for continuous-form (roll) paper. These values are established at the previous terminal session or by default and can be changed by the user to suit document needs. Minimum and maximum line widths that may be assigned are 5 and 140 characters, and minimum and maximum page depths are 5 and 98 lines. If the user has never specified line width and page depth, a line width of 60 characters and a page depth of 54 lines will be used by default. The commands for changing line width and page depth are described in the next chapter, and the command for changing lines per page for continuous-form paper is described in Chapter 5.

The fourth line (----t----t----) shows the system tab setting now in effect (left over from the previous terminal session), or the default system tab settings if the user has never specified system tab settings. Each t represents a tab setting. These tab settings will be stored with any document and will be used when the document is printed at the terminal. They may be cleared and changed by the user to suit document needs, as described in the next chapter.

The t's on the fifth line indicate the physical tab settings at the terminal. These tab settings are not stored with the document. If they are not positioned directly under the t's on line 4, the user should change them to agree with the system tab settings, as described in the next chapter.

The UNFORMATTED MODE message on the sixth line reminds the user that any text entered will be in the unformatted mode (that is, will be printed exactly as entered). If there is a document already in working storage (left over from the previous terminal session) when the user logs on, the message FORMATTED MODE may be printed on the sixth line instead.

The message on the seventh line shows the line number that will be assigned to the next line typed. If there is no document left in working storage from the last terminal session, this number will be 1.000; otherwise, it will be one greater than the last line number in the document still in working storage.

Shutdown Procedure

The user signals the end of the terminal session by typing the command

```
Ⓜend Ⓜ
```

This means strike the ATTN key, type the word "end", and strike the RETURN key. Depending on the computer installation, this command either immediately breaks the connection between the terminal and the computer or causes an exclamation mark (!) to be printed at the terminal. If the exclamation mark appears, type the following command on the same line:

```
OFF Ⓜ
```

This means type the word OFF in capital letters and then strike the RETURN key. Since the ATTN and RETURN keys are not displayed, this sequence of commands appears at the terminal as

```
end
TOFF
```

At this point a summary of the terminal and computer time for the session is automatically printed. After this summary message the user should disconnect the terminal from the computer by positioning the LOCAL-REMOTE switch at LOCAL, turn off the terminal and all associated equipment, and hang up the telephone receiver.

Xerox 3010 Terminal

The following Xerox 3010 features can be used if 3010 terminal output is invoked:

- Absolute vertical tabbing
- Absolute horizontal tabbing
- Forward/backward printing

By default, output is not formatted for the 3010 terminal. If a 3010 terminal is to be used, the following command should be input immediately after entering TEXT:

```
ⓂⓂⓂ xterm;3010 ⓂⓂⓂ
```

All subsequent print commands ("p") will then use the special 3010 features.

When in 3010 mode, a left margin can be specified by entering the following command:

```
ⓂⓂⓂ margin[(n)] ⓂⓂⓂ
```

where (n) represents the number of spaces in the left margin. For example, if a margin of 10 spaces is desired, type:

```
ⓂⓂⓂ margin 10 ⓂⓂⓂ
```

To exit 3010 terminal mode, the following command is given:

```
ⓂⓂⓂ xterm;{tty } ⓂⓂⓂ  
2741
```

4. BASIC TEXT COMMANDS

As illustrated in Chapter 2, a user can enter a document using as few as three TEXT commands. With a knowledge of fewer than 20 basic TEXT commands, the user will be able to handle most small or medium-size documents. These basic TEXT commands are described in this chapter under the following categories:

- Document formatting for printout
- Building a document
- Printing a document
- Editing a document
- Storing and retrieving a document
- Ending a terminal session

Document Formatting for Printout

Document formatting commands determine the appearance of a document when it is printed. They are used to specify the following: format mode (formatted or unformatted) to be used in entering lines of the document; margins (number of characters per line); page depth (number of lines per page); paragraph indentation; and tab settings.

Formatted Mode

Lines of text entered in the formatted mode will be automatically reformatted by TEXT and printed according to user commands. Lines to be entered in this mode must be preceded by the embedded command

`(ATTN)tf(RET)`

As discussed previously, this command means

1. Strike the ATTN key.
2. Type the letters t and f.
3. Strike the RETURN key.

Remember from Chapter 2 that the ATTN key produces an underline at the terminal, and the RETURN key ends the command and returns the typing element to the beginning of the next line. Thus this command appears at the terminal as

tf

TEXT then responds to this command by printing the message

FORMATTED MODE

When TEXT encounters this tf command during printout of the document, it automatically reformats succeeding lines of text (according to user instructions) until another format-mode command is encountered.

In the formatted mode the lines may be entered without regard to line length and page depth. The terminal operator then specifies to TEXT the line width (number of characters per line) and page depth (number of lines per page) to use in printing the document, and TEXT reformats and prints the document to those specifications. No attempt is made to hyphenate words that will extend beyond the specified line length — the entire word is simply typed on the next line. As a simple example, suppose the terminal user enters the following text in the formatted mode:

```
This is a demonstration of text
typed in the
formatted mode. This paragraph
will be automatically formatted on printout
by the
TEXT system.
```

Notice that the user has made no attempt to maintain even margins. This is perfectly permissible, since TEXT reformats lines that are entered in the formatted mode. Now suppose that the user instructs TEXT to print this example, using a line width of 40 characters. The printout will look like this:

```
This is a demonstration of text typed in
the formatted mode. This paragraph will
be automatically formatted on printout
by the TEXT system.
```

A line width of 30 characters will produce the following printout:

```
This is a demonstration of
text typed in the formatted
mode. This paragraph will be
automatically formatted on
printout by the TEXT system.
```

Notice in both printouts that each line contains as many words as possible without exceeding the specified line width. An optional printout format exists in which TEXT aligns the right margin to the exact line width specified by inserting extra blanks between words; this is called "right justification" and is discussed in Chapter 5.

Unformatted Mode

Lines of text entered in the unformatted mode are printed exactly as entered; that is, with the same line length and same spacing between words and lines. This mode is automatically in effect when the user logs on to TEXT. Once any other mode has been requested, TEXT may be returned to the unformatted mode with the command

```
(ATTN)tu(RET)
```

which means

1. Strike the ATTN key.
2. Type the letters t and u.
3. Strike the RETURN key.

This command appears at the terminal as

```
tu
```

TEXT responds to this command by printing the message

```
UNFORMATTED MODE
```

When this tu command is encountered during subsequent printout of a document, automatic formatting ceases and all succeeding data is printed exactly as it was entered until another format-mode command is encountered.

The unformatted mode is useful for putting certain words on certain lines, for leaving extra spaces between lines, and for setting up tables. Suppose, for example, that the user wants the titles "Section 6.31" and "WAIVER" printed as the first two lines of his document. He will enter the first two lines in the unformatted mode and the rest of the document in the formatted mode as follows:

```
(ATTN)tu(RET)
Section 6.31(RET)
WAIVER(RET)
(ATTN)tf(RET)
(RET)
No waiver of a breach of any provision(RET)
of this contract shall constitute a(RET)
waiver of any other breach, or of such(RET)
provision.(RET)
```

Remember that the ATTN key produces an underline and the RETURN key ends the line. Striking an extra RETURN key (that is, a line with only a RETURN key) produces a double space. This input will appear at the terminal as

```
tu
UNFORMATTED MODE
Section 6.31
WAIVER
tf
FORMATTED MODE
```

```
No waiver of a breach of any provision
of this contract shall constitute a
waiver of any other breach, or of such
provision.
```

UNFORMATTED MODE and FORMATTED MODE were typed by TEXT in response to the formatting commands `tu` and `tf`. With a line width of 35 characters, the final printout for this example will be

```
Section 6.31
WAIVER

No waiver of a breach of any
provision of this contract shall
constitute a waiver of any other
breach, or of such provision.
```

Notice that TEXT printed the first two lines just as they were entered, and rearranged and printed the remaining lines according to the line width specification.

Another application of the unformatted mode is in the construction of tables. For example, a user can enter a table as follows:

```
tu
UNFORMATTED MODE
Feature                Brand A  Brand B

Two-tone paint          Yes       Yes

Thermostatic control   Yes       Yes

Hypervated spamfratch  Yes       No
```

Here the example is shown only as it appears at the terminal. Again, UNFORMATTED MODE was printed by TEXT in response to the `tu` command.

TEXT also provides a special input mode – the table mode – that makes construction of tables even simpler; this advanced capability is discussed in Chapter 5, "Advanced Commands". Several other variations are also available within the unformatted mode, such as automatically centering lines in the middle of the page, or typing a designated heading or footing on each page of a multipage document. These capabilities are also discussed in Chapter 5, "Advanced Commands".

Line Width, Page Depth, and Tab Settings

Two basic commands are used in TEXT to set up the format for printing a document. One command sets the line width (maximum number of characters per line) and the page depth (maximum number of lines per page), and the other sets the tabs. In general both commands must be used when a new document is being constructed or when an old one is being retrieved from permanent storage.

Line Width and Page Depth Settings

The basic command used to set line width and page depth can be varied to specify line width and page depth separately or in a single command.

Line Width Setting Only. The line width setting allows the user to control the number of characters per line that are printed at the terminal. The command to set the line width has the form

`(ATTN)w(width)(RET)`

where (width) is the number of characters per line. (Remember from Chapter 2 that items in parentheses merely means that something is to be supplied by the user; the parentheses are not typed as part of the command.) For example, the command to set the line width to 70 characters is

`(ATTN)w70(RET)`

The line width setting applies only to data entered in the formatted mode, since data entered in the unformatted mode is printed exactly as it was entered. Line width settings may vary from a minimum of 5 characters per line to a maximum of 140 characters per line; if the user has never specified a line width, a setting of 60 characters per line will be used by default.

Page Depth Setting Only. The page depth setting allows the user to control the number of lines printed per page. The command to set the page depth has the form

`(ATTN)w;(depth)(RET)`

where (depth) is the number of lines per page. Note that the semicolon must be included in the command. For example, the command to set the page depth to 50 lines per page is

`(ATTN)w;50(RET)`

When the document is printed, TEXT will print the specified number of lines and then stop, giving the user a chance to insert a clean sheet of paper before signaling TEXT to continue (described later in this chapter under "Printing a Document"). The page depth setting applies to data entered in both the formatted and unformatted modes. Page depth may vary from a minimum of 5 lines per page to a maximum of 98 lines per page; if the user has never specified a page depth, a setting of 54 lines per page will be used by default.

Combined Line Width and Page Depth Settings. Line width and page depth may be set simultaneously with a single command of the form

`(ATTN)w(width);(depth)(RET)`

where (width) and (depth) are as described above. For example, the command to set a line width of 70 characters per line and a page depth of 50 lines per page is

`(ATTN)w70;50(RET)`

Tab Settings

Tab settings are used at the terminal in the same way that they are used at an ordinary typewriter, except that the user must indicate the tab settings to TEXT in a special way.

The tab settings used to enter a document are stored with the document and become "system tab settings"; however, the actual tab settings at the terminal, known as "physical tab settings", are not stored. System tab settings are merely an image of the actual physical tab settings at the terminal.

The terminal status message during TEXT log-on (see Chapter 3) indicates to the user the existing tab settings and points out the differences between the system tab settings stored with the document and the current physical tab settings at the terminal. The first line of t's in the status message shows the system tab settings, and the second line of t's shows the physical tab settings. If these two lines are different, the user should reset the tabs to agree, as described below. In resetting the tabs the user can reset the physical tab settings at the terminal to agree with the stored system tab settings, or he can reset both the system tab settings and physical tab settings.

Normally if the user is going to enter a new document, he will set both the physical and system tabs to the values required for the new document. If he is going to retrieve an existing document, he should set the physical and system tabs to match those of the retrieved document, as described under "Retrieving a Document" later in this chapter.

Clearing Tabs. Before resetting the physical tabs at the terminal, the user must clear the existing ones or both the new and the old tabs will be in effect. To clear the old physical tabs:

1. Position the typing element at the right margin by striking the TAB key several times.
2. Hold down the CLEAR key, strike the ATTN key and then the RETURN key, returning the typing element to the left margin. The CLEAR key may be released once the typing element is at the left margin.

Now the new tabs can be set as described below. Notice that it is not necessary to clear the system tabs; this is done automatically when new system tabs are set.

Setting New Tabs. To set physical tabs to agree with the system tabs stored with a document, space to the desired positions, and depress the SET key. To reset system tabs, type a command of the form

ⓐ (hyphens) t (hyphens) t (hyphens) t ... t ⓑ

This means strike the ATTN key, type a series of hyphens up to the desired tab setting, type the letter t at the desired tab setting, repeat hyphens and t's through the last tab setting, and strike the RETURN key. For example, suppose the user wants to reset the system tabs to every tenth character of a 55-character line; he can do this with the following command:

ⓐ-----t-----t-----t-----t-----t-----tⓑ

Now notice that there are more than 55 print positions in this command. The first five t's set the tab settings for the 55-character line, while the last t sets the tab setting for printing line numbers in case the user later requests a line-numbered printout. The last tab setting is described later in this chapter under "Printing Copies With Line Numbers".

To reset both physical tabs and system tabs at the same time, follow the instructions for setting system tabs and merely depress the SET key before typing each t.

Control of Paragraphs

When entering lines of text in the unformatted mode, the user simply enters all paragraphs exactly as they are to appear on printout. When entering lines of text in the formatted mode, however, the user must indicate paragraphs in a special way as described below.

In the formatted mode the user normally indicates a new paragraph by entering one or more blank lines (that is, by striking a RETURN key for each blank line). He then indicates paragraph indentation by indenting the first two paragraph lines exactly as they are to appear on the printout. The first line indicates the indentation for the first line of the paragraph, and the second line indicates the indentation for the rest of the paragraph. Indentation of the first two paragraph lines is handled with tab settings; it cannot be done with just the space bar. The remaining paragraph lines should be entered at the left margin of the input page, and TEXT will automatically indent them for printout. (The remaining paragraph lines should not be indented, since this may signal to TEXT that another new paragraph is being started, as described below.) The indentation spaces are included in the total line width; for example, if a line width of 50 characters is specified and the first line of a paragraph is indented five spaces, the effective line width for this line is 45 spaces. Table 1 shows examples of paragraphs as they are entered in the formatted mode, and then as they are printed with a line width of 50 characters.

As mentioned above, new paragraphs in the formatted mode are normally indicated by entering one or more blank lines. Another method of indicating new paragraphs must be used, however, for completely single-spaced documents that have no extra space between paragraphs. For these documents, the user merely indents the first two lines of the new paragraph as described above, and the different indentation of the first line, as compared to the second line of the preceding paragraph, signals TEXT that a new paragraph is starting. This method will not work, however, with block indented paragraphs since the indentation of the first line of the new paragraph is the same as that of the second line of the preceding paragraph. When this situation occurs, a t+0 embedded command should be used to signal the start of a new paragraph.

Widow Suppression

When printing a document (described later in this chapter), TEXT automatically provides for "widow suppression". A widow line is either the last line of a paragraph falling at the top of a page or the first line of a paragraph falling at the bottom of a page. This is considered by most writers as sloppy formatting, and therefore it is avoided whenever possible. In order to accomplish this, TEXT may put an extra line (one more than the specified page depth) at the bottom of a page or compose a page of one too few lines.

Table 1. Sample Paragraphs in Formatted Mode

Paragraph Type	Text as Entered	Text as Printed
Block	<p>During the past 20 years, the prohibitive costs of trial-and-error design methods have prompted the wide-spread use of simulation and mathematical modeling in research and design projects.</p> <p>By constructing a mathematical model before constructing the actual physical object or system, the designer predetermines the performance of the finished product with a high degree of accuracy.</p>	<p>During the past 20 years, the prohibitive costs of trial-and-error design methods have prompted the wide-spread use of simulation and mathematical modeling in research and design projects. By constructing a mathematical model before constructing the actual physical object or system, the designer predetermines the performance of the finished product with a high degree of accuracy.</p>
Block Indented	<p>During the past 20 years, the prohibitive costs of trial-and-error design methods have prompted the wide-spread use of simulation and mathematical modeling in research and design projects. By constructing a mathematical model before constructing the actual physical object or system, the designer predetermines the performance of the finished product with a high degree of accuracy.</p>	<p>During the past 20 years, the prohibitive costs of trial-and-error design methods have prompted the wide-spread use of simulation and mathematical modeling in research and design projects. By constructing a mathematical model before constructing the actual physical object or system, the designer predetermines the performance of the finished product with a high degree of accuracy.</p>
Indented	<p>During the past 20 years the prohibitive costs of trial-and-error design methods have prompted the wide-spread use of simulation and mathematical modeling in research and design projects. By constructing a mathematical model before constructing the actual physical object or system, the designer predetermines the performance of the finished product with a high degree of accuracy.</p>	<p>During the past 20 years, the prohibitive costs of trial-and-error design methods have prompted the wide-spread use of simulation and mathematical modeling in research and design projects. By constructing a mathematical model before constructing the actual physical object or system, the designer predetermines the performance of the finished product with a high degree of accuracy.</p>
Overhanging	<p>During the past 20 years, the prohibitive costs of trial-and-error design methods have prompted the wide-spread use of simulation and mathematical modeling in research and design projects. By constructing a mathematical model before constructing the actual physical object or system, the designer predetermines the performance of the finished product with a high degree of accuracy.</p>	<p>During the past 20 years, the prohibitive costs of trial-and-error design methods have prompted the wide-spread use of simulation and mathematical modeling in research and design projects. By constructing a mathematical model before constructing the actual physical object or system, the designer predetermines the performance of the finished product with a high degree of accuracy.</p>

Building a New Document

When the user first establishes a connection with TEXT, working storage contains whatever document was present when the user last logged off. At this point the user may continue to work on this same document, or may clear working storage and either retrieve an existing document from permanent storage or begin to enter the text of a new document. Retrieval of an existing document is described later in this chapter under "Storing and Retrieving a Document", while building a new document is described below.

Clearing Working Storage

To clear the document that is currently in working storage, type the command

`(ATTN)clear(RET)`

If the user wants to save the document that is in working storage, he must remember to store it before clearing working storage as described later in this chapter under "Storing and Retrieving a Document".

Setting Up the Format for Printout

Before entering a new document, the user should set up the format for printing the document. This includes setting the tabs and specifying line width and page depth as described earlier under "Line Width, Page Depth,

and Tab Settings". In brief review, physical tabs are set with the SET key, and system tabs are set with a TEXT command of the form

ⓐ(hyphens) t (hyphens) t (hyphens) t ... t ⓑ

Line width and page depth are set with a command of the form

ⓐw(width);(depth)ⓑ

where (width) and (depth) are to be replaced by the number of characters per line and the number of lines per page, respectively. It should be remembered that the width and depth of the document are significant only during print-out; the document is not being formatted as it is entered.

Entering a Document

Once the printing format is established, the user may begin entering the actual document. He must designate whether lines are being entered in the formatted mode or the unformatted mode. When the user first logs on, any lines typed will automatically be entered in the unformatted mode, and will therefore be printed exactly as entered. To enter lines in the formatted mode, recall that the user must first type the command

ⓐtfⓑ

to which TEXT responds with the message FORMATTED MODE. Any text entered after this point will automatically be formatted on printout according to the user-specified line width and page depth. The user may return to the unformatted mode at any time by typing the command

ⓐtuⓑ

to which TEXT responds with the message UNFORMATTED MODE.

TEXT automatically assigns a line number to each line as it is entered; the first is line 1, the second is line 2, and so forth. These line numbers are used in performing editing functions. The line numbers are not printed at the terminal as the lines are entered, but are retained in working storage; once a document has been entered, the user may request a line-numbered printout (as described later in this chapter under "Printing a Document").

Corrections While Entering a Line

If the user makes an error while entering a line, he may correct it as long as he has not struck the RETURN key to end the line. He may correct it by deleting the entire line and retyping it, or by backspacing to the point of error and retyping from that point on. Error correction after the RETURN key has been struck is described later in this chapter under "Editing a Document".

Line-Deletion Procedure

If most of a line is in error or if the error is at the beginning of a line, the user may wish to delete the entire line and reenter it. He does this by typing the following command and then retyping the line:

ⓐⓓⓔⓁⓔⓣⓔⓑ

Backspace Procedure

Correcting Text. If an error is not too far back in a line, it may be corrected simply by backspacing to the error and retyping from there on. The backspacing is done with a command of the form

ⓐⓑⓓⓔⓔⓔⓓⓔⓑ

where ⓓⓔⓔ represents the BACKSPACE key. The exact user procedure is as follows:

1. Strike the ATTN key.
2. Strike the BACKSPACE key as many times as needed to backspace to the first erroneous character. Each time the user strikes the BACKSPACE key, TEXT backspaces the typing element one position and deletes that character from working storage (even though it still shows at the terminal).
3. Manually advance the platen one line so that the line will be readable when the correct text is typed; otherwise the new text will be typed over the old and will be unreadable.

4. Retype from this point to the end of the line.
5. Strike the RETURN key.

Examples of errors corrected in this way are shown in Table 2.

Table 2. Correcting Lines Via the Backspace Procedure

Example as Typed	Example as Printed at the Terminal
This booklet por (ATTN) (BKSP) (BKSP) rovides	This booklet por_ rovides
(ATTN) w40 (ATTN) (BKSP) (BKSP) 35 (RET)	w40 -35-

Correcting Commands. Most TEXT commands may be corrected with the backspace procedure while they are being entered. For example, suppose that while entering the line width command the operator mistakenly types a line width of 40 characters per line instead of 35. He can correct this as shown in the second example in Table 2.

Number of Corrections. Any line may be corrected via the backspace procedure any number of times before the RETURN key is struck; the same characters may even be backspaced over and recorrected several times. The only restriction is that the user may not backspace over a tab; that is, over the striking of the TAB key to shift the typing element to the next tab. If he tries this, TEXT ignores the backspacing, prints the error message ****ILLEGAL BACKSPACE OVER TAB CHARACTER**, and permits the user to retry the operation.

Correcting Strikeovers and Underlines. If the user backspaces over a character that has a strikeover or an underline, both characters in the print position are deleted. For example, suppose the user enters the following portion of a line:

This movement is called a drigt

To change "drigt" to "drift", he strikes the ATTN key and backspaces over the t and the g; this deletes the t and the g and the underlines. Then he types the correct characters f and t, backspaces two positions, and retypes the underlines under those characters. This same rule applies if the user backspaces over the first character of a command, which is always underlined (in response to the ATTN key): both the first character of the command and the ATTN are deleted, and thus the user must restrike the ATTN key to reenter the command.

Printing a Document

Once a document is entered into working storage or is copied there from permanent storage, the user may request a printout by typing the basic print command

~~(ATTN)~~ p ~~(RET)~~

TEXT responds with the message

SIGNAL WHEN READY

which gives the user a chance to insert a clean sheet of paper before printing begins. To initiate the actual printout the user should then type the command

~~(ATTN)~~ ~~(ATTN)~~

In response to this sequence of commands, TEXT prints a copy of the document that is in working storage, rearranging lines of text that were entered in the formatted mode according to the user's line width specification. While TEXT is printing the document, it has control of the terminal keyboard and locks all keys except ATTN.

If the document is more than one page, TEXT pauses after typing the number of lines specified by the current page depth. The user then inserts another piece of paper and strikes the ATTN key twice to restart the printout. This process continues until the entire document is printed.

The user may stop printout at any time and get control of the keyboard by striking the ATTN key twice. He can then either enter another command or enter more text to be added to the end of the document; he cannot resume printout of the rest of the page, but he can restart printout from the beginning of the page (see pr command, Chapter 5).

There are many variations of the basic print command, including options to print a copy with right-justified margins, to print double-spaced or triple-spaced copies, to print on continuous-form paper, and to print only a specified part of the document. These variations are discussed in Chapter 5, "Advanced Commands". Only two additional variations of the basic print command are discussed here, and they both are used to obtain line-numbered copies of the document for editing.

Printing Copies with Line Numbers

As previously mentioned, TEXT assigns a line number to each line of a document and this line number is used for editing purposes. However, TEXT does not print the line number at the terminal as the line is entered; to obtain the line numbers, the user may enter the document and then request a printout with line numbers. This is done with one of two variations of the print command: one creates an as-entered copy, and one creates a formatted copy that looks like the final draft. These variations are discussed below.

Before requesting a printout with line numbers, the user must set both a physical and a system tab at the position where the line number is to be printed. This position should be the last tab setting on the line, and should be at least eight print positions in from the right edge of the page, since line numbers may consist of up to eight characters. When a line-numbered printout is requested, TEXT types the line number at the last tab setting. If a line happens to extend beyond the last tab position, TEXT will print the line number approximately 5 spaces to the right of the end of the line. (See "Tab Settings" earlier in this chapter.)

As-Entered Printout

The first variation of the line-numbered printout is the as-entered printout. With this form TEXT prints each line exactly as it was entered (including lines entered in the formatted mode), and prints the line number to the right of each line at the last tab setting. The user obtains this type of printout by typing the command

`(ATTN)pan(RET)`

where pan stands for "print, as-entered, with line numbers". TEXT responds with the message

SIGNAL WHEN READY

which gives the user a chance to insert a clean sheet of paper in the terminal before printing begins. To initiate the actual printout, the user should then type the command

`(ATTN)(ATTN)`

In response to this sequence of commands, TEXT prints a copy of the document that is in working storage, exactly as it was entered, and prints the line numbers at the last tab setting. Again, the user can stop printout at any time and get control of the keyboard by striking the ATTN key, and then striking the RETURN key after printing stops.

Figure 4 shows a copy of a document obtained with the as-entered-printout command. Note that this copy includes both text and embedded commands (line 13). Embedded commands are subject to editing just like any other part of the document.

SAMPLE DOCUMENT	1.000
	2.000
This is a sample document. This first	3.000
section is being entered in the	4.000
unformatted mode, which is the	5.000
default mode when the terminal	6.000
operator first connects to the system	7.000
and logs on.	8.000
Text in the unformatted mode is	9.000
always printed out exactly as it	10.000
was entered.	11.000
We now change to the formatted mode.	12.000
<u>t</u> f	13.000
	14.000
This text is being typed in the	15.000
formatted mode. The system	16.000
automatically takes care	17.000
of the formatting of any text typed in the	18.000
formatted mode, adjusting it to the line width	19.000
and page depth	20.000
specified by the terminal operator.	21.000

Figure 4. As-Entered Printout with Line Numbers

Formatted Printout

Instead of the as-entered printout, the user may want a line-numbered printout that is formatted according to the specified line width and page depth in order to see roughly how the final draft will look. To get this printout the user types the command

(ATTN)pn(RET)

where pn stands for "print, with line numbers". Again TEXT responds with the message

SIGNAL WHEN READY

giving the user a chance to insert a new sheet of paper. Actual printout can then be initiated by typing in the command

(ATTN)(ATTN)

Figure 5 shows the same text shown in Figure 4, but printed as a result of the formatted-printout command; the specified line width is 50 characters per line. Notice the differences: embedded commands (line 13) are not included, and lines entered in the formatted mode (lines 15 through 21) are reformatted and printed according to the specified line width of 50 characters. The underscores are added by TEXT to let the user know where each new input line starts; and the number to the right of the line is the line number corresponding to the first underscore. For example, look at the line marked as 17. The first underscore indicates the beginning of input line 17, and the second underscore indicates the beginning of the next input line, line 18. Thus the user may determine the line number of any printed line, even of reformatted printed lines that do not correspond to the lines as entered.

As-Entered Printout Versus Formatted Printout

The choice between the two line-numbered printout options just discussed depends on the personal preference of the user. The as-entered printout has the advantages of showing embedded commands and of showing each line as it exists in the computer. Its primary disadvantage, of course, is that data entered in the formatted mode does not look the same as it will on the final copy; that is, it is not formatted. The formatted printout, on the other hand, looks the same as the final draft, except for the line numbers in the right margin. Its disadvantages are that it does not show embedded commands and that it does not show the lines as they actually exist in the computer.

<u>S</u> AMPLE DOCUMENT	1.000
	2.000
<u>T</u> his is a sample document. This first	3.000
<u>s</u> ection is being entered in the	4.000
<u>u</u> nformatted mode, which is the	5.000
<u>d</u> efault mode when the terminal	6.000
<u>o</u> perator first connects to the system	7.000
<u>a</u> nd logs on.	8.000
<u>T</u> ext in the unformatted mode is	9.000
<u>a</u> lways printed out exactly as it	10.000
<u>w</u> as entered.	11.000
<u>W</u> e now change to the formatted mode.	12.000
	14.000
<u>T</u> his text is being typed in the <u>f</u> ormatted mode.	15.000
The system <u>a</u> utomatically takes care <u>o</u> f the	17.000
formatting of any text typed in the <u>f</u> ormatted	19.000
mode, adjusting it to the line width <u>a</u> nd page	20.000
depth <u>s</u> pecified by the terminal operator.	21.000

Figure 5. Formatted Printout with Line Numbers

Editing a Document

Once a document has been entered and a line-numbered copy obtained, it may be edited. Editing includes such functions as correcting lines or parts of lines, erasing lines, and inserting additional lines between existing lines. Both embedded commands and regular text are subject to editing. Almost all editing functions are performed by specifying the line number of the line to be edited.

Line Corrections

Basic Line-Editing Command

The user makes corrections by first "calling out" the line number of the line in error. He does this with the basic line-editing command, which has the form

ATTN(line no.)RET

where (line no.) is to be replaced by the number of the line to be edited. For example, to edit line 46 the user types the command

ATTN46RET

TEXT then prints line 46 at the terminal and stops at the end of the line. At this point the user can edit the line as if he had just entered it and had not yet struck the RETURN key. That is, he has four possibilities:

1. Leave the line unchanged, by striking the RETURN key to end it.
2. Correct the line, by striking the ATTN key, backspacing to an error, and retyping the rest of the line.
3. Delete the entire line, by striking the ATTN and RETURN keys.
4. Add more to the end of the line.

Phrase-Substitution Command

In addition to the basic line-editing command, TEXT also has a phrase-substitution editing capability. This enables the user to substitute any phrase (any sequence of characters) in place of any other phrase on a given line. The

phrase replaced may be part of a word, or even several words, and the phrase replacing it need not be the same length. The phrase-substitution command has the form

```
(ATTN)(line no.);(old phrase);(new phrase)(ET)
```

where

(line no.) is the number of the line to be edited.

(old phrase) is the phrase to be replaced.

(new phrase) is the replacement phrase.

For example, suppose that line 46 reads "voltage regulator must be repolarized" and that the user wants to change "polarized" to "polarized". He can do this with the command

```
(ATTN)46;polarized;polarized(ET)
```

If TEXT finds the phrase to be replaced, it makes the replacement and then prints the altered line at the terminal, stopping at the end of the line:

```
voltage regulator must be repolarized
```

the user can now exercise one of the four editing options described above.

If TEXT cannot find the phrase to be replaced, it prints the message

```
**NO SUCH PHRASE
```

As in the above example, the "phrase" to be replaced may be only part of a word; on the other hand, it may be a full word or several words. The only requirement is that the user include enough characters to uniquely identify the phrase being replaced. For example, instead of replacing "polarized" with "polarized", it is sufficient merely to substitute "polar" for "polor", or even "lar" for "lor". However, it is not sufficient to instruct TEXT to substitute "ar" for "or", since the word "regulator" on the same line also contains the phrase "or". TEXT substitutes the new phrase for the first occurrence of the phrase to be replaced; thus in the above example the request to substitute "ar" for "or" will result in the word "regulator" being changed to "regulatar", rather than the word "repolarized" being changed to "repolarized".

Replacing Short Words. Care must also be exercised in replacing short words such as "and", "if", "to", "a", etc., since they may be contained within a longer word elsewhere on the same line. For example, if line number 26 of a document is

```
In some lines, short words may be hidden in longer words.
```

and the user merely asks to have the phrase "in" replaced by "inside", the result at the terminal will be

```
26;in;inside
```

```
In some linsidee, short words may be hidden in longer words.
```

The first occurrence of the phrase "in" in the original line 26 was in the word "line", and therefore this is the phrase that TEXT replaced with "inside" (note that the first word of the line, "In", begins with a capital letter and hence cannot be confused with the phrase "in"). To prevent this the user must include enough additional characters to uniquely identify the phrase. In this case it is sufficient to include the space in front of the word "in":

```
26; in; inside
```

```
In some lines, short words may be hidden inside longer words.
```

Do not forget to include a blank in front of the replacement word "inside"; see what happens without the blank:

```
26; in;inside
```

```
In some lines, short words may be hiddeninside longer words.
```

In general it is advisable to include the space before or after a short word in case the word may be hidden inside another word on the same line.

Words That Appear Twice on a Line. The user must also uniquely identify a word that appears twice on one line, or TEXT automatically replaces the first occurrence of the word. This can be accomplished by including one or two characters from the word on either side of the word to be replaced. For example, suppose that line 93 of the document currently in working storage is

If the data light is not on, the information transmitted is lost.

and that the user wants to change the second "the" to "any". He can do this as follows:

93; the in; any in

If the data light is not on, any information transmitted is lost.

Supplying the first two characters of the next word is sufficient to distinguish the second "the" from the first. The comma before the second "the" could have been supplied instead.

Length of Phrases. A replacement phrase need not be the same length as the phrase being replaced; it may be shorter or longer and may even contain a different number of words than the phrase being replaced. For example, assume that line 7 of the document currently in working storage is

A rolling stone gathers moss.

and that the user wants to insert the word "no" before "moss". This can be accomplished with

7;ers moss;ers no moss

A rolling stone gathers no moss.

Deleting Words or Phrases. The same substitution command can also be used to delete a word or phrase. To delete a phrase, simply call out the line number and the phrase (followed by a semicolon) as if to replace it, but do not supply any replacement phrase. For example, suppose that line 77 of the document currently in working storage is

The cost of the auxiliary power unit is not included in the quoted price.

and that the user wants to delete the word "not". This can be accomplished with

77; not;

The cost of the auxiliary power unit is included in the quoted price.

The semicolon after "not" is mandatory. Notice the space before "not". Either the space before or after a word should be deleted along with the word; otherwise two spaces are left between remaining adjacent words, instead of one space as desired.

Correction of Most of Line or End of Line

Another variation of the basic line-editing command can be used to delete the remainder of a line beyond a given word or phrase. For example, suppose that line 68 of the document currently in working storage is

This is not true in some Asiatic countries, for example, Korea.

and that the user wants to delete everything after the word "countries". This can be accomplished as follows:

68;ries

This is not true in some Asiatic countries

Now the line may be edited as if the user had just typed it. That is, the user may do any of the following:

1. Strike the RETURN key to end the line.
2. Strike the ATTN key and backspace further into the line, deleting more characters, and then retype to the end of the line.
3. Strike the ATTN and RETURN keys to delete the line, and then reenter it.
4. Add to the end of the newly shortened line.

For instance, the user can delete line 68 beyond the word "countries", as before, and then add to the shortened line as follows:

```
68;ries
This is not true in some Asiatic countries such as China.
└──────────────────────────────────────────────────────────┘
      Printed by TEXT                                     Typed by user
```

The same result can be obtained by the phrase-substitution method, substituting the phrase "such as China," for the phrase ", for example, Korea. ", but it requires typing both phrases instead of just the new one. When most of a line has to be retyped, it is often quicker to delete most of the line and retype it as shown above instead of substituting one long phrase for another.

Semicolon Within Data to be Edited

One more feature of the basic line-editing command must be mentioned — the treatment of a semicolon within data to be edited. Remember that TEXT ordinarily treats a semicolon in a command as the end of a phrase. If a phrase to be edited by the basic line-editing command contains a semicolon, the semicolon must be the first character of the phrase; thus there will be two semicolons in a row. For example, suppose that line 3 of the document currently in working storage reads

```
disconnected from the supply tank; therefore the fluid
```

and the user wants to change it so that "therefore" begins a new sentence. This can be accomplished as follows:

```
3;; there;. There
disconnected from the supply tank. Therefore the fluid
```

The two consecutive semicolons signal TEXT that the second semicolon is part of the phrase to be replaced. The same rule applies if a semicolon appears in a phrase to be deleted or in a phrase marking the point beyond which the remainder of the line is to be deleted. But if a replacement phrase contains a semicolon, the semicolon need not be the first character. For example, assume that line 38 of the document currently in working storage reads

```
The "phrase" to be replaced may be only part of a word. On the other hand,
```

and that the user wants to combine the two sentences, using a semicolon. He can accomplish this as follows:

```
38;word. On;word;on
The "phrase" to be replaced may be only part of a word; on the other hand,
```

Line Erasures

TEXT provides a command that allows direct erasure of lines containing text or embedded commands, without first typing the line.

Erasing a Single Line. The command to erase a single line of text has the form

```
(ATTN)e(line no.)Ⓢ
```

where (line no.) is the line number of the line to be erased. For example, to erase line 273 of the current document the user types the command

```
(ATTN)e273Ⓢ
```

TEXT then prints the word ERASE and the first few words of line 273 in order to verify that the correct line is being erased, and waits for a response from the user. If the wrong line has been specified, the user cancels the command simply by striking the RETURN key. To proceed with the erasure, the user must type the letter y (for yes) and strike the RETURN key. TEXT then erases the line from working storage and prints the following message to verify that the erasure has been made: NEXT LINE NUMBER xx, where xx is the line number that will be assigned to the next line

the user enters at the end of the document. The complete command sequence to delete line 273 of the document appears at the terminal as

```
e273
ERASE-- for which the actual ?y
NEXT LINE NUMBER 531
```

Erasing Multiple Lines. The erase command may also be used to erase more than one line at a time. In this case the command may have either of the following forms

```
(ATTN)e(starting line no.)(ending line no.)(RET)
(ATTN)e(starting line no.)-(ending line no.)(RET)
```

where

(starting line no.) is the number of the first line to be erased.

(ending line no.) is the number of the last line to be erased.

For example, to erase lines 57 through 64 of the current document the user may type either of the following commands:

```
(ATTN)e57;64(RET)
(ATTN)e57-64(RET)
```

TEXT then prints a one-line message that contains the word ERASE, the first few words of the first line to be erased, the word THRU, and the first few words of the last line to be erased. If this indicates that a wrong line has been specified, the user can cancel the command by striking the RETURN key. To proceed with the erasure, the user must type the letter y and then strike the RETURN key, just as for a single-line erasure. TEXT then erases the lines from working storage and prints the message NEXT LINE NUMBER xx, where xx is again the line number that will be assigned to the next line the user enters at the end of the document. The complete procedure appears at the terminal as:

```
e57;64
ERASE-- of the first part --THRU-- be binding on ?y
NEXT LINE NUMBER 84
```

When the erase command is used, the line numbers of erased lines become unassigned (nonexistent), and any attempt to call out those lines for editing results in the error message **ILLEGAL LINE NUMBER. Erasing a line does not affect the line numbers of remaining lines; that is, the document is not renumbered.

Inserting/Moving Lines

Inserting Lines

One or more lines of text may be inserted anywhere in a document.

Inserting a Single Line. A single line of text may be inserted with the command

```
(ATTN)i(line no.)(RET)
```

where (line no.) is the line number to be given to the line. The user then types in a line, and that line is inserted in the document with the specified line number. TEXT verifies completion of the insertion by printing the message

```
NEXT LINE NUMBER xx
```

where xx is the line number that will be assigned to the next line the user enters at the end of the document. For example, the complete sequence to insert a line at line 63 will appear at the terminal as

```
i63
Insert this line as line number 63
NEXT LINE NUMBER 273
```

If a line already exists with the specified line number, the following question is printed:

REPLACE--first part of line?

A "y^(RT)" response causes the original line to be destroyed and replaced with the new one.

The insert command may also be used to insert a line between two existing lines, simply by specifying a line number that is between the two adjacent lines. The inserted line number need not be an integer; for example, to insert a line between lines 63 and 64, the user can type the command

^(ATTN)i63.5^(RET)

Inserted line numbers may have up to three digits after the decimal point (for example, 63.001 to 63.999).

Inserting Multiple Lines. More than one line may be entered by specifying a line number increment, using the following format:

^(ATTN)i(line number);(increment)^(RT)

If (line number) exists, TEXT prints out a question as described above. Then if a "y^(RT)" response is given, TEXT prints out the insert line number, and the user types in the first line which is inserted in the document and assigned the specified line number. TEXT now prints out a line number for the next line (equal to the previous line number plus the increment) and the user enters another line; this process continues for each successive line.

The insertion is terminated when an existing line is encountered, when the user types in another system command, or when the command

^(ATTN)iend^(RT)

is entered.

For example, suppose the user wishes to insert three lines between lines 63 and 64; the procedure appears as follows:

```
(ATTN)i(63.200);(.200)(RT)
63.200:
Insert the first line as 63.200
63.400:
Insert the second line as 63.400
63.600:
Insert the third line as 63.600
63.800:
iend
NEXT LINE NUMBER 273.00
```

Moving Lines

New lines of text or embedded commands may be built at the end of a document and moved between existing lines of the document, and existing lines may be moved from one point to another.

Moving a Single Line from the End of a Document. A single line may be entered at the end of a document and then moved to another point in the document with the command

^(ATTN)m(line no.)^(RT)

where (line no.) is the number of the line after which the line is to be moved. For example, assume that a new line is to be inserted after line 32 of the document currently in working storage. The user first enters the new line and strikes the RETURN key, which adds the new line to the end of the current document, and then types the command

^(ATTN)m32^(RT)

This command means "move the last line of this document after line 32". TEXT responds with a one-line message that includes the word AFTER, the first few words of the line specified in the move command, the word MOVE, and the first few words of the line to be moved. This lets the user check if the correct lines have been specified. If a wrong line has been specified, the user can cancel the move command simply by striking the RETURN key. To proceed with the move, the user must type the letter y (for yes) and strike the RETURN key. TEXT verifies the move by printing the message NEXT LINE NUMBER xx, where xx is the line number that will be assigned to the next line the user enters. Thus if line 32 of the current document reads "A new line is to be moved after this line.", the complete move procedure appears at the terminal as:

```
This line is to be moved after line 32
m32
AFTER-- A new --MOVE-- This line ?y
NEXT LINE NUMBER 84
```

This move command causes the last line of the document to be moved after line 32, and erases it from the end of the document. In addition, the moved line is assigned a new line number between 32 and the number of the next line. Thus if the next line after line 32 is line 33, the inserted line will be assigned the number 32.5. In this way lines can be inserted without destroying the numbering of the rest of the document. Another line moved after line 32 will be assigned a number between 32 and 32.5, or 32.2; a line moved after line 32.5 will be assigned a number between 32.5 and 33, or 32.7; and so on until all available numbers between 32 and 33 are used. Since line numbers in TEXT can have up to three digits after the decimal point (e.g., 32.001 to 32.999), up to 999 lines can be inserted between any two lines of the original document. If the user tries to insert more than this, TEXT responds with the error message

****TOO MANY LINES TO MOVE**

Moving Multiple Lines from the End of the Document. Several lines may be added to the end of a document and then moved to another point in the document. The command for this type of move has the form

```
(ATTN)m(line no.);(starting line no.)(RET)
```

where

(line no.) is the number of the line preceding the point of insertion.

(starting line no.) is the line number of the first of the lines to be moved.

For example, suppose that the user wants to add three new lines after line 27 (which reads "Three new lines are to be inserted after this line.") and that the last line of the document is line 76. He enters the three lines (which are assigned line numbers 77, 78, and 79) and types the command

```
(ATTN)m27;77(RET)
```

TEXT responds with a one-line message that includes the word AFTER, the first few words of the line that the insert is to follow, the word MOVE, the first few words of the first line to be moved, and the phrase THRU END OF DOCUMENT. Again, if this printout indicates that a wrong line has been specified, the user can cancel the command by striking the RETURN key. To complete the move, the user must type the letter y and then strike the RETURN key, just as for a single-line move. TEXT then completes the move and prints the message NEXT LINE NUMBER xx, where xx is again the line number that will be assigned to the next line the user enters. The complete move process is shown at the terminal as:

```
These three lines are to be
moved after
line number 27.
m27;77
AFTER-- Three new --MOVE-- These three --THRU END OF DOCUMENT ?y
NEXT LINE NUMBER 77
```

The command m27;77 inserts the three new lines (from line 77 to the end of the document) after line 27, and erases them from the end of the document. Since these lines are erased, NEXT LINE NUMBER again becomes line 77;

remember that this is the number to be assigned to the next line the user enters and is always at the end of the document. The three inserted lines are assigned new line numbers between the adjacent lines.

Determining Next Line Number. Note that to insert multiple lines the user must know the first line number of the lines to be moved. This can easily be determined by typing the following command before the lines are entered:

`(ATTN)n(RET)`

This command instructs TEXT to print the number to be assigned to the next line the user enters, which will of course be the first of the lines to be moved. Using the previous example, the printout at the terminal will be

`n
NEXT LINE NUMBER 77`

where NEXT LINE NUMBER 77 is printed by TEXT. Of course if the user forgets to request the next line number before entering the lines to be moved, he can always request the next line number afterward and then count back to the first line to be moved. For example, if the user enters 10 lines to be moved and he finds out via the next-line-number request that the next number is 90, the first line number to be moved is line 80.

Moving Groups of Existing Lines. A group of lines may be moved from one part of a document to another, with the following form of the move command:

`(ATTN)m(line no.);(starting line no.);(ending line no.)(RET)`

where

(line no.) is the number of the line preceding the point of insertion.

(starting line no.) is the first number of the lines to be moved.

(ending line no.) is the last number of the lines to be moved.

For example, assume that lines 130-136 of the document currently in working storage are

Move three lines after this line.	130
Not this one	131
nor this one	132
but this one	133
through	134
this one,	135
Stop.	136

To move lines 133-135 after line 130, type the command

`(ATTN)m 130;133;135(RET)`

This command says "after line 130, move lines 133 through 135". TEXT responds with a one-line message containing the word AFTER, the first few words of line 130, the word MOVE, the first few words of line 133, the word THRU, and the first few words of line 135. This again gives the user the opportunity to cancel the command if a wrong line has been specified. To cancel the command, he simply strikes the RETURN key. To complete the move, he types the letter y and then strikes the RETURN key. TEXT then completes the move and prints the message NEXT LINE NUMBER xx, where xx again is the line number that will be assigned to the next line the user enters. The complete procedure for moving these lines will appear at the terminal as:

```
m130;133;135
AFTER-- Move three --MOVE-- but this --THRU-- this one, ?y
NEXT LINE NUMBER 237
```

The three lines are moved and renumbered, and the old lines 133-135 are erased from the document. Note that when lines are moved from the middle of the document the erased line numbers become nonexistent, and any attempt to reference these line numbers results in an error message; but new lines can be moved or inserted at these line numbers.

Moving a Single Existing Line. To move a single line from one point to another point within a document, use the group-move command just described but specify the same line number for the beginning and the end of the lines to be moved. Thus to move only line 133 to after line 130 in the previous example, type the command

```
m130;133;133
```

This move operation will appear at the terminal as:

```
m130;133;133
AFTER-- Move three --MOVE-- but this ?y
NEXT LINE NUMBER 237
```

Precautions in Moving Groups of Lines

When moving groups of lines (from the end of a document or from within a document), certain precautions are in order.

Maintaining Paragraphs. If an entire paragraph is moved from one formatted section to another, remember to include at least one blank line before the paragraph if it is supposed to remain a separate paragraph. Otherwise the moved paragraph will be added to the preceding paragraph.

Maintaining Format Commands. If a line or section of text is moved from an unformatted section into a formatted section (or vice versa), remember to precede it with the appropriate formatting command and to follow it with the formatting command to return the document to its original formatting mode. Of course these formatting commands may be built and moved around the transplanted text in a separate move operation.

Storing and Retrieving a Document

The working storage and permanent storage areas of the computer were discussed in Chapter 2. In brief review, working storage is a temporary storage area containing the document currently being worked on. All documents are first entered into working storage and are eventually printed from this area. There may be only one document in working storage at a time (several documents added together in working storage are considered as one document). Permanent storage is used to save documents for future use. Documents cannot be edited or printed while in permanent storage; before editing or printing a document, the user must copy it into working storage. Documents are copied back and forth between working storage and permanent storage, cleared from working storage, and deleted from permanent storage with the commands described below.

Storing a Document

The user may permanently save a document that is in working storage by copying it into permanent storage; this is referred to as "storing" the document. When he stores the document, he stores a unique name along with it for identification purposes, and sometimes a code word (or "password") to restrict its availability to other users. All of this is handled with the "store" commands described below.

When a document is stored, TEXT automatically stores the current format specifications for printout (line width, page depth, and system tab settings) along with it, and then prints them when the document is retrieved.

Until a storage request is successfully completed, TEXT has control of and locks the terminal keyboard. On completion of storage, TEXT prints the following message at the terminal:

```
DOCUMENT STORED
```

Note that storing a document in permanent storage merely places a copy there; the document is not literally "picked up" and transferred to permanent storage. Thus the original still remains in working storage and may still be altered even though the changes will not appear in the copy in permanent storage.

If the store command is interrupted before completion (by hitting the ATTN key), the following message is printed:

```
STORE COMMAND INTERRUPTED
NUMBER OF LINES STORED: xxx
CONTINUE STORE ?
```

If the user given a "y" response, the store command continues; any other response stops the command with xxx lines stored.

Basic Store Command

The basic store command is used to copy a document from working storage into permanent storage. This command has the form

```
 $\text{ATTN}$ s;(name) $\text{RET}$ 
```

where (name) is the name to be assigned to the document. The document name may consist of from 1 to 31 alphanumeric characters (that is, alphabetic A-Z, a-z, and numeric 0-9); it may not contain any blanks. As an example, suppose the user wants to store the document that is currently in working storage into permanent storage, and give it the name EXAMPLE. He will type the command

```
 $\text{ATTN}$ s;EXAMPLE $\text{RET}$ 
```

TEXT acknowledges completion of storage by printing the message DOCUMENT STORED. Thus this example will appear at the terminal as

```
s;EXAMPLE
DOCUMENT STORED
```

Note that, depending on the length of the document being stored, there may be a substantial pause between the time the store command is given and the time the DOCUMENT STORED message is returned.

Store Command with Document Identification and Password

Variations of the store command allow the user to associate a "document identification" and a "document password" with the document.

Document Identification. The document identification is a brief description of the document and is printed along with the name of the document when the user requests a list of all documents stored in an account or requests to have a document deleted.

Document Password. Every document in permanent storage is associated with the account number under which the user "logged on"; this number is the first of the three log-on code words used to gain access to TEXT. Normally each TEXT user will have a different account number. These account numbers provide a limited form of document security; that is, a user logged on under one account cannot access documents in another account without knowing the account number and cannot ever delete documents in another account.

If several users share an account number, additional document security can be provided in the form of a password stored with a document. Unlike an account number, a password does not have to be authorized or approved by the computer center. The document password prevents unauthorized users from getting a document (that is, copying it into working storage) or deleting a document from permanent storage. If no password is supplied when the document is stored, anyone in the same account may get or delete a document; and anyone in another account may get the document if he knows the account number, but may not delete it.

Note: The document password used when storing a document has no connection with the log-on password that was discussed in the section titled "TEXT Log-On Procedure" in Chapter 3.

Command Format and Examples. To store a document with a document identification and a document password, use the following form of the store command:

```
s;(name);[(document identification)];[(document password)]␣
```

where

(name) is the name to be given to the document. It may consist of from 1 to 31 alphanumeric characters, with no blanks allowed.

(document identification) is the optional identification for the document. It may consist of from 1 to 63 alphanumeric characters, with blanks allowed.

(document password) is the optional password. It may consist of from 1 to 8 alphanumeric characters, with no blanks allowed.

Notice that everything following the document name in this store command is optional; that is, any of the entries may be omitted. If an entry is omitted at the end of the command, the semicolon preceding it should also be omitted. If an entry is omitted in the middle of a command, however, the semicolon must be included. Examples of this store command are shown in Figure 6 as they will appear at the terminal.

Additional Variations of Store Command

Three additional variations of the store command permit storing a portion of a document, renumbering a stored document, and storing a document in a format suitable for use by programs other than TEXT. These variations of the store command are described in Chapter 5, "Advanced Commands".

Duplicate Document Names

No two documents in the same account in permanent storage may have the same name. If the user attempts to store a document with a name that has already been used, TEXT prints the following message:

```
DUPLICATE NAME--DELETE OLD DOCUMENT?
```

If a "y[␣]" response is given, the old document will be deleted, and the store operation will continue. If any other response is given, the command will be cancelled.

<p><u>With all options</u> <u>s</u>;Fred;Employee payroll file;Groucho</p> <p><u>With document identification only</u> <u>s</u>;AbCd;NASA proposal</p> <p><u>With document password only</u> <u>s</u>;FILE1;;GET12345</p> <p><u>With no options</u> <u>s</u>;EXAMPLE</p>
--

Figure 6. Store Command with Various Options

If the old document was stored with a document identification and the user specifies a new one in the store command, the new identification replaces the old one. If the user does not specify a new identification, the following message is displayed:

DO YOU WISH TO RETAIN OLD DOC ID?

If the user types in a "y" response, the old document identification will be retained; any other response causes the old identification to be deleted.

Storing a Nonexistent Document

If the user types a store command and there is nothing in working storage to store, TEXT types the error message

**NOTHING TO STORE

Retrieving a Document

The user may "retrieve" a document from permanent storage by copying it into working storage. When retrieving the document, the user must specify the document name and its password, if one was stored with the document. All of this is handled with variations of the "get" command as described below. Note that the get command actually copies the named document into working storage; it does not literally "pick up" the document and transfer it to working storage. Thus a copy still remains in permanent storage. Once the document has been copied into working storage, it may be worked on as if the user had just entered it at the terminal.

TEXT acknowledges retrieval of a document by printing the number of lines in the document, the format specifications in effect when the document was stored, and the next line number to be assigned. This is described in more detail below.

Get Command

The basic get command is used to retrieve a document previously stored in the same account with no password. It has the form

(ATTN)g;(name)(RT)

where (name) is the name stored with the document. For instance, suppose the user wants to retrieve a document named EXAMPLE; the command is

(ATTN)g;EXAMPLE(RT)

This command says "get the document named EXAMPLE which is in permanent storage in my account". If a document password was stored with the document, then that password must be supplied to retrieve it. For example, if a document named "Secret" was stored with document password "woid", the get command to retrieve this document is

(ATTN)g;Secret;woid(RT)

If the document to be retrieved is in another account, that account number must be supplied along with the document name in the get command. For example, if the document named EXAMPLE was stored (with no password) in account number C371B641, and the user at the terminal is logged on under a different account, the get command to retrieve this document is

(ATTN)g;EXAMPLE.C371B641(RT)

Note that the document name and the account number are separated by a period, not a semicolon. If the document to be retrieved is in another account and was stored with a document password, both the account number and the password must be supplied in the get command. For example, if the document named "Secret" was stored with the password "woid" in account number 7364, and the requesting user is logged on in another account, the get command to retrieve this document is

(ATTN)g;Secret.7364;woid(RT)

Responses to Get Command

Successful Retrieval. When a document has been successfully retrieved from permanent storage, TEXT prints messages that reflect the number of lines in the document, the format specifications (line width, page depth, and system tab settings) that were in effect when the document was stored, and the line number that will be assigned to the next line the user enters. (Depending on the length of the document, there may be a substantial pause between

the time the get command is given and the time these messages are returned.) As an example, note the following TEXT response to the retrieval of a document named CAT:

```
DOCUMENT:CAT
LINES: 260
WIDTH 65  DEPTH 35  CONTINUOUS-FORM DEPTH 72
-----+-----+
UNFORMATTED MODE
NEXT LINE NUMBER 261.000
```

At this point TEXT compares the saved document width, depth, and tabs to those in working storage. If they are the same, nothing further is printed and the get command processing is terminated. However, if they differ, the following message is printed:

```
RESTORE SAVED DOCUMENT WIDTH, DEPTH, TABS?
```

If the user responds with a "y" sequence, working storage will be reset to the width, depth, and tabs of the saved document and the following message will be printed:

```
SAVED DOCUMENT FORMAT RESTORED
```

If any other response to the above question is received, TEXT leaves the working storage format unchanged and prints:

```
WORKING STORAGE FORMAT RETAINED
```

If working storage is clear (empty) when the get request is issued, the requested document is simply copied into working storage; the copy is an exact image of the document when it was stored, including line numbers. If a document is already in working storage, however, TEXT adds the retrieved document to the end of the existing document and automatically renumbers it. In this case TEXT also prints a message containing the last line number of the existing document. The user can determine the new line numbers of the retrieved document by adding the old line numbers to this number. For example, assume line 200 is the last line of a document currently in working storage. Then if a document named PART2, having 100 lines, is retrieved and added to working storage, TEXT prints the following messages:

```
DOCUMENT: PART2
LINES: 100
ADDED AFTER LINE: 200.000
WIDTH 65  DEPTH 35  CONTINUOUS-FORM DEPTH 72
-t---t---t
FORMATTED MODE
NEXT LINE NUMBER 301.000
```

Notice in this case that the lines will be renumbered by adding 200 to the old line numbers. Thus, line 1 of PART2 becomes 201 (that is, 200+1) of working storage, line 20 becomes 220, line 37.64 becomes line 237.64, etc.

This capability of adding one document to another document in working storage allows a large document to be created in parts and then combined. The only restriction is that the same format specifications (line width, page depth, and tabs) should be used in creating each part, since the final combined document may have only one set of print-out format specifications.

Unsuccessful Retrieval. If the user tries to retrieve a document that does not exist or that exists in another account and the account number is incorrectly given, TEXT responds to the get command with the message

```
**DOCUMENT DOES NOT EXIST
```

If the user incorrectly supplies or does not supply a needed password in the get command, TEXT responds with the message

```
**DOCUMENT ACCESS DENIED
```

Deleting a Document

A user may delete a document from permanent storage in the same account at any time by using the delete command; he can never delete a document in another account. To delete a document that was stored without a password, use a basic delete command of the form

```
(ATTN)d;(name)(RET)
```

where (name) is the name that was stored with the document. For example, assume that the document named "Hobart" was stored without a document password; the command to delete this document is

```
(ATTN)d;Hobart(RET)
```

TEXT responds by printing the document identification, or merely the document name if the document was stored without document identification. At this time the user can either cancel the command by striking the RETURN key or proceed with the deletion by typing the letter y and then striking the RETURN key. To acknowledge deletion of the document, TEXT prints a message of the form

```
DOCUMENT DELETED
```

The entire procedure to delete the document named "Hobart" appears at the terminal as

```
d;Hobart
DELETE--Division 3 Employee File    ?y
DOCUMENT DELETED
```

where "Division 3 Employee File" is the document identification that was stored with the document.

If the document was stored with a document password, the user must supply that password along with the document name. For example, if the document "Hobart" was stored with the document password 17348, the TEXT command to delete this document is

```
(ATTN)d;Hobart;17348(RET)
```

If the user does not supply the correct document password in the delete command, TEXT prints the following error message and does not delete the document:

```
**DOCUMENT ACCESS DENIED
```

Clearing Working Storage

Only one document is in working storage at a time. If a document currently exists in working storage and another document is entered at the terminal or retrieved from permanent storage, the new document will be added to the end of the existing document rather than replacing it. If the user does not want to add the new document to the existing one, he must first type the following command to clear working storage

```
(ATTN)clear(RET)
```

If the user wants to save the document that is in working storage, he must remember to store it before clearing working storage. Otherwise the document will be destroyed.

Ending a Terminal Session

To end a terminal session, the user types the command

```
(ATTN)end(RET)
```

This either immediately breaks the connection between the terminal and the computer or causes an exclamation mark (!) to be printed at the terminal. If the exclamation mark appears, type the following command directly after it (on the same line) to complete the logging-off process:

`OFF !`

The logging-off process is also described in Chapter 3.

5. ADVANCED COMMANDS

The commands used in advanced operations are presented in two groups: embedded commands and system commands. The user should not attempt to use these advanced commands until he feels comfortable with the basic commands that were described in Chapter 4.

Embedded Commands

Chapter 2 defined an embedded command as one that is stored with a document for action during subsequent printout, and one that usually specifies formatting or indexing information. (Formatting information controls the appearance of a document when it is printed, and indexing information facilitates quick access to certain portions of the document.) Two embedded commands – the unformatted-mode command and the formatted-mode command – were described in Chapter 4. The remaining embedded commands are described below.

Format-Mode Commands

Chapter 4 introduced the formatted and unformatted modes for entering documents or sections of documents. This chapter describes four other text-entry modes for handling special types of text:

Centered Mode
Heading Mode
Footing Mode
Table Mode

The first three of these modes are variations of the basic unformatted mode, while the fourth is an entirely different text-entry mode designed to handle tabular data.

Centered Mode

Lines entered in this mode are automatically centered on printout. To enter lines in this mode, the user precedes the lines with the command

`(ATTN)tuc(ET)`

This command merely adds a character ("c" for "centered") to the basic unformatted mode command; this is true of all variations of the unformatted mode. TEXT responds to this command by printing the message

UNFORMATTED MODE

Each line the user enters after this point (up to another formatting command) will be centered in the middle of the page when the document is printed. The middle of the page is determined as half the line width specified for the document; if the document line width is changed, centering is adjusted accordingly on printout.

For example, suppose the user wants to center a three-line heading of a document and format the rest of the document to a specified line width. He will enter the heading in the centered mode and the rest of the document in the formatted mode as shown in Figure 7. The final printed document is shown in Figure 8 with a line width of 80 characters.

Notice that text entered in the centered mode is indeed unformatted: each input line is printed as one output line.

```
uc
UNFORMATTED MODE
Bulletin S36-1
```

```
Operating Instructions for
Model S/36 Room Air Conditioner
```

```
f
FORMATTED MODE
```

The Model S/36 Room Air Conditioner is capable of giving many hours of trouble-free service if properly used and maintained. This bulletin describes the operation of the unit; installation and maintenance instructions are given in Bulletin No. S36-2, "Installation and Maintenance of Model S/36 Room Air Conditioner".

Figure 7. Entering Lines in Centered Mode

Bulletin S36-1

Operating Instructions for
Model S/36 Room Air Conditioner

The Model S/36 Room Air Conditioner is capable of giving many hours of trouble-free service if properly used and maintained. This bulletin describes the operation of the unit; installation and maintenance instructions are given in Bulletin No. S36-2, "Installation and Maintenance of Model S/36 Room Air Conditioner".

Figure 8. Printout of Lines Entered in Centered Mode

Heading and Footing Modes

These modes facilitate the printing of one or more lines as headings and footings on specified pages of a document, and also allow automatic page numbering. To enter lines in the heading mode, the user precedes the lines with a command of the form

Ⓢtuh(nn)Ⓢ

where (nn) is a code that specifies the pages on which the heading is to be printed; it must be one of the codes given in Table 3. If the page code is omitted, TEXT automatically prints the headings on all except the first page (i.e., code 23). Text responds to the heading-mode command with the message

HEADING MODE

All lines entered after this up to the next formatting command will automatically be typed at the top left margin of succeeding pages when the document is printed.

Table 3. Page Codes for Headings/Footings

Code [†]	Meaning
12	Heading/footing is to be typed on every page.
13	Heading/footing is to be typed on every odd-numbered page.
23	Heading/footing is to be typed on every page except the first.
24	Heading/footing is to be typed on every even-numbered page.
35	Heading/footing is to be typed on every odd-numbered page except the first.

[†]Note that this code represents the first two pages on which the heading or footing will be printed. Thus for a code of 12, the first two pages will be 1 and 2; for a code of 13, the first two pages will be 1 and 3; etc.

To enter lines in the footing mode, the user precedes the lines with a command of the form

`(ATTN)tuf(nn)(U)`

where nn is one of the codes in Table 3 and specifies the pages on which the footing is to be printed. If the page code is omitted, TEXT automatically prints the footing at the bottom of every page (i. e., code 12). TEXT responds to the footing-mode command with the message

FOOTING MODE

All lines entered after this up to the next formatting command will automatically be typed as a footing at the bottom left margin of succeeding pages when the document is printed.

Examples of these two format-mode commands are shown below. The first example introduces a heading that is to be printed on every page of a document, and the second introduces a footing that is to be printed on every odd-numbered page.

`(ATTN)tuh12(U)`
`(ATTN)tuf13(OT)`

The text of a heading or footing is unformatted; that is, each input line prints as one output line. There is no attempt to combine lines to fulfill line-width requirements as there is with formatted text.

An example of an actual heading entry as it might appear at the terminal is shown below.

```
tuh23
HEADING MODE
Demois Project
FINAL REPORT
```

```
tf
```

In this example the user requests the heading mode, and types a heading to be printed at the top of every page of the document except the first page. The heading includes two lines of text and three blank lines to separate the heading from the rest of the printed page. The command to enter the formatted mode (tf) terminates the heading mode.

An actual footing entry might appear at the terminal as

```
tuf
FOOTING MODE
t+2
CHAPTER 4
PROBABILITY THEORY
tu
_
```

In this example the user requests the footing mode, requests that the footing be separated from the rest of the printed page with two blank lines (this command is detailed later in this chapter), types a two-line footing to be printed on every page of the document, and requests the unformatted mode

Blank Lines Between Heading/Footing and Main Text. The blank lines that separate a heading or footing from main text must be included as part of the heading or footing entry, or there will be no separation from the main text. For example, if two lines are to appear between a heading and the first line of text, the user enters the heading lines and then either strikes the RETURN key twice or types the command t+2 as shown in the above examples. He then ends the heading mode with another format-mode command and proceeds with entering text. The same principle applies to footings.

Effect of Heading/Footing Command During Printout. A heading or footing can be introduced at any point in a document, and will take effect as follows during printout:

1. If a heading-mode command is encountered anywhere on a page, the command will be executed immediately and the heading will appear on that page if requested. Thus for instance, if a heading is to be printed on the first page of a document, the heading-mode command may occur anywhere within that page.
2. If a footing-mode command is encountered on a page, the command will be executed immediately and the footing will appear on that page if requested and if enough room is left on the page.

Changing or Discontinuing a Heading/Footing. A heading or footing can be changed at any time merely by entering a new one; each new heading or footing discontinues the existing one. To discontinue a heading or footing without adding a new one, the user should request a heading or footing mode and then immediately request another format mode (table mode, centered mode, formatted mode, etc.). For example, the following commands can be used to discontinue an existing heading mode:

```
(ATTN)tuh(REF)
(ATTN)tf(REF)
```

Notice that the first command requests the heading mode, and the second command requests the formatted mode.

Page Depth With Heading/Footing. During printout a heading or footing is always printed within the specified page depth, and the number of lines in the main text are adjusted to accommodate the heading or footing. For example, if the user specifies a page depth of 40 lines per page and a heading contains eight lines (including blank lines between heading and text), only 32 lines of the document will be printed on each page. Thus each page will always contain the page depth specified by the user.

Placement of Heading/Footing on Page. As previously mentioned, headings and footings are normally printed beginning at the left page margin. In addition, the user may specify that they be centered, printed against the right margin, or printed alternately against the left and right margins. He does this by using variations of the heading-mode and footing-mode commands as shown in Table 4.

Automatic Page Numbering. The user can employ the automatic page-numbering feature by typing the following command at the point the page number is to appear in a heading or footing:

```
(ATTN)(ATTN)
```

Table 4. Placement of Heading/Footing on Page

Desired Heading/Footing Placement	Heading/Footing Command to Use [†]
<u>Left Margin</u> Heading printed against left margin of each page. Footing printed against left margin of each page.	tuh(nn) tuf(nn)
<u>Right Margin</u> Heading printed against right margin of each page. Footing printed against right margin of each page.	tuhr(nn) tufr(nn)
<u>Centered</u> Heading centered within specified page width. Footing centered within specified page width.	tuhc(nn) tufc(nn)
<u>Alternated</u> Heading printed against left margin on even-numbered pages and right margin on odd-numbered pages. Footing printed against left margin on even-numbered pages and right margin on odd-numbered pages.	tuha(nn) tufa(nn)
[†] Remember that (nn) in the command must be replaced by one of the codes in Table 3.	

When TEXT encounters this command during printout of a heading or footing, it automatically prints the page number and then continues if there is more text to be printed. An example of entering this command is shown below:

```
tuh23
HEADING MODE
Feasibility Study
Hobart Emission Control Device
Page _
tf
_
```

This example will cause the current page number to be printed on the last line of a three-line heading. Notice that the command to print the page number is entered after the word "Page", as evidenced by the underline. TEXT automatically substitutes the current page number in place of this command as each page is printed. Also notice the blank between the word "Page" and the underline. The user must supply blanks between the command and adjacent words, or there will be no space between the page number and adjacent words on printout.

The user may also want to print a page number by itself, with no other heading or footing information. To print just the page number at the bottom of the page, for example, the user may enter the following commands:

```
tuf
FOOTING MODE
t+3
_
tf
_
```

In this example the user requests the footing mode, requests that three lines be skipped between the main text and the footing, spaces to the location for the page number and enters the command to print the page number, and then requests another format mode. This series of commands will cause the page number to be printed five spaces from the left page margin at the bottom of the page.

Table Mode

The table mode greatly simplifies the entry of long or complicated tables. Tables can of course be constructed using the unformatted mode as shown in Chapter 4 (and repeated below as Figure 9), but that method requires manually spacing the entries to proper alignment and is thus time-consuming and error-prone. The table mode is a great improvement over entering tables in the unformatted mode. Before describing the table mode, however, it is necessary to introduce the nomenclature used in connection with tables.

Table Nomenclature. A table is a collection of data (numbers, words, phrases, etc.) arranged in a symmetrical manner. The table constructed in Figure 9 is an example. Each individual item in a table is called an entry; thus in Figure 9 the word "Feature" is an entry, as are the phrases "Two-tone paint", "Brand A", etc. The individual entries of a table are aligned horizontally and vertically for ease in reading and understanding the table. The entries lined-up horizontally are called rows, and those lined-up vertically are called columns. For example, Figure 10 shows a table that has three rows and three columns and visually defines the direction of rows and columns. Individual entries in this table are A, B, C, etc.

Entries are normally typed into a table by rows. Thus to construct the table in Figure 10 the user will normally type first A, then B, and then C to enter the first row; then D, E, and F for the second row; and so forth. In addition, the table mode allows entries to be input by columns. Entries input by columns (A, D, G, then B, E, H, etc.) are then correctly arranged by TEXT.

Table Entry by Rows. To enter tables by rows, type the following table-mode command:

```
A table;r;(n)␣
```

where (n) is the number of lines to be skipped between each output row; (n) may be any integer value from 0 to 98. If (n) is omitted, one blank line will be inserted between each row.

TEXT responds by printing the message

```
TABLE MODE--PROVIDE FORMAT
```

At this point the user must type in a format for the table of the form

```
f.....f...f...␣
```

<u>tu</u> UNFORMATTED MODE			
<u>Feature</u>	<u>Brand A</u>	<u>Brand B</u>	
Two-tone paint	Yes	Yes	
Thermostatic control	Yes	Yes	
Hypervated spamfratch	Yes	No	

Figure 9. Entry of a Table in Unformatted Mode

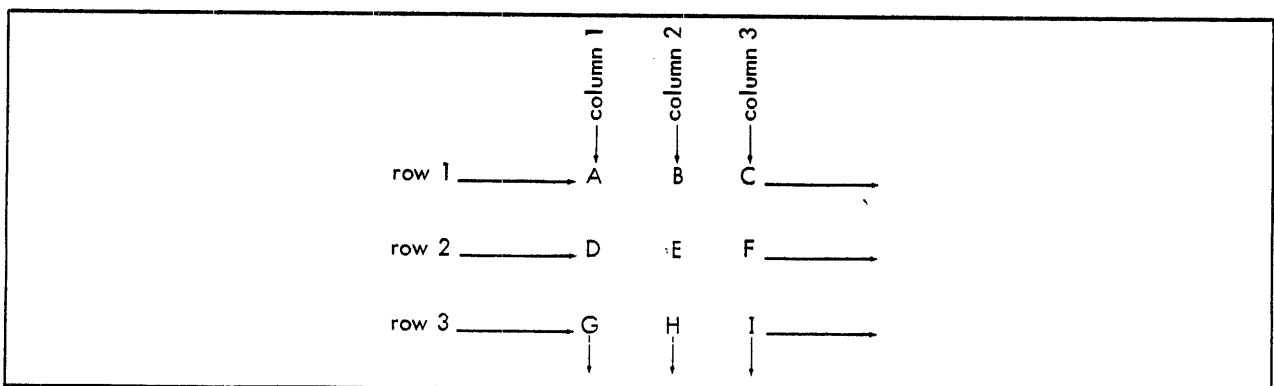


Figure 10. Table with Three Rows and Three Columns

Here each f signifies the beginning of a new column, and the periods specify the width of each column (any f may be replaced by a u, and any number of dots immediately preceding each f or u may be replaced by spaces, as will be described shortly). The table format entry is terminated by a RETURN keystroke. This format specification is assigned a line number just like any other line of the document. Following this format specification, the user types the entries for the table; he follows each individual entry with two RETURN keys, and signals the end of each row by typing the command

tend^(R)

For example, if the user specifies a table format with three columns (three f's, as above), then three entries must be typed to complete each row. The table input is terminated when the user enters another format-mode command (tf, tu, etc.).

An example of entering a table by rows is shown in Figure 11; this is the same table that was previously entered in the unformatted mode (see Figure 9). Notice in Figure 11 that three columns have been specified: the first column starts at the left margin and is 25 spaces wide (the f plus 24 dots); the second column starts at position 26 and is 9 spaces wide; and the third column starts at position 35 and is also 9 spaces wide. Three rows of data are input, with each row having three entries. When the user requests a printout of the document, TEXT will print this table as shown in Figure 12.

```

table;r
TABLE MODE--PROVIDE FORMAT
f.....f.....f.....
Feature

Brand A

Brand B
tend
Two-tone paint

Yes

Yes
tend
Thermostatic control

Yes

Yes
tend
Hypervated spamfratch

Yes

No
tend
tf

```

Figure 11. Entry of a Table in Table Mode, by Rows

<u>Feature</u>	<u>Brand A</u>	<u>Brand B</u>
Two-tone paint	Yes	Yes
Thermostatic control	Yes	Yes
Hypervated spamfratch	Yes	No

Figure 12. TEXT Printout of Figure 11

Notice that TEXT has formed the data into a three-column table, with a column beginning at each f in the table format specification, as illustrated below:

```
f.....f.....f..... (format specification)
  Feature      Brand A Brand B (TEXT printout)
```

Also notice that the entries in Figure 11 are each terminated by two RETURNS. This is mandatory; two entries separated by only a single RETURN will be treated as a single entry. The extra RETURNS are not necessary, however, before or after the tend command.

The example shows how the table mode can simplify the entry of long and complicated tables. Furthermore, the table mode enables the user to experiment with different formats. If the format of the table is unsatisfactory when printed (columns too close together or too far apart, etc.), only the format specification (f..f..) need be changed; it is not necessary to retype any of the table entries.

TEXT treats each column entry as a formatted line of the specified column width (indicated by the f plus the number of dots in the format specification). Thus if a particular column entry is too wide for the column, TEXT automatically continues it on the next line of the same column. For example, suppose that the format specification for the table of Figure 11 is given as

```
f.....f.....f.....
```

This specifies a width of only 17 spaces for the first column, while two entries in the column are longer than 17 characters. TEXT will handle the printout as shown in Figure 13. Notice in this figure that TEXT continues the oversize column entries on the next line and aligns the remaining rows. Any entry may overflow its column width (that is, overflows are not indicated to the first column as shown in this example), and TEXT will automatically continue the column on the next line (and as many following lines as necessary) and realign the remaining rows.

In TEXT tables an undesirable situation may arise if an entry is the same length as the column specification: no space will be left between that entry and the adjacent one on printout. For example, if the first column width for the example of Figure 11 is shortened to 14 characters (f plus 13 dots), the second line of the table will be printed as:

```
Two-tone paintYes    Yes
```

because the column entry "Two-tone paint" is exactly 14 characters long. The user can avoid this situation by inserting one or more spaces immediately before each f in the format specification. This instructs TEXT to leave that many spaces between the columns. These spaces are not included in the specified line width, which is still determined by the f and the number of dots following it. An example of inserting spaces in the format specification is shown below:

```
f..... f.....f.....
```

This format specification instructs TEXT to print up to 12 characters in the first column and to leave at least two spaces between the first and second columns. The second column begins at print position 15. In general it is a good practice to include one or more spaces before each f in the format specification, so that printed table entries will not run together.

<u>Feature</u>	<u>Brand A</u>	<u>Brand B</u>
Two-tone paint	Yes	Yes
Thermostatic control	Yes	Yes
Hypervated spamfratch	Yes	No

Figure 13. TEXT Printout of Figure 11 with Different Format Specification

The user may type the letter u instead of the letter f in any position of a table-mode format specification. The letter u means that a column is to start at that position, but that the column is to be unformatted; that is, entries in this column are to be printed exactly as they were entered. This means that entries for an unformatted column should be no wider than the specified column width.

Unformatted columns may be used to control the placement of table entries; they are especially useful when indenting or centering entries in a column or when creating multiline entries. Figure 14 shows an example of entering a table with unformatted columns, and Figure 15 shows a TEXT printout of the table. Notice in this example that the unformatted columns are used to force the column headings to appear on two separate lines,

```

table;r
TABLE MODE--PROVIDE FORMAT
u..... u..... u.....
  Time
  (sec.)

  Temp.
  (F)

  Press.
  (psia)
tend
  10.0

  74.3

  14.7
tend
  20.0

  93.2

  17.9
tend
  30.0

  23.2
tend
  40.0

  37.4
tend
tu

```

Figure 14. Entry of a Table with Unformatted Columns and Blank Entries

<u>Time</u> <u>(sec.)</u>	<u>Temp.</u> <u>(F)</u>	<u>Press.</u> <u>(psia)</u>
10.0	74.3	14.7
20.0	93.2	17.9
30.0		23.2
40.0		37.4

Figure 15. TEXT Printout of Figure 14

Table entries may also be left blank. The user specifies a blank entry by striking two RETURN keys in succession. Thus there will be four RETURN keystrokes in succession: the normal two to terminate the previous entry, and two to indicate that the current entry is blank. Blank entries are shown in Figures 14 and 15.

It should be noted that formatted and unformatted columns may also be mixed in a single table.

Table Entry by Columns. TEXT also permits the entry of tabular data by columns. To enter tables by columns, type the following table-mode command:

ATTNtable;c;(n) END

where (n) has the same meaning as in entry by rows.

The number and location of columns is then specified in the table format specification (f...f... etc.) just as for table entry by rows, and the individual table entries are typed, starting with the leftmost column. The user follows each entry with two RETURNS, and signals the end of each column by typing the command

ATTNtend END

The table mode for entry by columns is ended by entering another format-mode command (tf, tu, etc.).

Figure 16 shows the same table that was used to demonstrate table entry by rows (Figure 11), but entered here by columns. The printout for this table will be the same as shown in Figure 12. Notice that everything else except the initial table mode command is the same as for input by rows: each table entry must be followed by two RETURNS, columns denoted by an f in the table format specification are automatically formatted by TEXT, the letter f in the format specification may be replaced by the letter u if the column is to be unformatted, and one or more blanks may precede each f (or u) if spaces are to be left between columns.

```

table;c
TABLE MODE--PROVIDE FORMAT
f.....f.....f.....
Feature

Two-tone paint

Thermostatic control

Hypervated spamfratch
tend
Brand A

Yes

Yes

Yes
tend
Brand B

Yes

Yes

No
tend
tu

```

Figure 16. Entry of a Table in Table Mode, by Columns

Miscellaneous Embedded Commands

TEXT provides several miscellaneous embedded commands to perform various functions. These include commands to keep lines of text together, to tag words or phrases, to define blocks of text, to insert a "stop code", to insert a number of blanks in a line, to skip lines, to eject the current page, and to type special characters. These are presented in the following paragraphs.

Keeping Lines Together

Frequently several lines of text must be printed on a single page without overflowing to a following page. Examples are computer listings and various types of tables that will be discontinuous if printed partly on one page and partly on the next. Control is accomplished by specifying the beginning and end of lines that are to be kept together on a page. When these lines are encountered during printout, TEXT determines if enough space is left on the page. If adequate space remains, TEXT prints the lines; if there is not enough room, TEXT ejects the current page and prints the lines at the top of the next page. The user specifies the beginning of such lines by typing the command

(ATTN)t(RET)

and the end of the lines by typing the command

(ATTN)t(RET)

Everything between these two commands will be printed on the same page. The user may also end a group of lines to be kept together and begin another group with a single command

(ATTN)t(RET)

An example of enclosing lines of text with these commands is shown in Figure 17, where they are used to ensure that the table will be printed on the same page. It should be noted that a block callout (tc command) is not allowed within lines of text to be kept together.

```
t(  
table;r  
TABLE MODE--PROVIDE FORMAT  
f..... f..... f.....  
Feature  
  
Brand A  
  
.  
.  
.  
  
Hypervated spamfratch  
  
Yes  
  
No  
  
t)  
tf
```

Figure 17. Entry of a Table that is to be Printed on One Page

There are two minor variations of the command that begins a group of lines to be kept together. The first variation allows the user to specify the spacing of the group of lines, independent of the rest of the document. The spacing allowed and the command variation to use for each spacing are shown below:

<u>Spacing of Lines</u>	<u>Command</u>
Single-space	ⒶTTN)t(sⒶ
Double-space	ⒶTTN)t(dⒶ
Triple-space	ⒶTTN)t(tⒶ

These commands override the spacing specification of the print command. They determine the spacing of the group of lines to be kept together, regardless of the spacing of the rest of the document.

The second variation of this command is allowed only if one of the spacing options has also been specified. Thus this variation of the command can have one of three forms:

ⒶTTN)t(snⒶ
 ⒶTTN)t(dnⒶ
 ⒶTTN)t(tnⒶ

where n indicates that the section does not need to be kept together on the same page. This appears to be contradictory – a "keep-together" section that does not have to be kept together – but it really means that the command has been introduced to define a section of text with specified spacing (s, d, or t), not as a section that is to be kept together. Thus the n option is meaningful only in conjunction with the s, d, or t option.

Keeping a Phrase Together

Sometimes a group of words should not be split across output lines. (Typically such a phrase is a name, city, or math equation.) To keep a phrase together, regardless of the number of words in the phrase, this command is given:

ⒶTTN)tk:(phrase)ⒶTTN
 or
 ⒶTTN)tk:(phrase)Ⓐ

Tags

TEXT allows any word or phrase within a document to be "tagged" for easy access at a later time. For example, the user can tag every appearance of the phrase "Figure 5"; then if the figure number has to be changed, the figure references can be found without a line-by-line search of the entire document. Tags are also useful in compiling an index or table of contents, as described below. The command to tag a word or phrase is

ⒶTTN)tg:(phrase)ⒶTTN

Notice that this command ends with an ATTN key instead of the normal RETURN key, since a RETURN will automatically end the line. The use of the ending ATTN allows the user to continue entering a line after a word or phrase has been tagged. TEXT responds to this second ATTN key by backspacing and underlining the last character of the tagged phrase. For example, the tagging of the phrase "Table 7" in the following line appears at the terminal as:

Reference to tg:Table 7 shows that the GNP for

Every occurrence of the same phrase may be tagged wherever it appears in the document, and then located and edited later via the "find" command (described later in this chapter under "System Commands").

There are two variations of the tag command that facilitate automatic compilation of an index or a table of contents. To tag a word or phrase that is to be included in an index, use the following variation of the tag command:

ⒶTTN)ti:(phrase)ⒶTTN

An example of entering an index tag is shown below:

```
the ti:secured bonds get preferential treatment, since
```

The phrase "secured bonds" will then appear in the document index, along with the line number or page number of every tagged occurrence of the phrase. (The command used to actually compile the index is described later in this chapter under "Index Command".)

To tag a title that is to be included in a table of contents, use the following variation of the tag command:

```
(ATTN)tt(n):(phrase)(ATTN)
```

where (n) is an optional digit from 0 through 9, indicating the title level; 0 is the highest or most significant level, and 9 is the lowest or least significant. The title levels control the indentation of titles in the table of contents; titles of level 0 are indented the least, and those of level 9, the most. If no digit is supplied, TEXT assumes a value of 0. An example of entering a table of contents tag is shown below:

```
tt1:Section 4
tt2:Paragraph 1
The benefits of this policy are not payable in the event the insured
```

The title "Section 4" will then appear in the table of contents at the first indentation, and the phrase "Paragraph 1" will appear at the second indentation. (The command used to actually compile the table of contents is described later in this chapter under "Index Command".)

Blocks

The user can define a block of text somewhere within a document, and then insert that block anywhere else in the document. This saves the user from having to repeatedly type the same block of text. The block of text is given a "name", and then inserted by that name wherever desired. To define such a block, precede it with a command of the form

```
(ATTN)tb;(name)(RET)
```

where name is the name to be assigned to the block so that it may be referenced later. The user then enters the desired block of text and terminates it with the command

```
(ATTN)te(RET)
```

A block must stand alone; it cannot begin or end in the middle of a line. In addition, "keep together" – t(and t) commands – are not allowed within a block definition.

To insert the block later, type a command of the form

```
(ATTN)tc;(name)(RET)
```

where name is the same as above. TEXT will insert the block of text wherever this command is encountered; there is no limit to the number of times the same block may be inserted. The block will adopt the format mode (formatted mode, unformatted mode, etc.) in effect at the point of insertion, unless the block itself contains another format-mode command. Note that the block is inserted during printout only at the points where it is referenced. It is not inserted where it is defined.

Variations of the block insertion command allow the user to specify where the block of text is to be inserted on a page. These variations are described in Table 5. Notice that some variations of the command (those with the options any, bot, and top) do not require that the block be inserted immediately, but allow TEXT to insert it on the current page or the following page.

Table 5. Variations of the Block Callout Command

Command	Meaning
<code>(ATTN)tc;(name)␣</code>	Insert the named block at this point. (Part of the block may be printed on subsequent pages if not enough room on current page.)
<code>(ATTN)tc;(name);unc␣</code>	Insert the named block at this point if enough space is left on the page; otherwise eject the current page and print the block at the top of the next page.
<code>(ATTN)tc;(name);bot␣</code>	Insert the named block at the bottom of this page if there is room; otherwise print it at the bottom of the next page.
<code>(ATTN)tc;(name);top␣</code>	Insert the named block at the top of the next page.
<code>(ATTN)tc;(name);any␣</code>	Insert the named block at the bottom of this page if there is room; otherwise continue printing on this page and print the block at the top of the next page.

Stop Codes

To stop printing at any point during printout of a document, the user must enter the "stop code" command where the stop is to occur. This command has the form

`(ATTN) (ATTN)`

When this command is entered at the terminal, the first ATTN produces the usual underline and the second ATTN advances the typing element one space; consequently the command appears as a single underline.

A stop code causes TEXT to stop during printout and wait for the user to type something and then signal to continue printing. The user instructs TEXT to resume printing by slowly striking the ATTN key twice. Stop codes are most frequently used in documents such as form letters, where many copies of the same document must be printed with only minor changes (the name of the addressee, for example). In these cases stop codes are entered where the variable information is to appear; when printing stops at these points, the user types the variable information and then resumes printing by striking the ATTN key twice. Any number of stop codes may be used in a document.

Figure 18 shows the use of a stop code in the entry of part of a document; notice that the stop code has been entered after the word "Dear". Figure 19 shows a printout of this partial document, where the user has typed the name "Senator Claghorn" when TEXT stopped at the stop code and has then struck the ATTN key twice to resume printout. Many copies of this letter can be printed, with the user typing the name of the addressee and with the system automatically printing the rest of the letter. (Also see "Form-Letter Mode" under "System Commands" later in this chapter.)

There are two restrictions in the use of the stop-code command. First, it should not be used in formatted portions of a document, since it will throw off the formatting of the printed line. (The user can insert a stop code, however, with a variation of the insert-blanks command described next.) Second, stop codes `(ATTN) (ATTN)` in a heading or footing indicate automatic page numbering, as described earlier in this chapter.

```

t

Dear _:

f
    The Flimgratz Manufacturing Company is deeply concerned about the Gizmo Bill
    currently under consideration by the Senate. Passage of this bill would unduly
    penalize us as one of the world's only manufacturers of genuine leather Flimgratzes,
    while at the same time rewarding our competitors who produce cheaper (and, we feel,
    inferior) plastic "imitation Flimgratzes." We feel that this legislation is unfair
    and discriminatory and ....
    
```

Figure 18. Entry of Document Containing a Stop Code

Dear Senator Claghorn:

The Flimgratz Manufacturing Company is deeply concerned about the Gizmo Bill currently under consideration by the Senate. Passage of this bill would unduly penalize us as one of the world's only manufacturers of genuine leather Flimgratzes, while at the same time rewarding our competitors who produce cheaper (and, we feel, inferior) plastic "imitation Flimgratzes." We feel that this legislation is unfair and discriminatory and

Figure 19. TEXT Printout of Figure 18

Inserting Blanks

The user can insert extra blanks in a line by typing an embedded command of the form

`(ATTN)t*(n)(ATTN)`

where (n) is the number of blanks that are to be inserted. For example, the command to insert 10 blanks in a line is

`(ATTN)t*10(ATTN)`

Notice that this command ends with another ATTN key instead of a RETURN key, so that the user may continue with the same line after entering the command. The second ATTN key causes TEXT to backspace and underline the previous character. Thus the use of this command to insert, say, seven blanks will appear at the terminal as follows:

The Greek word for this phenomenon is t*Z, meaning

In general this command is used to leave space for later insertion of special characters or artwork. It can also be used when a stop code is employed in formatted printout as described below.

Notice that extra blanks can also be inserted simply by using the space bar. This may cause some of the blanks to be lost, however, since TEXT will start a new line with a word and will ignore any blanks left over from the previous line. On the other hand, TEXT always preserves blanks introduced with the t*(n) command, even if they fall at the beginning of a new line of printout.

A variation of this command can be used when the user wants to insert a stop code in formatted text. The user instructs TEXT to stop and informs it of the maximum number of characters to be inserted with the command

`(ATTN)t*(n)s(ATTN)`

where (n) is the maximum number of characters that the user will type when TEXT stops printing. For example, to stop printing and permit the user to insert a maximum of 15 characters, the command will be

`(ATTN)t*15s(ATTN)`

TEXT will stop printing when it encounters this command, just as for a stop code, and will assume that the user is going to type 15 characters. If the information to be typed contains fewer than 15 characters, the user should space out the remaining positions before resuming printout so that the line will still be correctly formatted. If the user enters more than 15 characters, however, the line will not be correctly formatted. For example, suppose the user enters the following text:

Congratulations to the t*12s family! You have been
chosen as one of only 500,000,000 families
to participate in our tremendous sweepstakes!

Printed with a line width of 55 characters per line, this will look like

Congratulations to the Rebysnap family! You have
been chosen as one of only 500,000,000 families to
participate in our tremendous sweepstakes!

where the operator typed the name "Rebysnap" and four spaces when TEXT stopped printing at the t*12s command, and then signaled TEXT to continue printing.

Skipping Lines

Blank lines may be skipped on a page to leave room for artwork, illustrations, tabular data, etc. TEXT provides a "skip" command for this purpose and allows the user to specify where on a page he wants the lines to be skipped. The basic form of this command is called the conditional skip command and has the form

$$\textcircled{\text{ATTN}}\text{t}+(n)\textcircled{\text{RET}}$$

where (n) is the number of lines to be skipped and must be a number less than 99. For example, the command to conditionally skip three lines will be

$$\textcircled{\text{ATTN}}\text{t}+3\textcircled{\text{RET}}$$

The effect of the conditional skip is to skip the specified number of lines if enough room is left on the page. If not, as many spaces as possible are skipped on the current page and the remaining skips are lost; normal printing resumes at the top of the next page. This command is normally used to leave extra spacing between sections of a document or before and after illustrations. Consequently conditional skips are usually small. They are not used to leave space for illustrations since all of the spaces may not be skipped.

Another use of this command is to signal the start of a new paragraph in entirely single spaced documents. In this case, the following command is entered between paragraphs:

$$\textcircled{\text{ATTN}}\text{t}+0\textcircled{\text{RET}}$$

A variation of the basic skip command will cause the system to always skip the specified number of lines, even if some are placed at the bottom of one page and some at the top of the next. This command has the form

$$\textcircled{\text{ATTN}}\text{t}+(n);\text{now}\textcircled{\text{RET}}$$

where (n) is again the number of lines to be skipped. This command is used where a specific number of lines must separate two sections of the document, but the separating lines do not have to be all on the same page. It is not used to leave space for artwork that cannot be split over two pages. Note that this form of the skip command is equivalent to giving (n) $\textcircled{\text{RET}}$ key strokes, i.e., (n) actual blank lines.

Another variation, the unconditional skip command, forces TEXT to leave the specified number of lines all on one page. If sufficient room remains on the current page, the lines are skipped; if not enough room remains, the current page is ejected and the specified lines are skipped at the top of the next page. The unconditional skip command has the form

$$\textcircled{\text{ATTN}}\text{t}+(n);\text{unc}\textcircled{\text{RET}}$$

This command is normally used to leave space for insertion of illustrations, tabular data, or artwork, since the specified amount of space is always left on a single page.

All of the skip commands discussed so far cause the specified number of blank lines to be left at the point where the command is encountered. There are three other variations of the skip command that allow the user to designate where the spaces are to be left on a page and that allow TEXT to continue printing until that location is reached. These variations of the skip command are presented in Table 6 along with the three already described.

Table 6. Variations of the Skip Command

Command [†]	Meaning
$\textcircled{\text{ATTN}}\text{t}+(n)\textcircled{\text{RET}}$	Skip (n) lines immediately if sufficient room is left on the page; otherwise skip as many lines as possible on this page and ignore the rest.
$\textcircled{\text{ATTN}}\text{t}+(n);\text{now}\textcircled{\text{RET}}$	Skip (n) lines immediately if sufficient room is left on the page; otherwise skip as many lines as possible on this page and skip the rest of the lines at the top of the next page.

[†]Where (n) must be replaced by the number of lines to be skipped.

Table 6. Variations of the Skip Command (cont.)

Command†	Meaning
Ⓐt+(n);uncⓇ	Skip (n) lines immediately if sufficient room is left on the page; otherwise eject the current page and skip them at the top of the next page.
Ⓐt+(n);botⓇ	Skip (n) lines at the bottom of this page if sufficient room is left; otherwise skip them at the bottom of the next page.
Ⓐt+(n);topⓇ	Skip (n) lines at the top of the next page.
Ⓐt+(n);anyⓇ	Skip (n) lines at the bottom of the current page if sufficient room is left; otherwise skip them at the top of the next page.

Ejecting a Page

As previously mentioned, the number of lines skipped in a skip command must be 98 or less. If the number 99 is specified, TEXT ejects the current page (even though the specified page depth has not been reached) and begins printing at the top of the next page. Thus the page-eject command is

Ⓐt+99Ⓡ

This command is used whenever a portion of a document (such as a new chapter) must begin on a new page.

Special Characters

Occasionally a need arises for special characters not normally found on the standard type element. If a special character occurs often, the user can leave the appropriate number of blanks for each occurrence and then insert the required characters manually after the document is printed. If relatively few special characters are required, however, another type element with the appropriate special characters can be substituted during printout. To do this, the user defines the special characters in a command at the point where he wants the characters printed. When TEXT reaches this point during printout it stops printing and the user can substitute the appropriate type element and signal TEXT to print the special characters; he then replaces the original type element and signals TEXT to resume normal printout. The command to inform TEXT that special characters are to be printed has the form

Ⓐt#(characters)Ⓐt

where (characters) is as described below. Notice that this command is terminated with an ATTN key instead of a RETURN key, so that the user may continue the same line after entering the command. The second ATTN causes TEXT to backspace and underline the preceding character when the command is entered at the terminal.

The special characters to be typed can be defined in one of two ways, depending on whether the document is to be printed at the typewriter terminal or at the high-speed printer in the computer center. For printout at the typewriter terminal, the characters in the t# command above are those that are in the same position as the desired characters are on the special type element. For example, assume that a special type element exists with italic characters (in the same position as the standard characters) and that the user wants to type the word "never" in italics. When entering the document, he will type the following command at the point where he wants the italics:

Ⓐt#neverⒶt

During printout TEXT will stop at this command, and the user will replace the standard type element with the italic type element and strike the ATTN key twice to continue printout. TEXT will then print the word "never" in italics and pause for the user to replace the standard type element. Normal printout is resumed when the user strikes the ATTN key twice. This example is shown in Figure 20 as it appears when entered at the terminal and when printed by TEXT.

The second method of defining special characters is for use when the document is to be printed on the high-speed printer at the computer center, and involves a special auxiliary file that associates the standard characters in the t# command with special (hexadecimal) high-speed-printer characters. The creation of this auxiliary file is described later in this chapter under "Auxiliary Files".

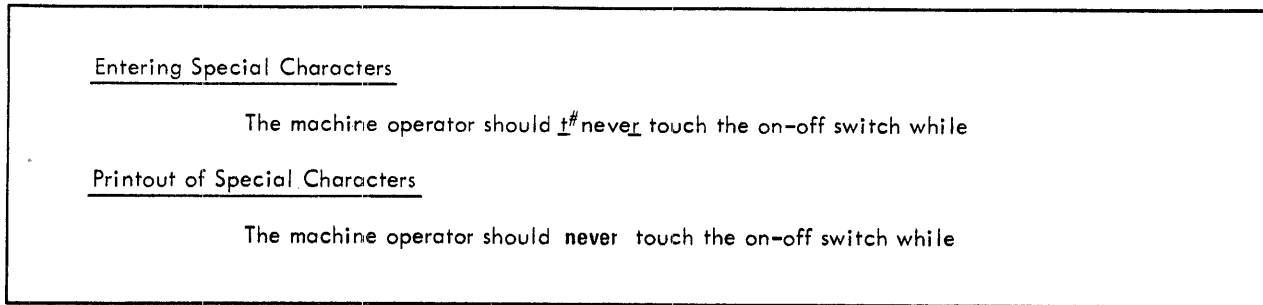


Figure 20. Entry and Printout of Special Characters

System Commands

Chapter 2 defined a system command as one that causes the system to perform an immediate function. Some of these system commands, including editing commands, the print command, and document storage and retrieval commands, were explained in Chapter 4. The remaining TEXT system commands are described below.

Width and Spacing Commands

Continuous Forms Option

Chapter 4 introduced the width command to set the line width for printout of formatted text and the page depth for all printout. An additional variation of the width command sets the physical page depth for continuous-form paper (that is, paper that comes in rolls instead of individual sheets). This variation has the form

$\textcircled{\text{ATTN}} w(\text{width});(\text{depth});(\text{cdepth})\textcircled{\text{RT}}$

where

width is the line width (characters per line) for printout of text entered in the formatted mode.

depth is the page depth (printed lines per page) for all printout.

cdepth is the physical page depth for continuous-form paper.

After printing a "page" on continuous-form paper, TEXT skips to the top of the next "page" and begins printing, without the usual stop for the user to change paper. The (cdepth) option on the width command enables the user to specify how much space to leave between the "pages" printed on continuous-form paper. TEXT will treat the continuous-form paper as though it were sheets of paper of the length specified by (cdepth). For example, if the specified page depth is 40 lines per page, and the specified continuous-form page depth is 50 lines per page, TEXT will print 40 lines and then skip 10 lines before starting printout of the next "page". If the (cdepth) option is not included in the width command, a continuous-form page depth of 66 lines per page is used by default (this corresponds to a standard 11-inch page). A print command option, discussed later in this chapter under "Print Commands", informs TEXT that continuous-form paper is being used for printout.

Temporary Width Setting Command

The system width setting of a document may be overridden during a printout operation with the command

$\textcircled{\text{ATTN}} tw(n)\textcircled{\text{RT}}$

where "n" is a number between 5 and 130 which designates the temporary width. To return to the current system width setting, the temporary width setting command is used with no options specified:

$\textcircled{\text{ATTN}} tw\textcircled{\text{RT}}$

Temporary Spacing Command

The user can modify the spacing option of a previous print command (see "Print Commands") with a temporary spacing command. The format of this command is

`(ATTN)ts; [s] d [t] (RET)`

where "s", "d", and "t" stand for single spacing, double spacing, and triple spacing, respectively. To revert to the spacing option originally specified in the print command, the temporary space command is issued with no options:

`(ATTN)ts(RET)`

Editing Commands

TEXT document editing commands described in Chapter 4 were:

<code>(ATTN)(line no.)(RET)</code>	to call out a specified line for editing.
<code>(ATTN)BCKSP...BCKSP</code>	to correct a typing error.
<code>(ATTN)RET</code>	to delete the current line.
<code>(ATTN)e(line no.)(RET)</code>	to erase a specified line.
<code>(ATTN)i(line no.)(RET)</code>	to insert a specified line.
<code>(ATTN)m(line no.)(RET)</code>	to move a specified line.

In each of these commands requiring a line number, the computer-assigned number had to be specified. This section introduces the concept of the current line pointer (CLP), which permits specifying line numbers by plus or minus offsets from a given line number, and describes the TEXT commands that may use the CLP. Also described is the "find" command, which searches through the entire document to locate lines by keyword, tagged word, or phrase.

CLP and Relative Line Numbers

The current line pointer (CLP) may be set to any existing line number selected by the user. Other line numbers may then be specified as plus or minus offsets from the CLP. For example, suppose that the CLP is set to line 100 of a document with lines numbered 99, 100, 101, 101.5, and 102; relative line number +1 refers to line 101, -1 refers to line number 99, +2 refers to line number 101.5, and +3 refers to line number 102. Specifying the relative line number as +0 or -0 indicates the line to which the CLP is set (in this example, to line number 100). Relative line numbers may be used in any TEXT command that requires line numbers, but must be preceded by a + or - sign for TEXT to distinguish a relative line number from an absolute line number; absolute line numbers are never preceded by a sign.

Editing by relative offsets from the CLP is frequently used for editing large portions in the middle of a document. For example, the user may set the CLP to the first line of the section to be edited (e.g., line 403) and then simply edit lines +0, +1, +2, etc. Relative line numbers are also convenient when moving sections of a document from the end of the document into the middle, as described below.

The command for setting the CLP to a specific line number is

`(ATTN)clp;(line no.)(RET)`

where (line no.) would be 403 for the preceding example. Table 7 shows some of the possible uses of offsets from the CLP in various TEXT editing commands.

The last example in Table 7 shows that relative and absolute line numbers may be mixed in a single command; TEXT distinguishes relative from absolute line numbers by the presence or absence of a sign (+ or -) preceding the number.

Table 7. Use of Relative Line Numbers in TEXT Commands

Command	Meaning
<code>(ATTN)+5(REF)</code>	Call out relative line number +5 (fifth line following CLP).
<code>(ATTN)-3; hte; the(REF)</code>	Replace "hte" with "the" on relative line number -3 (third line before CLP).
<code>(ATTN)e-37(REF)</code>	Erase relative line number -37.
<code>(ATTN)e+0;+23(REF)</code>	Erase relative line numbers +0 through +23.
<code>(ATTN)m+0(REF)</code>	Move the last line of document after relative line number +0.
<code>(ATTN)m-1;216;219(REF)</code>	Move absolute line numbers 216 through 219 after relative line number -1 (to precede the CLP).

If the CLP is not set by the user, TEXT sets it to the last line of the document. TEXT also automatically resets the CLP to the end of the document each time a new line is added to the document; however, typing a system command does not affect the CLP. The fact that the CLP automatically points to the last line of the document may be used to advantage, particularly in the move command. The procedure in Chapter 4 for creating a new paragraph or group of lines at the end of a document and moving them to the middle of the document required that the user first obtain (via the n command) the line number of the first line to be moved. By using relative line numbers with the CLP set to the end of the document, this step may be bypassed. For example, the following command will move the last three lines of the document to follow line number 46:

`(ATTN)m46;-2(REF)`

It is not necessary to know the actual line number of the third-to-last line of the document; only that it is line number -2 relative to the CLP. Note that the relative line number of the first line to be moved is always one less than the actual number of lines to be moved, because the count starts with line ±0. To obtain the relative line number for the first line to be moved, simply count back from the last line of the document, starting with 0, -1, -2, etc.

The command to reset the CLP from anywhere in the document to the end is

`(ATTN)clp;r(REF)`

In editing sequences that use the CLP and comprise several steps, the CLP should be set to the desired line number as the last step in the sequence; otherwise if an intervening step involves typing in a new line, TEXT will reset the CLP to the end of the document, cancelling the CLP setting selected by the user.

The user can determine the location of the CLP by typing the command

`(ATTN)clp(REF)`

A variation of the TEXT "find" command is also available to find the line number to which the CLP is currently set. This is described below.

Find Command

The TEXT find command locates

1. Any specified phrase, provided that the phrase is entirely on one input line.
2. "Tagged" words and defined blocks.
3. The position of the current line pointer.

It can be seen that this is a powerful capability in that it frees the user from having to know line numbers in performing editing functions. This will be particularly useful when changing, deleting, or replacing certain words or phrases that may be scattered throughout the document. However, it should be noted that editing by line number is generally faster and more efficient than gaining access via the search method. Line-number access is preferred whenever a single phrase is to be edited and the line number is known or is easily obtainable.

Finding a Phrase. To find a phrase anywhere in the document, the user merely specifies it in the find command and TEXT will start searching for it. The form of the basic find command is

`(ATTN)f;(phrase)(RET)`

When the basic command is used, the search begins at the line following the CLP and continues until either the word or phrase is found or the search returns to its starting point. If necessary, the search will "wrap around" from the end of the document to the beginning. If the phrase is not found, TEXT will print the message

`**NO SUCH PHRASE`

and the CLP will remain at its previous position. If the phrase occurs several times in the same document, the search will be terminated when the first occurrence of the phrase is found.

The user may want to confine the search to only a portion of the document. This can be done by specifying a range of line numbers in the find command, as follows:

`(ATTN)f(starting line no.)-(ending line no.);(phrase)(RET)`

If only one line number is specified, the search will start on that line.

Another form of the find command may be selected if the user wants to alter or replace a specified phrase:

`(ATTN)f;(old phrase);(new phrase)(RET)`

To delete a phrase, the specification following the second semicolon is simply left blank – the same procedure as in line editing (Chapter 4).

The user may want to find all occurrences of a phrase rather than just the next occurrence. The inclusion of the letter "a" (which stands for "all") in the command causes all occurrences to be found, and replaced if a new phrase is supplied:

`(ATTN)fa;(old phrase);(new phrase)(RET)`

Similarly, the addition of the letter "f" signifies that the first occurrence on each line be found, and replaced if a new phrase is supplied:

`(ATTN)ff;(old phrase);(new phrase)(RET)`

If a new phrase is not specified, then the line numbers of each occurrence will be printed in the following form:

`PHRASE FOUND ON LINE(S): xxxx. xxx xxxx. xxx`

If a new phrase is specified, the old phrase will be replaced for each occurrence found and the line numbers printed in the following form:

`PHRASE FOUND AND REPLACED ON LINE(S):
xxxx. xxx
xxxx. xxx`

Each line that is found (and optionally changed) can be displayed after its line number by specifying the "d" option; for example:

`(ATTN)fad;(old phrase);(new phrase)(RET)`

Note that the CLP is set to the line number of the last occurrence that is found. After the command is finished, the following message is printed:

`CLP AT LINE: xxxx. xxx`

Additional Variations of the Find Command. Variations of the find command permit the user to find and edit tagged phrases, titles, or block definitions without searching the entire document. These command variations are listed and described in Table 8. Notice that there is an additional character in each command to indicate the type of text to be found. The terms "tagged", "index-tagged", "title", and "block definition" as used in Table 8 have the following meanings:

1. A "tagged" phrase is any phrase that appears in an embedded command of the form

$\textcircled{\text{ATTN}}\text{tg}:(\text{phrase})\textcircled{\text{ATTN}}^\dagger$

2. An "index-tagged" phrase is any phrase that appears in an embedded command of the form

$\textcircled{\text{ATTN}}\text{fi}:(\text{phrase})\textcircled{\text{ATTN}}^\dagger$

3. A "title" refers to a phrase that is tagged as a title; that is, a phrase that appears in an embedded command of the form

$\textcircled{\text{ATTN}}\text{tt}(n):(\text{phrase})\textcircled{\text{ATTN}}^\dagger$

4. A "block definition" refers to text that is preceded by an embedded command of the form

$\textcircled{\text{ATTN}}\text{tb};(\text{name})\textcircled{\text{REF}}^{\dagger\dagger}$

and followed by the embedded command

$\textcircled{\text{ATTN}}\text{te}\textcircled{\text{REF}}^{\dagger\dagger}$

[†]See "Tags" earlier in this chapter.

^{††}See "Blocks" earlier in this chapter.

Table 8. Variations of the Find Command

Command	Meaning
$\textcircled{\text{ATTN}}\text{f};(\text{phrase})\textcircled{\text{REF}}$	Set the current line pointer to the first line containing a phrase.
$\textcircled{\text{ATTN}}\text{f};(\text{old phrase});(\text{new phrase})\textcircled{\text{REF}}$	Replace the first occurrence of a phrase with a new phrase.
$\textcircled{\text{ATTN}}\text{fa};(\text{old phrase});(\text{new phrase})\textcircled{\text{REF}}$	Replace all occurrences of a phrase with a new phrase.
$\textcircled{\text{ATTN}}\text{ff};(\text{old phrase});(\text{new phrase})\textcircled{\text{REF}}$	Replace the first occurrence of a phrase on each line with a new phrase.
$\textcircled{\text{ATTN}}\text{fad};(\text{old phrase});(\text{new phrase})\textcircled{\text{REF}}$	Replace all occurrences of a phrase with a new phrase; display found lines.
$\textcircled{\text{ATTN}}\text{fg};(\text{phrase})\textcircled{\text{REF}}$	Set the current line pointer to the first line containing a tagged phrase.
$\textcircled{\text{ATTN}}\text{fg};(\text{old phrase});(\text{new phrase})\textcircled{\text{REF}}$	Replace the first occurrence of a tagged phrase with a new phrase.
$\textcircled{\text{ATTN}}\text{fga};(\text{old phrase});(\text{new phrase})\textcircled{\text{REF}}$	Replace all occurrences of a tagged phrase with a new phrase.
$\textcircled{\text{ATTN}}\text{fi};(\text{phrase})\textcircled{\text{REF}}$	Set the current line pointer to the first line containing an index-tagged phrase.
$\textcircled{\text{ATTN}}\text{fi};(\text{old phrase});(\text{new phrase})\textcircled{\text{REF}}$	Replace the first occurrence of an index-tagged phrase with a new phrase.

Table 8. Variations of the Find Command (cont.)

Command	Meaning
<code>(ATTN)fia;(old phrase);(new phrase)(REF)</code>	Replace all occurrences of an index-tagged phrase with a new phrase.
<code>(ATTN)ft;(title)(REF)</code>	Set the current line pointer to the first line containing a title.
<code>(ATTN)ft;(old title);(new title)(REF)</code>	Replace the first occurrence of a title with a new title.
<code>(ATTN)fta;(old title);(new title)(REF)</code>	Replace all occurrences of a title with a new title.
<code>(ATTN)fb;(name)(REF)</code>	Set the current line pointer to the first line of a named block.
<code>(ATTN)fb;(old name);(new name)(REF)</code>	Replace the name of a block definition with a new name. This command has limited applicability, since it changes the name in the block definition but not the name in the command used to "call out" the block.

Finding the Current Line Pointer. The user can determine the location of the CLP simply by typing the command

`(ATTN)f(REF)`

This command causes TEXT to print the line number to which the CLP is set.

Printout Commands

Chapter 4 introduced the following print commands:

`(ATTN)p(REF)` to print a formatted copy without line numbers.
`(ATTN)pn(REF)` to print a formatted, line-numbered copy.
`(ATTN)pan(REF)` to print an as-entered, line-numbered copy.

Additional variations of the print command are described below; these variations permit the following:

- As-entered printout.
- Right-justified printout.
- Formatted printout with line numbers and underlines.
- Formatted printout with line numbers, but without underlines.
- Double-spaced and triple-spaced printout.
- Printout on continuous-form paper.
- Printout of parts of a document.
- Printout with page numbering starting at a specified number.
- Reprint of page just printed.

Also described below are commands for printing form letters and for printing on the high-speed printer at the computer center.

Print Commands

As-Entered Printout. A document may be printed exactly as it was entered, by typing the command

`(ATTN)pa(REF)`

Printout With Justified Right Margins. A document that has been entered in the formatted mode can be printed with right margins justified to a user-specified line width. TEXT accomplishes the justification by adding extra

Double-spaced and triple-spaced, line-numbered, and right-justified printouts may also be obtained as described below under "Combining Print Commands". The spacing option (including the single-space default) can be overridden by a subsequent temporary spacing command. (Refer to the previous section in this chapter entitled "Width and Spacing Commands".)

Printout on Continuous-Form Paper. The user may instruct TEXT to print on continuous-form paper (that is, paper on a long roll) by typing the command

`(ATTN)pc(RET)`

After one full page of printout, TEXT simply proceeds to the next page without the usual pause for the user to insert a clean sheet of paper. The individual printed pages may be cut apart later, or torn apart if perforated paper is used.

The number of blank lines between printed pages is determined by the difference between the printed page depth and the continuous-page depth previously specified by the user. For example, suppose the user had specified 30 printed lines per page and a continuous-form page depth of 51 lines, with the command

`(ATTN)w60;30;51(RET)`

When TEXT prints the last of the 30 lines, it then skips 21 blank lines (the difference between 51 and 30) before starting to print the next page. If the continuous-form page depth specified is less than the printed page depth, printed pages will be continuous and will not be separated by blank lines.

Printing on continuous-form paper can considerably speed up the printout of rough drafts, since no user attention is needed and no time is wasted changing paper.

Printing Parts of a Document. The user may instruct TEXT to begin printout at a specified line or phrase or to print only a specified range of lines. To print from line 30 through the end of the document, for example, the command is

`(ATTN)p30(RET)`

To start printout with the phrase "On the other side" on line 30, the command is

`(ATTN)p30;On the other side(RET)`

To print only lines 30 through 60, the command is

`(ATTN)p30-60(RET)`

It should be noted that if the printout is started in the middle of a paragraph of formatted text (that is, after the second paragraph input line) the rest of the paragraph will be printed against the left margin instead of being correctly indented. Succeeding paragraphs will have the correct indentation.

Printout Beginning at Specified Page Number. Recall that striking the ATTN key twice within a heading or footing (tuh and tuf formatting commands) causes TEXT to print the current page number for each page. When different parts of the same document are being developed and printed separately, however, it may be desirable to have the page numbering for later sections of the document start with a number other than page 1. To do this, type the command

`(ATTN)p;;(n)(RET)`

where (n) is the number at which page numbering is to start. For example, if the first page of the document is to be assigned page number 173, the command is

`(ATTN)p;;173(RET)`

This variation of the print command is meaningful only when a heading or footing is used with automatic page numbering on printout.

Reprint of Last Page. During printout of a document, TEXT stops printing at the bottom of each page to allow the user to insert another sheet of paper (except when printing on continuous-form paper). If the user discovers at this time that the page is inserted crookedly, improperly centered, or marred, it is not necessary to retype the entire document for a good copy of this page. The following command can be used to reprint the last page (with the same print options as in the original print command):

ⓐⓐpr ⓐⓐ

This command may also be used to restart printout if the user stopped printout in the middle of the page (by striking the ATTN key twice). In this case the entire page will be reprinted starting at the top of the page; it is not possible to resume printout at the point of interruption when printing has been stopped with the ATTN and RETURN keys. When the page has been reprinted, printout of the rest of the document continues under control of the original print command.

Combining Print Commands. Many of the print options that have been described can be combined into a single print command (except for the reprint option, r, which must always appear in a separate print command). For example, it is possible to use one print command to obtain a printout that is right-justified, with line numbers, without underlines, triple spaced, on continuous-form paper, starting at a line in the middle of the document, etc. The only restriction is that the print options must appear in a certain order in the command. The correct order of the options is shown in the general form of the print command in Appendix A. The reader should consult this appendix before attempting to combine print options in one command.

Form-Letter Mode

The form-letter mode enables the user to print multiple copies of a document (typically a form letter) without having to retype the print command each time. The command to initiate the form-letter mode is

ⓐⓐform ⓐⓐ

TEXT will acknowledge this command by a small movement of the type element. The user should then type the print command for the form letter, using any valid combination of options to describe how the letter is to be printed (for example, p for ordinary formatted printout, or pj for right-justified printout, etc.). TEXT will respond to the print command in the normal way by printing the message SIGNAL WHEN READY. The user should then insert a fresh piece of paper and strike the ATTN key twice to begin actual printout. TEXT will print to the end of the letter, will wait for the user to insert another piece of paper and strike the ATTN key twice, and will print another copy of the page. This process repeats itself, printing an additional copy after each ATTN ATTN, until the user types the following command to terminate printing of the form letter:

ⓐⓐ ⓐⓐ

The letter will normally contain stop codes so that the user can type name, address, date, etc., on each copy.

Printout on High-Speed Printer

A copy of the document currently in working storage may be printed at the computer on a high-speed line printer instead of at the typewriter terminal. The command for this is

ⓐⓐxp ⓐⓐ

This causes a formatted copy of the document to be printed at the high-speed printer, just as the basic print command does at the typewriter terminal. In addition, the print options described for the standard print command may also be used with this command. Some of the possible print options are shown below:

ⓐⓐxpa ⓐⓐ	Print, as entered.
ⓐⓐxpj ⓐⓐ	Print, right-justified.
ⓐⓐxpnw ⓐⓐ	Print, with line numbers, without underlines.
ⓐⓐxpd ⓐⓐ	Print, double-spaced.
ⓐⓐxpt ⓐⓐ	Print, triple-spaced.

It should be noted that the user's account must be authorized by the computer center to use the high-speed printer. If authorization has not been granted, the xp command will result in an error message.

Line Printer to Queue Command

TEXT automatically forces all line printer output into the symbiont queue immediately after an x-type print or report command is issued. The user can change to manual queueing mode with the command

`(ATTN)lp(RET)`

All subsequent printer output requires other "lp" commands to force the output into the queue. To return to automatic queueing mode, the command is

`(ATTN)lp:auto(RET)`

Any line printer output generated prior to this command is forced into the queue at the time the command is given, then all subsequent output is automatically enqueued.

Storage and Retrieval Commands

Chapter 4 discussed the store and get commands, to copy documents back and forth between working and permanent storage; the delete command, to delete a document from permanent storage; and the clear command, to clear working storage. This chapter presents variations of the store command that permit renumbering a document and storing it on an Edit-compatible disk file, and a variation of the get command to copy an Edit file into TEXT-compatible form. Also discussed are the transmittal of a document to magnetic tape at the computer center, and the formation and use of auxiliary files that permit the use of special characters and the compilation of a document index.

Store Command

Chapter 4 introduced the store command to copy a document from working storage to permanent storage. The basic form of this command (without document identification or password) is

`(ATTN)s;(name)(RET)`

where (name) is the name to be given to the stored document. The document identification and password options were also discussed in Chapter 4, as were TEXT responses to the store command and to various error conditions. Discussed below are several other variations of the store command, and the general form of the store command with all its legal options and their meanings.

Store a Portion of a Document. The user can store a portion of a document by specifying a range of line numbers, as follows:

`(ATTN)s(starting line no.)-(ending line no.);(name)(RET)`

Renumber a Document. The following variation of the store command instructs TEXT to store a document and renumber the document lines:

`(ATTN)sr;(name)(RET)`

where name is the document name. When renumbering, TEXT will assign the first document line the number 1,000 and will try to use whole number increments in assigning new line numbers (1, 2, 3, etc.). However, since the maximum line number allowed is 9999,999, documents which contain 10,000 or more lines will be renumbered using smaller increments. This command may be desirable if insertions, erasures, and/or rearrangement of the document have created many noninteger line numbers or many gaps in line numbers. (Note, however, that any line-numbered copy previously obtained will now be invalid, so that the stored-and-renumbered document will have to be retrieved and a new line-numbered copy printed if further editing is necessary.)

Edit-Compatible Files. A document in permanent storage is in reality stored on a RAD or disk-pack file. The form of this file is peculiar to TEXT, however, and makes the document inaccessible to the Edit processor, another

program available under the Control Program-Five System. To store a document on an Edit-compatible file, use the following form of the store command:

`(ATTN)sx;(name)(REF)`

where (name) is the name to be assigned to the Edit-compatible file. The lines (i.e., records) of this file will be numbered the same as the lines of the document. It should be noted that the document is copied intact to the Edit file, including any embedded ATTNs that may not be meaningful to other processors using the Edit file. The user should also be aware that the Edit processor limits the size of any line in the file to 140 characters.

Combining Store Commands. Many of the storage options that have been described (such as specifying document identification and passwords, renumbering, storing to Edit-compatible files, etc.) may be combined into a single store command. The only restriction is that the options must appear in a certain order, as shown in the store command general format in Appendix A. The reader should consult this appendix before attempting to combine store options.

Get Command

Chapter 4 introduced the get command to copy a document from permanent storage into working storage for editing or printing. The basic form of this command (without an account number or document password) is

`(ATTN)g;(name)(REF)`

where (name) is the name of the document to be retrieved. The document password option was also discussed in Chapter 4, as were TEXT responses to the get command and to various error conditions. Another variation of the get command is discussed below.

Retrieving Edit Files. A file created with Edit under CP-V is in a form not compatible with TEXT; however, such a file may be copied into working storage in a TEXT-compatible form with the following variation of the get command:

`(ATTN)gx;(name)(REF)`

where (name) is the name of the RAD or disk-pack file created by Edit. The line numbers of such a document are the same as the Edit line numbers for the original Edit file.

Auxiliary Files

Auxiliary files provide additional information about a document and must be "read in" along with the document as described below. The first line of an auxiliary file always defines the type of auxiliary file.

Special-Character Auxiliary File. As described earlier in this chapter under "Special Characters", TEXT permits the user to supply a string of standard characters in the command

`(ATTN)t#(characters)(ATTN)`

and then define, in a special-character auxiliary file, a special hexadecimal character string to be printed in its place on the high-speed printer at the computer center. The special-character auxiliary file is a document, created with TEXT and then stored in permanent storage just like any other document, in which the first line contains only the letter d, and succeeding lines contain special-character-string definitions of the following form:

`(standard characters):X'(hexadecimal digits)'`

where

(standard characters) is the character string in the t# command.

(hexadecimal digits) is a string of one or more pairs of hexadecimal digits (00 through FF) representing the hexadecimal code for the special printer characters.

No embedded commands are permitted in the auxiliary file. Once this type of special-character auxiliary-file document has been created, it must be "read in" with the actual document itself before printout of the document begins (as described below under "Reading in Auxiliary Files"). The same special-character auxiliary file may be used with several documents.

Each line in the auxiliary file may contain one or several special-character-string definitions; if a line contains several definitions, they must be separated by semicolons. For example, the definitions `mt:X'4C'`, `gt:X'6E'`, and `pe:X'4C7E'` can be specified in either of the following ways:

<code>mt:X'4C'</code> <code>gt:X'6E'</code> <code>pe:X'4C7E'</code>	}	one definition per line
<code>mt:X'4C';gt:X'6E';pe:X'4C7E'</code>		several definitions per line

Index-Word Auxiliary File. The user may also define another type of auxiliary file to facilitate compilation of a document index. This auxiliary file contains a list of words and is simply a document created with TEXT and stored in permanent storage just like any other document. The words may be those to appear in the document index, in which case the first line of the auxiliary file document must contain only the letter s; or they may be those to be excluded from a "full" index, in which case the first line must contain only the letter x. In the first case TEXT will search the entire document and will compile an index listing the line number or page number of every appearance of the words; in the second case TEXT will compile an index of all words in the document except those in the auxiliary file. In either case the auxiliary file document must be "read in" as described below.

As many lines as necessary may be used in the auxiliary file to list the indexing words. Words in the file may be listed one per line or several per line. If several words are entered per line, semicolons must be used for separation. For example, the index words "command", "embedded command", and "system commands" can be specified in either of the following ways:

<code>command</code> <code>embedded command</code> <code>system command</code>	}	one per line
<code>command;embedded command;system command</code>		several per line

The actual TEXT command to compile an index based on an index-word auxiliary file is described under "Index Command" later in this chapter.

Reading in Special-Character Auxiliary Files. Special-character auxiliary files reside in permanent storage just like any other document, and must be "read in" from permanent storage before they can be used. The command to read in a special-character auxiliary file is

`(ATTN)rd;(name)(RET)`

where (name) is the name of the file. The special-character definitions in this auxiliary file will then be used in printing the document in working storage. More than one special-character auxiliary file may be read in for use in printing a single document; the auxiliary files will be added together, and any conflicting definitions will be replaced by the latest definitions.

If the special-character auxiliary file was stored with a document password, use the following command form to read in the file:

`(ATTN)rd;(name);(document password)(RET)`

If the document is stored in another account, use the following command form to read in the file:

`(ATTN)rd;(name).(account number)(RET)`

Finally, if the document is stored in another account and has a document password associated with it, use the following command form to read it in:

`(ATTN)rd;(name).(account number);(document password)(RET)`

Reading in Index-Word Auxiliary Files. The index files reside in permanent storage just like any other document, and must be read in from permanent storage before they can be used. The commands to read in index-word auxiliary files are

`(ATTN)rs;(name)(RET)`

for files defining words for an index, and

`(ATTN)rx;(name)(RET)`

for files defining words to be excluded from indexing, where (name) in both cases is the name of the index-word auxiliary file. These files apply to an index to be compiled for the document currently in working storage, and must be read in before the index program is called via the index command (see "Index Command" later in this chapter). If more than one index-word auxiliary file is read in for compiling a document index, the files will be added together.

If the index-word auxiliary files were stored with document passwords, the commands to read them in are

`(ATTN)rs;(name);(document password)(RET)`
`(ATTN)rx;(name);(document password)(RET)`

If they are in another account, the commands are

`(ATTN)rs;(name).(account number)(RET)`
`(ATTN)rx;(name).(account number)(RET)`

Finally, if they are stored in another account and with document passwords, the commands to read them in are

`(ATTN)rs;(name).(account number);(document password)(RET)`
`(ATTN)rx;(name).(account number);(document password)(RET)`

Auxiliary files that have been read in may be cleared with the following commands:

`(ATTN)clear;d(RET)` for special-character files.
`(ATTN)clear;s(RET)` for index-word files.
`(ATTN)clear;x(RET)` for files of words to exclude from index.

In addition, the basic command to clear working storage also clears all auxiliary files that have previously been read in. This command is

`(ATTN)clear(RET)`

Use of Magnetic Tape

Any document may be transferred from permanent storage to magnetic tape at the computer center, and then retrieved from magnetic tape back into permanent storage at any future time.

Storing a Document. The command to store a document on magnetic tape is

`(ATTN)xt:(reel number);s;(name)(RET)`

where

(reel number) is from one to four characters (letters and numbers) and is used to label and identify the tape onto which the document is stored.

(name) is the name of the document in permanent storage; this same name will be associated with the document on the tape.

Storing a document on magnetic tape is similar to storing a document in permanent storage. That is, each document on tape has a name, and the tape is associated with the account number under which it was created.

If the document in permanent storage has a password to restrict access to it, the command to store it on magnetic tape is

`(ATTN)xt:(reel number);s;(name);(document password)(RET)`

It should be noted that the document on tape cannot have a password.

If the specified tape reel already has one or more documents on it, and it is desired to add this document to the others rather than replace them, the command is

`(ATTN)xta:(reel number);s;(name)(RET)`

or

`(ATTN)xta:(reel number);s;(name);(document password)(RET)`

if the document in permanent storage has a password.

Before using the xt command, the user should send the computer operator a message that magnetic tape is to be used, to give him time to locate and mount the required tape. (The message is discussed below under "Administrative Commands".) Use of the xt command also requires that the user's account be authorized by the computer center to use magnetic tapes. After the xt command has been issued, the terminal keyboard will be locked until the transfer to tape has been completed.

Retrieving a Document. To retrieve a document from magnetic tape into permanent storage, the command form is

`(ATTN)xt:(reel number);g;(name)(RET)`

where

(reel number) is the reel number assigned when the document was stored on the tape.

(name) is the name assigned to the document when it was stored on the tape; this same name will be assigned to the document when it is copied into permanent storage.

If the document is to be copied into permanent storage with a document password, then the command to retrieve it from magnetic tape is

`(ATTN)xt:(reel number);g;(name);(document password)(RET)`

If the tape was created in a different account, that account number must be supplied to retrieve a document from the tape. This command is

`(ATTN)xt:(reel number);g;(name).(account number)(RET)`

Finally, to retrieve a document that was stored under a different account and assign it a document password when it is copied into permanent storage, the command is

`(ATTN)xt:(reel number);g;(name).(account number);(document password)(RET)`

As with the command to store a document, the user should first send the computer operator a message that magnetic tape is required (via the xo command), and the user's account must be authorized by the computer center to use magnetic tapes.

Administrative Commands

Administrative commands do not have a direct effect on a document. They include commands to request the current terminal printout status (line width, page depths, tab settings); to obtain a report of all documents stored in permanent storage; and to send messages to the computer operator.

Terminal Status

The TEXT command to request the current status of the terminal (current line width, page depths, and tab settings) is

`(ATTN)?(RET)`

TEXT will respond by printing the current terminal printout status in the form

```
WIDTH 80  DEPTH 40  CONTINUOUS-FORM DEPTH 50
-----t---t---t-----t
      t   t   t           t
```

Here the first line shows the line width and page depths set with the width command (w80;40;50 in this case), the second line shows the system tab settings, and the third line shows the physical tab settings.

Storage Reports

The user may instruct TEXT to print a report of all standard documents in permanent storage in the user's or some other account. If no account is specified, the logon account is assumed. Documents stored with passwords and those stored in edit-compatible format will not be listed by this command. The report may be obtained either at the terminal or at a high-speed printer in the computer center.

Typewriter Terminal. To obtain a storage report at the terminal, the command is

`(ATTN)report[.(account)](RET)`

TEXT will respond to this command as follows

STORAGE REPORT FOR ACCOUNT: account

This will be followed by a listing of all nonpassworded documents in the account. Additionally, displayed with each document will be its document identification (if it was stored with one), the operator identification, the date it was stored, and the number of lines in the document. The report ends with a count of the number of documents listed.

High-Speed Printer. To obtain a storage report on the high-speed printer at the computer center, the command is

`(ATTN)xreport[.(account)](RET)`

The content of this report is the same as for a report printed at the terminal. The user must then pick up the report at the computer center or arrange to have it delivered or mailed if the computer center is remotely located. To use this command, the user account must be authorized to use the high-speed line printer.

Messages to the Computer Operator

The user may have to send a message to the computer operator to mount a magnetic tape or to instruct him on the delivery of printouts from the high-speed printer at the computer center. The command to transmit a message to the computer operator is

`(ATTN)xo;(message)(RET)`

where (message) is the actual message to the computer operator. The message will then be printed at the operator's console in the computer center.

Index Command

The index command instructs TEXT to compile various types of indexes and tables of contents for the document currently in working storage. The command to compile an index of certain words is

```
(ATTN)index;full;line(RET)
```

Then, if one or more auxiliary files of index words have been read in (see the `rs` command earlier in this chapter under "Auxiliary Files"), TEXT will compile an index with the line numbers of all words listed in the auxiliary file. If a file of words to be excluded from the index has been read in (see the `rx` command, also under "Auxiliary Files"), but not a file of index words, TEXT will compile an index with the line numbers of all words in the document except the words to be excluded. If neither type of auxiliary file has been read in, TEXT will compile an index of every occurrence of every word in the document, with line numbers.

To produce an index with page numbers instead of line numbers, merely replace the word "line" in the index command with the word "page" as shown:

```
(ATTN)index;full;page(RET)
```

An index of all index-tagged words and phrases (introduced by the `ti:` command) can be produced with the command

```
(ATTN)index;tag;line(RET)
```

In this case any auxiliary files read in are ignored, and an index of all index-tagged words and phrases in the document is compiled, with line numbers. For an index with page numbers, merely replace the word "line" in this command with the word "page".

Another variation of the index command will produce an index, with line numbers, of all TEXT block definitions (introduced by the `tb` command):

```
(ATTN)index;block;line(RET)
```

Again, replacing the word "line" with the word "page" will create an index with page numbers.

Finally, to compile a table of contents, the command is

```
(ATTN)index;title;line(RET)
```

This command will compile a table of contents of all titles tagged in the document (with the `tt(n):` command) with line numbers. For a table of contents with page numbers, replace the word "line" with the word "page". Titles will be indented in this table of contents according to their level (n), with level 0 the least indented and level 9 the most.

In all the above commands that compile an index or table of contents with page numbers, TEXT may be instructed to start the page numbering with a specific number by replacing "page" with "page(n)", where (n) is the number to be assigned to the first page of the document. If this number is omitted, TEXT assumes the first page of the document is page 1.

It should also be noted that all indexes by line are performed on the as-entered document whereas indexes by page are performed on the document as formatted for print. This is especially significant when blocks are included in a document. When this occurs, a line number index will find items in the block definition as opposed to a page index which will index on the block callouts.

As an example of the index procedure, suppose that the user wants to compile an index with page numbers of all appearances of the phrases "command", "embedded command", and "system command" in the document named TEXTREF in permanent storage. The complete procedure to do this (including building the auxiliary file) will appear at the terminal as:

```
clear                                clears working storage.
s                                     }
command;embedded command;system command } builds index-word auxiliary-file document.
```

s;indexwords
c|ear
rs;indexwords
g;TEXTREF
index;full;page

stores the file and names it "indexwords".
clears working storage again.
reads in auxiliary file.
gets main document.
compiles index.

File Output Commands

Output from the print, index, and report commands normally is directed to the user's terminal. Using the file output command, this output can instead be put to a specified permanent storage file in EDIT-compatible form. The general format of this command is:

file;(name)[;(password)]

where "name" is a valid CP-V file name, and "password" is used if the document is to be password protected.

The file output command must be given immediately prior to a print, index, or report command, or it has no effect.

If a document already exists with the same name, the following message is printed:

DUPLICATE NAME--DELETE OLD DOCUMENT?

If a "y^{es}" response is given, the old document is deleted, and the file command remains in effect; any other response cancels the file command.

6. TEXT SUBSYSTEMS

This chapter describes certain TEXT capabilities that are more extensive than simple commands.

Name/Address Files

TEXT provides the user with the capability of creating special name/address files. These files are stored TEXT documents, distinguished from other documents only by content. They usually (but not necessarily) consist of names and addresses, which can be merged into TEXT documents during printout. This is accomplished by using embedded TEXT commands (in the document referencing the file) that specify which field in the current name/address entry to use. For example, the full name or the last name only can be specified as the selected field. TEXT will extract the desired field and insert it at the appropriate place in the document. Any number of these commands may be used in a document.

Name/Address File Construction. When creating a name/address file, all material is entered in the unformatted mode, and must be textual information only; embedded commands are not allowed. Each line is entered starting at the left-hand margin.

Similar information should be in the same relative position within entries; for example, all names should be on the first line of each entry. Entries are separated by one blank line.

Lines containing non-printing information (ghost lines) are identified with a slash ("/") in column 1. These lines are ignored during all print operations and in the calculation of the line number of the name or city-state line. For example, the name line by default is the first non-ghost line.

The name line is normally the first line of an entry (although this may be changed using the form command, as described later in this section). The name line consists of a persons name (first, middle, last), preceded optionally by a prefix and followed optionally by a suffix. The first name is considered to be the character string immediately following the prefix (if any) and the last name is the character string immediately preceding the suffix (if any). For example:

Mr. John Doe, Jr.

Prefixes can be either standard (see Table 9) or non-standard. If standard, then the prefix starts in column 1 of the name line and is followed immediately by a space and the first name. If non-standard, the first character of the prefix must be an asterisk (*):

*xxx John Doe, Jr.

If the prefix consists of two or more parts normally separated by spaces, an asterisk should be used as the separator:

Capt. *&*Mrs. John Doe

Suffixes can also be standard (see Table 10) or non-standard. Standard suffixes are separated from the last name by a space or a comma, whereas non-standard suffixes require a comma. The comma used for this purpose must immediately follow the last name.

If the last name consists of two or more parts that are separated by a space, such as Van Dyke, an asterisk must be entered between the parts to indicate that they are a unit:

Van*Dyke

The city-state line is normally the last line of an entry in the name/address entry (although this may be changed using the form command; see below). The city-state line consists of the city, state, and ZIP code. TEXT assumes that the first characters on the last line are the city and that the last five numeric characters are the ZIP code. The last non-numeric characters on the line are considered to be the state. The city and the state must be separated with a comma to avoid possible ambiguities.

Table 9. Standard Prefixes

ADM	HON	PROF & MRS
Adm	Hon	Prof & Mrs
ADM.	HON.	PROF. & MRS.
Adm.	Hon.	Prof. & Mrs.
BRIG	LT	PVT
Brig	Lt	Pvt
BRIG.	LT.	PVT.
Brig.	Lt.	Pvt.
CAPT	MAJ	REV
Capt	Maj	Rev
CAPT.	MAJ.	REV & MRS
Capt.	Maj.	Rev & Mrs
COL	MISS	REV.
Col	Miss	Rev.
COL.	MR	REV. & MRS.
Col.	Mr	Rev. & Mrs.
CPL	MR & MRS	RT
Cpl	Mr & Mrs	Rt
CPL.	MR.	RT.
Cpl.	Mr.	Rt.
CPT	MR. & MRS.	SEN
Cpt	Mr.	Sen
CPT.	MRS	SEN.
Cpt.	Mrs	Sen.
DR	MRS.	SFC
Dr	Mrs.	Sfc
DR & MRS	MS	SFC.
Dr & Mrs	Ms	Sfc.
DR.	MS.	SIR
Dr.	Ms.	Sir
DR. & MRS.	PFC	SISTER

Table 9. Standard Prefixes (cont.)

Dr. & Mrs.	Pfc	sister
FR	PFC.	SR
Fr	Pfc.	Sr
FR.	PROF	SR.
Fr.	Prof	Sr.
GEN.	PROF.	
Gen.	Prof.	

Table 10. Standard Suffixes

ASST	RET
Asst	ret
ATTY	RN
Atty	rn
CPA	SR
cpa	Sr
DDS	Treas
dds	USA
DVM	usa
dvm	USAF
ESQ	usaf
Esq	USMC
II	usmc
III	UNS
IV	usn
JR	V. P.
Jr	2ND

Table 10. Standard Suffixes (cont.)

LLB	2nd
11b	3RD
MD	3rd
md	4TH
PRES	4th
Pres	

Tstop Command. This command causes the specified field from the name/address file to be inserted in the target document. The format of the command is:

Ⓐtstop;(field-id)Ⓐ

or

Ⓐtstop;(field-id)Ⓔ

where "field-id" is as follows:

<u>field-id</u>	<u>Meaning</u>
city	city
entry	entire entry
fname	first name
line(n)	line n of entry
lname	last name
name	entire name line
pname	prefix and last name
prefix	name prefix
state	state
suffix	name suffix
zip	ZIP code

Form Command. This version of the form command causes the specified name/address file to be associated with working storage. It also initiates a series of questions which allow the user to completely describe the various characteristics of the forthcoming form-letter printout. This includes:

- Specifying the relative line number within an entry where the various fields can be found (see "Tstop Command").

- Specifying which name/address entry is the first significant one of this run.
- Specifying which name/address file entries are to be selected for print-merge operations.

The command has the following format:

␣

where (name) is the name of the name/address file, (acct. no.) is the optional account number of that file, and (password) is the optional password.

TEXT responds to this command with the question

CHANGE FIELD LINE NUMBERS?

The user answers with

...which indicates no change, or

y ...which indicates that a field line number change is desired.

If the answer is "y", TEXT responds with

ENTER FIELD CHANGE CRITERIA

The field change criteria are entered in the following form:

(field id.);(line no.)...

The fields that can be changed and their default values are:

<u>field id.</u>	<u>line no. default</u>
city	last line of entry
name	line 1
state	last line of entry

(It should be noted that changing the name field also changes fname, lname, pname, prefix, and suffix fields. Similarly, a change of city or state will affect the zip.)

TEXT next asks the question

ANY START-AT-ENTRY CRITERIA?

The user answers with

...which means start at the first name/address entry, or

y ...which indicates that start-at-entry criteria will be entered.

If the answer is "y", TEXT responds with

ENTER START SELECTION CRITERIA

The selection criteria is entered in the following form:

(field id.);(condition);(string)...^(RT)

where (field id.) is the same as for the tstop command, and (condition) is as follows:

<u>condition</u>	<u>meaning</u>
eq	equal
ne	not equal
lt	less than
gt	greater than
le	less than or equal
ge	greater than or equal

and where (string) is a character string used by the select function for comparison purposes. All comparisons are made with character strings. Compares are done with special asterisks for non-standard items deleted (i.e., the compare string should look like the printed item rather than the item in its name/address entry).

TEXT then asks the question

ANY SELECTION CRITERIA?

The user answers with

- ^(RT) ...which means select the entire name/address file, or
- y^(RT) ...which indicates that selection criteria will be entered.

If the answer is "y^(RT)", TEXT responds with:

ENTER SELECTION CRITERIA

The selection criteria is answered in the same form as described above for start-selection criteria.

At this point, the requested name/address file has been associated to the document in working storage. The next command should be some form of the basic print command ("p"). Form letters are then printed with the appropriate information merged from the name/address file. At the end of each form letter, TEXT will pause and wait for an "^(ATN) ^(ATN)" response before proceeding with the next iteration.

Display-Oriented Editing

TEXT provides the user with a special editing capability intended exclusively for use on a display terminal. Currently, only the OMRON 8025 terminal is supported. See TEXT program description (706412-11B00) for interfacing to other terminals. It allows the user to display a range of lines on the screen, do local editing, and then reinsert the range back into working storage. This facility is invoked via the display command; its general format is as follows:

^(ATN)display[(starting line no.)-(ending line no.)];[add]^(RT)

where the specified range of text lines is to be displayed on the screen for local editing. If no line numbers are specified, the entire document is selected for editing. If only a starting line number is specified, the selected range for editing will be that line through the end of the document. The "add" parameter signifies that information is to be added to whatever is currently in working storage.

Command Operation. When the "display; add" command is entered, the screen is cleared and the cursor is positioned near the upper left corner. The user then keys in information utilizing any local editing features provided by the display unit. When all information has been entered or the screen becomes full, the user homes the cursor and hits the break key. TEXT then asks:

ANY MORE LINES TO ADD?

All responses cause the current screen information to be added to the end of working storage. A response of "y^{RET}" clears the screen and add mode continues after transmission is completed. All other responses terminate the display mode after transmission.

Display-oriented editing is initiated by keying in the "display; (range)" command. In doing so, a range of lines is selected for editing and is displayed on the screen. If too many lines were requested, only those lines that will fit on the screen are shown; the remaining lines are automatically displayed for editing in screen-sized blocks after current screen editing is completed. If at any time the user desires to interrupt the displaying of lines on the screen, the BREAK key may be hit; this creates the same condition as if end-of-screen had been encountered. Local editing may then proceed as described in the following paragraphs.

Note that the first two lines on the screen are not used. They are reserved for TEXT messages; any information typed there will be lost.

After the text lines have been displayed, the operator may change anything on the screen by using the movable cursor and the terminal editing features. These features usually include character and line replacement, insertion, and deletion.

When all screen editing is complete, the operator homes the cursor and presses the BREAK key. (It is important to remember to home the cursor prior to hitting the BREAK key. If this is not done, screen information can be destroyed.) TEXT then prints the following message to request further action:

NEXT ACTION - R(EPLACE)/A(DD)/E(DIT)/C(LEAR)?

The operator now types one of four characters followed by a RETURN. The characters have the following meanings:

- R Replace the original text range with that on the screen. Original screen lines will replace corresponding text lines. Insert lines will be assigned new line numbers in a manner consistent with normal sequential numbering; for example, a line inserted on the screen which falls between text lines 50 and 51 will be assigned a line number between 50 and 51.
- A Add the lines on the screen to the end of the document. The original text lines remain unchanged. If possible, the lines will be assigned consecutive integer line numbers starting with the first integer beyond the last working storage line; otherwise, smaller increments will be used.
- E Further editing is required. The original text lines will remain unchanged until further action is requested by pressing BREAK again.
- C Clear the screen and leave the original text lines unchanged.

Normally at the completion of the editing cycle for a screen, the next block of lines in the selected range will be displayed. To inhibit this action, follow the R, A, E, or C with an S (stop). If it is desired to extend the selected edit range, follow the R, A, E, or C with an N (next). This will reset the end of range to the end of document. Thus the general form of the response is:

$\left. \begin{matrix} R \\ A \\ E \\ C \end{matrix} \right\} \left\{ \begin{matrix} N \\ S \end{matrix} \right\}^{\text{RET}}$

Since it is easy to forget when the end of range has been encountered, TEXT will signal this condition with the message

END OF DISPLAY RANGE, DO YOU WISH TO CONTINUE EDITING?

A "y[Ⓞ]" response indicates a desire to continue editing. All other responses are considered negative.

Special Screen Conventions. Each document line will be preceded by an identifying line number when first displayed. The form of the number is "XX#Y". This number will be used to distinguish original document lines from new lines added during the local editing phase. If the original text lines are to be replaced with those on the screen, they will be used to match screen lines with working storage lines. New lines inserted on the screen should start at the left margin and should not have line numbers preceding them. Their relative screen position will be used to assign new working storage line numbers.

Document lines are terminated by a [Ⓞ] character. One document line may occupy several screen lines. When the lines are first displayed, each new document line will start on a new screen line.

The [Ⓞ] character used to identify commands will be represented on the screen with an "at" sign ("@"). Because of this convention, the character "Ⓞ" will be represented by the appearance of two consecutive "at" signs, i.e., "@@".

Since it is necessary to preserve the integrity of tab characters, all tabs will be represented on the screen in the following form:

@TTT...@

That is, all tab strings will be replaced by upper case T's bracketed by "@" signs.

Operational Restrictions. There are certain operational restrictions caused by common display terminal limitations:

- Caution should be exercised when editing documents which contain underscored words or phrases. This is because most displays have destructive overstriking (original character is destroyed when overstriking). An underscored phrase will not appear on the screen as such and therefore will not be transmitted back to TEXT properly.
- The operation of the transmit function is not consistent among the various different types of display units available today. One of these differences involves the way that logical lines are delimited from one another. A problem occurs with those displays that assume each screen line is a complete logical line when transmitting. In this case, if a text line is too long to be represented on a single screen line, it will be transmitted back as two or more separate text lines.

Error Messages. The following error messages may appear when display-oriented editing is in process:

**EMBEDDED COMMAND ERROR(S), DO YOU WISH TO REEDIT?

This message indicates that one or more embedded command errors were found on the current screen. A response of "y[Ⓞ]" signals a desire to reedit the screen. Any other response will cause the screen information to be accepted as is. The "@" sign preceding those commands with errors will not be translated into an [Ⓞ].

**TOO MANY LINES TO INSERT, DO YOU WISH TO REEDIT?

This message indicates that there is not enough room to insert the specified number of insert lines. A response of "y[Ⓞ]" signals a desire to reedit the screen. Any other response will cause the screen information to be accepted as is except that the critical insert range will be deleted.

**NOTHING TO DISPLAY

This message indicates that either working storage has been cleared or an empty range was specified for editing. The command action is terminated.

Batch Operations

A limited set of TEXT commands may be used in the batch mode. All input is considered to be commands, and no editing is allowed. Input that normally comes from the terminal is read through the M:SI DCB. Output that is normally directed to the terminal is written through one of the three DCB's as described below.

Batch Commands

The following TEXT commands are allowed in the batch mode:

```
clear
g
p
xp
report
xreport
index
form (only valid with name/address file)
w
-----
n
?
end
```

All input through M:SI is expected to be TEXT commands, with the first letter of the command in column one.

Output DCB's

Output will be directed through one of the following DCB's:

M:LO	p, report, and index commands. The output format is the same as if directed to a terminal. This DCB can be reassigned.
F:LO	xp, xreport commands. This DCB is directed unconditionally to the line printer. Output format is standard TEXT line printer format (i. e., overstrike buffer, etc.).
F:LL	all other TEXT output. This DCB can be reassigned.

Working Storage

The working storage area that is used for batch operations will not be saved after a job is complete, and will not conflict with the on-line working storage area.

Document Settings

The width, depth, and tab settings will be adjusted automatically after each "get" command.

APPENDIX A. SUMMARY OF TEXT COMMANDS

Table A-1 summarizes all embedded commands, which become part of an actual document; and Table A-2 summarizes all system commands, which cause immediate action and are not stored with a document. Commands are further subdivided within each table according to function. In addition, the following conventions are used in these tables to present the format of the commands:

1. ATTN symbol. This symbol means to strike the ATTN key.
2. RET symbol. This symbol means to strike the RETURN key.
3. BS symbol. This symbol means to strike the BACKSPACE key.
4. Parentheses (). Items in parentheses indicate data that must be supplied by the user. The parentheses are not typed as part of the command. For example, in the command shown as

(ATTN)(line no.)

the user must replace "(line no.)" with an actual document line number, such as

(ATTN)73(RET)

Other commands may require that the parenthesized data be replaced with a word or phrase instead of a number.

5. Brackets []. An item between brackets is optional. For example, in the following command format, the letter c is optional:

(ATTN)tu[c]

Thus this command may be typed in either of the following ways, depending on the command variation desired:

(ATTN)tu without c option

(ATTN)tuc with c option

If several items are placed one under the other between brackets, any one or none of the items may be typed. For example, notice the following command format:

(ATTN)t+(n);

now
unc
bot
top
any

 (RET)

If (n) is 5, this command can be typed in any of the following ways:

(ATTN)t+5(RET)

(ATTN)t+5;now(RET)

⋮

(ATTN)t+5;any(RET)

6. Braces { }. Items placed one under the other between braces require the selection of one of the items. For example, notice the following command format:

(ATTN)table;

r
c

 (RET)

This command can be entered in either of the following ways:

$\text{\textcircled{ATTN}}\text{table};r\text{\textcircled{REF}}$

$\text{\textcircled{ATTN}}\text{table};c\text{\textcircled{REF}}$

7. **Semicolons.** Semicolons are used in some commands to separate items. An example is the command

$\text{\textcircled{ATTN}}w[(width)];[(depth)];[(cdepth)]\text{\textcircled{REF}}$

The user must observe two rules when semicolons are used in a command:

- a. If an item is omitted at the end of a command, the semicolon preceding the item should also be omitted. Thus if "(depth)" and "(cdepth)" are omitted from the end of the above command and if "(width)" is 30, the command will be typed as

$\text{\textcircled{ATTN}}w30\text{\textcircled{REF}}$

- b. If an item between two semicolons is omitted, the semicolons should still be typed. Thus if "(depth)" is omitted in the above command and if "(width)" and "(cdepth)" are 30 and 80 respectively, the command will be typed as

$\text{\textcircled{ATTN}}w30;;80\text{\textcircled{REF}}$

Table A-1. Embedded Commands

Type of Command	Command	Meaning	Page
Document Format-Mode Commands	Formatted Mode		
	$\text{\textcircled{ATTN}}f\text{\textcircled{REF}}$	Request for formatted mode.	13
	Unformatted Mode		
	$\text{\textcircled{ATTN}}tu\text{\textcircled{REF}}$	Request for unformatted mode.	14
	$\text{\textcircled{ATTN}}tuc\text{\textcircled{REF}}$	Request for centered mode.	38
	General Form		
	$\text{\textcircled{ATTN}}tu[c]\text{\textcircled{REF}}$		
	Heading/Footing Modes		
	$\text{\textcircled{ATTN}}tuh(nn)\text{\textcircled{REF}}$	Request for basic heading mode.	39
	$\text{\textcircled{ATTN}}tuf(nn)\text{\textcircled{REF}}$	Request for basic footing mode.	40
	$\text{\textcircled{ATTN}}tuhr(nn)\text{\textcircled{REF}}$	Request for heading printed against right page margin.	42
	$\text{\textcircled{ATTN}}tufr(nn)\text{\textcircled{REF}}$	Request for footing printed against right page margin.	42
	$\text{\textcircled{ATTN}}tuhc(nn)\text{\textcircled{REF}}$	Request for heading to be centered.	42
	$\text{\textcircled{ATTN}}tufc(nn)\text{\textcircled{REF}}$	Request for footing to be centered.	42
	$\text{\textcircled{ATTN}}tuha(nn)\text{\textcircled{REF}}$	Request for heading to alternate, against right margin on odd-numbered pages and against left margin on even-numbered pages.	42
	$\text{\textcircled{ATTN}}tuha(nn)\text{\textcircled{REF}}$	Request for footing to alternate, against right margin on odd-numbered pages and against left margin on even-numbered pages.	42
	(nn)	Heading/Footing on	
	12	Every page.	40
	13	Every odd-numbered page.	40
	23	Every page except the first.	40
	24	Every even-numbered page.	40
	35	Every odd-numbered page except the first.	40
	General Form		
	$\text{\textcircled{ATTN}}tu\begin{bmatrix} h \\ c \\ a \end{bmatrix}[(nn)]\text{\textcircled{REF}}$		

Table A-1. Embedded Commands (cont.)

Type of Command	Command	Meaning	Page
Document Format-Mode Commands (cont.)	Table Mode		
	$\text{\textcircled{TT}}\text{table};; [n]\text{\textcircled{TT}}$	Request for table mode, with entry by rows. (n) specifies number of blank lines between output rows.	43
	$\text{\textcircled{TT}}\text{table};; [(n)]\text{\textcircled{TT}}$	Request for table mode, with entry by columns. (n) specifies number of blank lines between output rows.	47
	General Form		
	$\text{\textcircled{TT}}\text{table}; \begin{Bmatrix} r \\ c \end{Bmatrix}; [(n)]\text{\textcircled{TT}}$		
	End Row or Column		
	$\text{\textcircled{TT}}\text{end}\text{\textcircled{TT}}$	Signals end of a row or column.	44
	Column Specifications		
	$\dots f \dots f \dots f \text{\textcircled{TT}}$	Request for formatted columns within table mode.	43
	$\dots u \dots u \dots u \text{\textcircled{TT}}$	Request for unformatted columns within table mode.	44
$\dots u \dots f \dots u \text{\textcircled{TT}}$	Request for mixed formatted and unformatted columns within table mode.	47	
Miscellaneous Embedded Commands	Keeping Text Together[†]		
	$\text{\textcircled{TT}}t\text{\textcircled{TT}}$	Basic request to begin a portion of text that must be kept together on one page.	48
	$\text{\textcircled{TT}}t\text{\textcircled{TT}}$	Request to end text that is to be kept together.	48
	$\text{\textcircled{TT}}t\text{\textcircled{TT}}$	Request to end one portion of text to be kept together and to begin another.	48
	$\text{\textcircled{TT}}t\text{\textcircled{TT}}s$	Request to begin single-spaced text that must be kept together.	49
	$\text{\textcircled{TT}}t\text{\textcircled{TT}}sn$	Request to begin single-spaced text that does not have to be kept on one page.	49
	$\text{\textcircled{TT}}t\text{\textcircled{TT}}d$	Request to begin double-spaced text that must be kept together.	49
	$\text{\textcircled{TT}}t\text{\textcircled{TT}}dn$	Request to begin double-spaced text that does not have to be kept on the same page.	49
	$\text{\textcircled{TT}}t\text{\textcircled{TT}}t$	Request to begin triple-spaced text that must be kept together.	49
	$\text{\textcircled{TT}}t\text{\textcircled{TT}}(tn)\text{\textcircled{TT}}$	Request to begin triple-spaced text that does not have to be kept on same page.	49
	General Form to Begin Text to be Kept Together		
	$\text{\textcircled{TT}}t \begin{Bmatrix} s \\ d \\ t \end{Bmatrix} [(n)]\text{\textcircled{TT}}$		
	Keeping a Phrase Together		
$\text{\textcircled{TT}}tk:(phrase) \begin{Bmatrix} \text{\textcircled{TT}} \\ \text{\textcircled{TT}} \end{Bmatrix}$	Request to keep a phrase together, without splitting across output lines.	49	
Tags			
$\text{\textcircled{TT}}tg:(phrase)\text{\textcircled{TT}}$	Request to define a basic tag.	49	
$\text{\textcircled{TT}}ti:(phrase)\text{\textcircled{TT}}$	Request to define an index tag.	49	
$\text{\textcircled{TT}}tt[(n)]:(phrase)\text{\textcircled{TT}}$	Request to define a title tag of level (n), where (n) is 0-9.	50	

[†]The parentheses shown in these commands are part of the command and must be typed.

Table A-1. Embedded Commands (cont.)

Type of Command	Command	Meaning	Page				
Miscellaneous Embedded Commands (cont.)	Block Definitions						
	Ⓜb;(name)Ⓜ	Request to begin a block.	50				
	ⓂeⓂ	Request to end a block	50				
	Block Callouts						
	Ⓜtc;(name)Ⓜ	Request to call out (print) a block at this point.	50				
	Ⓜtc;(name);uncⓂ	Request to print a block at this point if it will fit; otherwise to eject the current page and print the block at the top of the next page.	50				
	Ⓜtc;(name);botⓂ	Request to print a block at bottom of first available page.	50				
	Ⓜtc;(name);topⓂ	Request to print a block at top of first available page.	50				
	Ⓜtc;(name);anyⓂ	Request to print a block at this point if it will fit; otherwise to continue to printout text and print the block at top of next page.	50				
	General Form for Block Callout						
	Ⓜtc;(name); <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>unc</td></tr><tr><td>bot</td></tr><tr><td>top</td></tr><tr><td>any</td></tr></table> Ⓜ	unc	bot	top	any		
	unc						
	bot						
	top						
	any						
	Stop Code/Insert Page Number						
ⓂⓂ	In ordinary text, request for TEXT to stop printing at this point.	51					
	In headings and footings, request to insert a page number at this point.	41					
Inserting Blanks							
Ⓜt*(n)Ⓜ	Request to insert blanks in a line.	53					
Ⓜt*(n);Ⓜ	Request to insert blanks and stop for user to manually type data during printout; allows the use of a stop code in formatted text and still have the formatting be correct.	53					
Skipping Lines							
Ⓜt+(n)Ⓜ	Request to skip a specified number of lines at this point if there is room; otherwise to skip to bottom of page.	53					
Ⓜt+(n);nowⓂ	Request to skip lines at this point; if there is not enough room on this page, the remaining lines will be skipped at top of next page.	53					
Ⓜt+(n);uncⓂ	Request to skip lines at this point if there is room; otherwise to eject current page and skip all lines at top of next page.	53					
Ⓜt+(n);botⓂ	Request to skip lines at the bottom of this page if there is room; otherwise at the bottom of the next page.	53					
Ⓜt+(n);topⓂ	Request to skip lines at the top of the next page.	53					
Ⓜt+(n);anyⓂ	Request to skip lines at the bottom of this page if there is room; otherwise at the top of the next page.	53					

Table A-1. Embedded Commands (cont.)

Type of Command	Command	Meaning	Page
Miscellaneous Embedded Commands (cont.)	<u>Skipping Lines (cont.)</u> <u>General Form</u> $\text{Ⓜ}t+(n); \left[\begin{array}{l} \text{now} \\ \text{unc} \\ \text{bot} \\ \text{top} \\ \text{any} \end{array} \right] \text{Ⓜ}$		
	<u>Ejecting a Page</u> $\text{Ⓜ}t+99\text{Ⓜ}$	Request to eject the current page.	54
	<u>Special Characters</u> $\text{Ⓜ}t\#(\text{characters})\text{Ⓜ}$	Request to define special characters.	54

Table A-2. System Commands

Type of Command	Command	Meaning	Page
Terminal Format Commands	<u>Terminal Type</u> $\text{Ⓜ}x\text{term}; \left\{ \begin{array}{l} \text{tty} \\ 2741 \\ 3010 \end{array} \right\} \text{Ⓜ}$	Request to change output format for different type of terminal.	12
	<u>Line Width and Page Depth Settings</u> $\text{Ⓜ}w(\text{width})\text{Ⓜ}$	Request to set line width to a specified number of characters per line.	16
	$\text{Ⓜ}w;(\text{depth})\text{Ⓜ}$	Request to set page depth to a specified number of lines per page.	16
	$\text{Ⓜ}w(\text{width});(\text{depth})\text{Ⓜ}$	Combined request to set line width and page depth.	16
	$\text{Ⓜ}w;;(\text{cdepth})\text{Ⓜ}$	Request to set physical page depth for continuous-form paper to a specified number of lines per page.	55
	<u>General Form</u> $\text{Ⓜ}w[(\text{width});[(\text{depth});[(\text{cdepth})]]\text{Ⓜ}$		
Terminal Format Commands	<u>Temporary Width and Spacing Commands</u> $\text{Ⓜ}tw[(n)]\text{Ⓜ}$	Overrides the system width setting of a document during printout; "n" is a number between 5 and 130.	55
	$\text{Ⓜ}ts; \left[\begin{array}{l} s \\ d \\ t \end{array} \right] \text{Ⓜ}$	Overrides spacing option of print command; single (s), double (d), or triple (t) spacing may be requested.	56
	<u>Tab Settings</u> $\text{Ⓜ}-----t-----t\text{Ⓜ}$	Request to set system tabs (to set physical tabs at the same time, depress the SET key before each t).	17
Editing Commands	<u>Calling Out Lines[†]</u> $\text{Ⓜ}(\text{line no.})\text{Ⓜ}$	Request to print a line and stop for editing.	23
	$\text{Ⓜ}(\text{starting line no.})-(\text{ending line no.})\text{Ⓜ}$	Request to print a group of lines.	23
	$\text{Ⓜ}(\text{line no.});(\text{old phrase});(\text{new phrase})\text{Ⓜ}$	Request to replace a phrase on a specified line with a new phrase.	24
	$\text{Ⓜ}(\text{line no.});(\text{phrase})\text{Ⓜ}$	Request to delete a phrase on a specified line.	25
	$\text{Ⓜ}(\text{line no.});(\text{phrase})\text{Ⓜ}$	Request to delete the rest of a line beyond a certain phrase.	25

[†] Line numbers in these commands may be the actual line numbers or offsets from the current line pointer (CLP).

Table A-2. System Commands (cont.)

Type of Command	Command	Meaning	Page
Editing Commands (cont.)	<u>Corrections on Current Line</u>		
	<code>ⓂⓈⓃⓈ . . . ⓂⓈⓃ</code>	Request to backspace and erase erroneous characters.	19
	<code>ⓂⓈⓃⓈⓂ</code>	Request to erase the current line.	19
	<u>Erasing Lines[†]</u>		
	<code>ⓂⓈⓃⓈ(line no.)ⓂⓈⓃ</code>	Request to erase a specified line.	27
	<code>ⓂⓈⓃⓈ(starting line no.)-(ending line no.)ⓂⓈⓃ</code> <code>ⓂⓈⓃⓈ(starting line no.);(ending line no.)ⓂⓈⓃ</code>	Request to erase a group of lines.	27
	<u>Inserting Lines[†]</u>		
	<code>ⓂⓈⓃⓈi(line no.);[(increment)]ⓂⓈⓃ</code>	Request to insert one or more lines.	27
	<code>ⓂⓈⓃⓈiendⓂⓈⓃ</code>	Request to terminate insert mode.	27
	<u>Moving Lines[†]</u>		
	<code>ⓂⓈⓃⓈm(line no.)ⓂⓈⓃ</code>	Request to move the last line of the document to after a given line.	28
	<code>ⓂⓈⓃⓈm(line no.);(starting line no.)ⓂⓈⓃ</code>	Request to move a group of lines from the end of the document to after a given line.	28
	<code>ⓂⓈⓃⓈm(line no.);(starting line no.);(ending line no.)ⓂⓈⓃ</code>	Request to move a group of lines from the middle of the document to after a given line.	29
	<u>Find Command^{††}</u>		
	<code>ⓂⓈⓃⓈf;(phrase)ⓂⓈⓃ</code>	Request to set the current line pointer to the line containing a phrase.	58
	<code>ⓂⓈⓃⓈf;(old phrase);(new phrase)ⓂⓈⓃ</code>	Request to replace the first occurrence of a phrase with a new phrase.	58
	<code>ⓂⓈⓃⓈf;(starting line no.)-(ending line no.);(phrase)ⓂⓈⓃ</code>	Request to limit search to range of line numbers.	58
	<code>ⓂⓈⓃⓈfa;(old phrase);(new phrase)ⓂⓈⓃ</code>	Request to replace all occurrences of a phrase with a new phrase.	58
	<code>ⓂⓈⓃⓈff;(old phrase);(new phrase)ⓂⓈⓃ</code>	Request to replace first occurrence of a phrase on each line with a new phrase.	58
<code>ⓂⓈⓃⓈfad;(old phrase);(new phrase)ⓂⓈⓃ</code>	Request to replace all occurrences of a phrase with a new phrase, displaying the found line.	58	
<code>ⓂⓈⓃⓈfg;(phrase)ⓂⓈⓃ</code>	Request to set the current line pointer to the line containing a tagged phrase.	59	
<code>ⓂⓈⓃⓈfg;(old phrase);(new phrase)ⓂⓈⓃ</code>	Request to replace the first occurrence of a tagged phrase with a new phrase.	59	
<code>ⓂⓈⓃⓈfga;(old phrase);(new phrase)ⓂⓈⓃ</code>	Request to replace all occurrences of a tagged phrase with a new phrase.	59	
<code>ⓂⓈⓃⓈfgd;(old phrase);(new phrase)ⓂⓈⓃ</code>	Request to replace the first occurrence of a tagged phrase on each line with a new phrase, displaying the found lines.	58	
<code>ⓂⓈⓃⓈfi;(phrase)ⓂⓈⓃ</code>	Request to set the current line pointer to the line containing an index-tagged phrase.	59	
<code>ⓂⓈⓃⓈfi;(old phrase);(new phrase)ⓂⓈⓃ</code>	Request to replace the first occurrence of an index-tagged phrase with a new phrase.	59	
<code>ⓂⓈⓃⓈfia;(old phrase);(new phrase)ⓂⓈⓃ</code>	Request to replace all occurrences of an index-tagged phrase with a new phrase.	59	
<code>ⓂⓈⓃⓈft;(title)ⓂⓈⓃ</code>	Request to set the current line pointer to the line containing a title.	59	

[†]Line numbers in these commands may be the actual line numbers or offsets from the current line pointer (CLP).

^{††}In all the find commands the search begins at the current setting of the current line pointer.

Table A-2. System Commands (cont.)

Type of Command	Command	Meaning	Page
Editing Commands (cont.)	Find Command (cont.) <code>@ft;(old title);(new title);</code> <code>@fta;(old title);(new title);</code> <code>@fb;(name);</code> <u>General Form</u> <code>@f [g] [t] [b] [d] [(starting line no.)-(ending line no.)];</code> <code>[(phrase) (old phrase);(new phrase)]</code>	Request to replace the first occurrence of a title with a new title. Request to replace all occurrences of a title with a new title. Request to set the current line pointer to the first line of a named block.	60 60 60
	<u>CLP Positioning</u> <code>@clp;(line no.);</code> <code>@clp;</code> <code>@f</code>	Request to set the current line pointer to a specified line number. Request to reset the current line pointer to end of document. Request to print the value of the current line pointer.	57 57 58
	<u>Display Oriented Editing</u> <code>@display</code> <code>@display;add</code> <code>@display(starting line no.)-(ending line no.);</code> <code>@display(starting line no.);</code> <u>General Form</u> <code>@display[(starting line no.)-(ending line no.)];[add]</code>	Request to display document on screen for local editing. Same as above, and specifies that information is to be added to whatever is currently in working storage. Request to display a range of lines on screen for local editing. Request to display range of lines from line specified to end of document.	77 77 77 77
	<u>Line Printer to Queue</u> <code>@lp</code> <code>@lp;auto</code>	Request all line printer output not to be put into symbiont queue. All subsequent printer output requires other lp commands to force output into the queue. Request all line printer output to automatically go to symbiont queue.	64 64
Printout Commands	<u>Reprint Last Page</u> <code>@rpf</code>	Request to reprint last page.	63
	<u>Form-Letter Mode</u> <code>@farm</code>	Request to enter form-letter mode. This command should be followed by variation of the print command.	63
	<u>Begin Printout</u> <code>@@</code>	Request to begin printout or resume printout (after print command, after stop at stop code, etc.).	20
	<u>Stop Printout</u> <code>@st</code>	Request for printing to stop.	19

Table A-2. System Commands (cont.)

Type of Command	Command	Meaning	Page
Printout Commands (cont.)	Print Command		
	<code>@p</code>	Request for a final formatted printout (single-spaced, no line numbers, no right justification).	20
	<code>@pa</code>	Request for an as-entered printout.	60
	<code>@pj</code>	Request for a printout with right-justified margins.	61
	<code>@pnl</code>	Request for a formatted printout with line numbers, and with underlines under the first character of each input line.	22
	<code>@pnw</code>	Request for a printout with line numbers but without underlines under the first character of each input line.	61
	<code>@pd</code>	Request for a double-spaced printout.	61
	<code>@pt</code>	Request for a triple-spaced printout.	61
	<code>@pc</code>	Request for printout on continuous-form (roll) paper.	62
	<code>@p(starting line no.)</code>	Request to print from a specified line to the end of the document.	61
	<code>@p(starting line no.)-(ending line no.)</code>	Request to print a group of lines.	62
	<code>@p(starting line no.);(phrase on starting line)</code>	Request to print from a phrase to the end of the document.	62
	<code>@p(starting line no.)-(ending line no.);(phrase on starting line)</code>	Request to print from a phrase to a specified line.	62
	<code>@p;;(first page no.)</code>	Request for a printout with automatic page numbering beginning at a specified page; that is, the first page printed will have the specified page number instead of the usual page 1.	62
		General Form	
	<code>@p [a] [n] [d] [starting line no.] [c] [starting line no.)-(ending line no.]; [j] [r] [(phrase on starting line)]; [(first page no.)]</code>		
	Common Combinations		
	<code>@panl</code>	Request for an as-entered printout with line numbers.	60
	<code>@pd</code>	Request for a right-justified, double-spaced printout.	61
	<code>@pjtl</code>	Request for a right-justified, triple-spaced printout.	61
	<code>@pac</code>	Request for an as-entered printout on continuous-form paper.	62
	<code>@panc</code>	Request for an as-entered printout with line numbers, on continuous-form paper.	62
	Printer on High-Speed Printer		
	Same as print command, except substitute xp for p. Also, the n and c options are not valid.	Request for a printout at the line printer in the computer center.	63
	General Form		
	<code>@xp [a] [n] [d] [starting line no.] [c] [starting line no.)-(ending line no.]; [j] [r] [(phrase on starting line)]; [(first page no.)]</code>		
Storage and Retrieval Commands	Store a Document		
	<code>@s;(name)</code>	Basic request to store a document.	32
	<code>@s;(name);(document identification)</code>	Request to store a document with identification code.	33
	<code>@s;(name);;(document password)</code>	Request to store a document with a document password.	33
	<code>@s;(starting line no.)-(ending line no.);(name)</code>	Request to store a portion of a document.	64
	<code>@sx;(name)</code>	Request a renumber and store a document.	64
	<code>@sx;(name)</code>	Request to store a document in an Edit-compatible file.	65
		General Form	
	<code>@s [r] [x] [(starting line no.)-(ending line no.);(name); [(document identification)]; [(document password)]:</code>		

Table A-2. System Commands (cont.)

Type of Command	Command	Meaning	Page
Storage and Retrieval Commands (cont.)	<u>Retrieve (Get) a Document</u>		
	<code>@g;(name)</code>	Basic request to retrieve a document.	34
	<code>@g;(name);(document password)</code>	Request to retrieve a document that has a document password.	34
	<code>@g;(name).(account number)</code>	Request to retrieve a document from another account number.	34
	<code>@gx;(name)</code>	Request to retrieve a document from an Edit file.	65
	<code>@gx;(name);(document password)</code>	Request to retrieve a document that has a document password from an Edit file.	65
	<u>General Form</u>		
	<code>@g[x];(name)[.(account number)];[(document password)]</code>		
	<u>Delete a Document</u>		
	<code>@d;(name)</code>	Request to delete a document from permanent storage.	36
	<code>@d;(name);(document password)</code>	Request to delete a document that has a document password.	36
	<u>Clear Working Storage</u>		
	<code>@clear</code>	Request to clear working storage and all auxiliary files.	36
	<u>Transfer Document to/From Magnetic Tape</u>		
	<code>@xt:(reel number);s;(name)</code>	Request to store a document from permanent storage to a magnetic tape reel.	67
<code>@xt:(reel number);s;(name);(document password)</code>	Request to store a document with a document password on a magnetic tape reel.	67	
<code>@xta:(reel number);s;(name)</code>	Request to add a document to others already on the specified reel.	67	
<code>@xt:(reel number);g;(name)</code>	Request to retrieve a document from a magnetic tape reel.	67	
<code>@xt:(reel number);g;(name);(document password)</code>	Request to retrieve a document from a magnetic tape reel and give it a document password.	67	
<u>General Form</u>			
<code>@xt[a];(reel number);$\begin{matrix} s \\ s \end{matrix}$;(name);[(document password)]</code>			
<u>"Read In" Auxiliary Files</u>			
<code>@rd;(name)</code>	Request to read in auxiliary file containing special characters.	67	
<code>@rd;(name);(document password)</code>	Request to read in special-character auxiliary file with document password.	67	
<code>@rd;(name).(account no.)</code>	Request to read in special-character auxiliary file from another account number.	67	
<code>@rds;(name)</code>	Request to read in auxiliary file containing index words.	67	
Same options as rd command for document password and account number.			
<code>@rx;(name)</code>	Request to read in auxiliary file containing words to exclude from index.	67	
Same options as rd command for document password and account number.			
<u>General Form</u>			
<code>@rfs;$\begin{matrix} d \\ s \\ x \end{matrix}$;(name)[.(account number)];[(document password)]</code>			
<u>Clear Auxiliary Files</u>			
<code>@clear;d</code>	Request to clear any special-character auxiliary files previously read in.	67	
<code>@clear;s</code>	Request to clear any index-word auxiliary files previously read in.	67	
<code>@clear;x</code>	Request to clear any auxiliary files previously read in that contain words to exclude from an index.	67	

Table A-2. System Commands (cont.)

Type of Command	Command	Meaning	Page
Storage and Retrieval Commands (cont.)	<u>Name and Address Files</u> $\text{stm}n\text{stop};(\text{field-id})\text{stm}$ or $\text{stm}n\text{stop};(\text{field-id})\text{sm}$	Request to insert the specified field from the name/address file into the target document.	75
	$\text{stm}n\text{form};(\text{name})[.(\text{acct. no.})];(\text{password})\text{sm}$	Request to associate the specified name/address file with working storage.	76
	Administrative Commands	$\text{stm}n\text{m}$ $\text{stm}n\text{?}\text{sm}$ $\text{stm}n\text{end}\text{sm}$ $\text{stm}n\text{report}[(\text{account})]\text{sm}$ $\text{stm}n\text{xreport}[(\text{account})]\text{sm}$ $\text{stm}n\text{xo};(\text{message})\text{sm}$	Request to print the line number to be assigned to the next line the user enters. Request for terminal format status (line width, page depth, and tab settings). Request to end a terminal session. Request to print a storage report at the terminal. Request to print a storage report at the line printer in the computer center. Request to send a message to the computer operator.
Index Command	$\text{stm}n\text{index};\text{full};\text{page}\text{sm}^\dagger$ $\text{stm}n\text{index};\text{full};\text{line}\text{sm}$ $\text{stm}n\text{index};\text{tag};\text{page}\text{sm}^\dagger$ $\text{stm}n\text{index};\text{tag};\text{line}\text{sm}$ $\text{stm}n\text{index};\text{block};\text{page}\text{sm}^\dagger$ $\text{stm}n\text{index};\text{block};\text{line}\text{sm}$ $\text{stm}n\text{index};\text{title};\text{page}\text{sm}^\dagger$ $\text{stm}n\text{index};\text{title};\text{line}\text{sm}$ <u>General Form</u> [†] $\text{stm}n\text{index};\left[\begin{array}{l} \text{full} \\ \text{tag} \\ \text{block} \\ \text{title} \end{array}\right];\left[\begin{array}{l} \text{line} \\ \text{page} \end{array}\right]\text{sm}$	Request for a full index by page numbers. Request for a full index by line numbers. Request for a tag index by page numbers. Request for a tag index by line numbers. Request for a block index by page numbers. Request for a block index by line numbers. Request for a table of contents by page numbers. Request for a table of contents by line numbers.	70 70 70 70 70 70 70 70
File Output Command	$\text{stm}n\text{file};(\text{name})[(\text{password})]\text{sm}$	Request to route output of p, index, or report command to permanent storage file.	71

[†]For page numbering not beginning with page 1, replace "page" with "page (n)", where (n) is the page number of the first page of the document.

APPENDIX B. SUMMARY OF TEXT MESSAGES

A number of messages are printed by TEXT at the terminal. These messages are listed and explained in Table B-1. Note that messages preceded by asterisks (**) indicate error messages.

Table B-1. TEXT Messages

Message	Meaning
ACTION CANCELLED	The user has made an invalid request and it has been cancelled, or he has cancelled his own erase or move command.
ANY START-AT-ENTRY CRITERIA ?	This message appears following entry of a field change in response to a name/address form command. A \odot indicates that the first name/address entry is the start. A "y \odot " indicates that start-at-entry criteria will be entered.
ANY SELECTION CRITERIA ?	This message appears following entry of a start selection criteria input in response to a name/address form command. A \odot response indicates that the entire name/address file should be selected. A "y \odot " response indicates that selection criteria should be entered.
AFTER-- text ₁ --MOVE-- text ₂ ?	<p>This is a response to a request to move a given line, where</p> <p style="padding-left: 40px;">text₁ represents the first few words of the line that the moved line is to follow.</p> <p style="padding-left: 40px;">text₂ represents the first few words of the line to be moved.</p> <p>This allows the user to verify that he has specified the correct line. To actually move the line, the user should type the letter y (for yes) and then strike the RETURN key. To cancel the move request (if the wrong line has been specified), he should strike only the RETURN key.</p>
AFTER-- text ₁ --MOVE-- text ₂ --THRU-- text ₃ ?	<p>This is a response to a request to move a group of lines from one point to another, where</p> <p style="padding-left: 40px;">text₁ represents the first few words of the line that the moved lines are to follow.</p> <p style="padding-left: 40px;">text₂ represents the first few words of the first line to be moved.</p> <p style="padding-left: 40px;">text₃ represents the first few words of the last line to be moved.</p>

Table B-1. TEXT Messages (cont.)

Message	Meaning
AFTER-- text ₁ --MOVE-- text ₂ --THRU-- text ₃ ? (cont.)	This allows the user to verify that he has specified the correct lines. To actually move the lines, the user should type the letter y (for yes) and then strike the RETURN key. To cancel the move request (if the wrong lines have been specified), he should strike only the RETURN key.
AFTER-- text ₁ --MOVE-- text ₂ --THRU END OF DOCUMENT ?	<p>This is a response to a request to move a group of lines from the end of a document, where</p> <p style="padding-left: 40px;">text₁ represents the first few words of the line that the moved lines are to follow.</p> <p style="padding-left: 40px;">text₂ represents the first few words of the first line to be moved.</p> <p>This allows the user to verify that he has specified the correct lines. To actually move the lines, the user should type the letter y (for yes) and then strike the RETURN key. To cancel the move request (if the wrong lines have been specified), he should strike only the RETURN key.</p>
CHANGE FIELD LINE NUMBERS ?	This is a response to the name/address form command. A $\text{\textcircled{N}}$ indicates no change; A "y $\text{\textcircled{N}}$ " indicates that a field line number change is desired.
CLP AT LINE: xxxx. xxx	This indicates the position of the CLP after a find command has been issued and completed.
**DATA LOST--RETYPE REMAINDER OF LINE	The user has entered too long a line. The portion correctly received is printed at the terminal and entered into working storage. The rest of the line should be retyped.
**DOCUMENT ACCESS DENIED	The user has attempted to get or delete a document but has supplied an improper document password. The request should be repeated with proper password.
DOCUMENT DELETED	This is a response to a delete command. It confirms that the specified document has been deleted from working storage.
**DOCUMENT DOES NOT EXIST	The user has attempted to get or delete a document that does not exist, or has misspelled the document name in the get or delete request.
**DOCUMENT IN USE	The user has attempted to get a document that is currently being stored by another user. The request should be repeated in a few minutes.

Table B-1. TEXT Messages (cont.)

Message	Meaning						
DOCUMENT STORED	This is a response to a store command. It confirms that the specified document has been successfully stored in permanent storage.						
**DOCUMENT TOO LARGE	The user has attempted to retrieve a document that is too large for remaining working storage. Working storage should be cleared and the document request repeated.						
DO YOU WISH TO RETAIN OLD DOC ID?	The user is replacing an old document with a new one, but has not specified a new id. The old id may be retained by responding "y [Ⓜ] "; any other response deletes the old id.						
**DUPLICATE NAME--DELETE OLD DOCUMENT?	The user has attempted to store a document with a name that already exists. A response of "y [Ⓜ] " will delete the document and continue the store operation.						
**EMBEDDED COMMAND ERROR(S), DO YOU WISH TO REEDIT?	This message indicates that one or more embedded command errors were found on the current screen. A "y [Ⓜ] " response indicates that the user wishes to reedit the screen; any other response indicates the user does not wish to reedit.						
END OF DISPLAY RANGE, DO YOU WISH TO CONTINUE EDITING?	This message indicates that the user has reached the end of the display range. A "y [Ⓜ] " response indicates the user desires to continue; any other response indicates the user does not desire to continue.						
ENTER FIELD CHANGE CRITERIA	This message requests a field change in response to a name/address form command. The field change criteria are entered in the form (field id.);(line no.)... [Ⓜ] The field id may be city, name, or state.						
ENTER SELECTION CRITERIA	This message is in response to the name/address form command. See the ENTER START SELECTION CRITERIA message for the form of the selection criteria.						
ENTER START SELECTION CRITERIA	This message requests start selection criteria in response to a name/address form command. The start selection criteria is of the form (field id.);(condition);(string)... [Ⓜ] where (field id.) is the same as the tstop command, and condition is <table border="0" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th style="text-align: left;"><u>Condition</u></th> <th style="text-align: left;"><u>Meaning</u></th> </tr> </thead> <tbody> <tr> <td style="text-align: left;">eg</td> <td style="text-align: left;">equal</td> </tr> <tr> <td style="text-align: left;">ne</td> <td style="text-align: left;">not equal</td> </tr> </tbody> </table>	<u>Condition</u>	<u>Meaning</u>	eg	equal	ne	not equal
<u>Condition</u>	<u>Meaning</u>						
eg	equal						
ne	not equal						

Table B-1. TEXT Messages (cont.)

Message	Meaning										
ENTER START SELECTION CRITERIA (cont.)	<table border="1"> <thead> <tr> <th data-bbox="917 241 1031 273"><u>Condition</u></th> <th data-bbox="1079 241 1193 273"><u>Meaning</u></th> </tr> </thead> <tbody> <tr> <td data-bbox="950 294 982 325">lt</td> <td data-bbox="1079 294 1193 325">less than</td> </tr> <tr> <td data-bbox="950 346 982 378">gt</td> <td data-bbox="1079 346 1226 378">greater than</td> </tr> <tr> <td data-bbox="950 399 982 430">le</td> <td data-bbox="1079 399 1307 430">less than or equal to</td> </tr> <tr> <td data-bbox="950 451 982 483">ge</td> <td data-bbox="1079 451 1356 483">greater than or equal to</td> </tr> </tbody> </table> <p data-bbox="860 514 1356 577">The (string) is a character string used by the select function for comparison.</p>	<u>Condition</u>	<u>Meaning</u>	lt	less than	gt	greater than	le	less than or equal to	ge	greater than or equal to
<u>Condition</u>	<u>Meaning</u>										
lt	less than										
gt	greater than										
le	less than or equal to										
ge	greater than or equal to										
ERASE-- text ₁ --THRU-- text ₂ ?	<p data-bbox="860 619 1372 651">This is a response to an erase command, where</p> <p data-bbox="917 672 1372 735">text₁ represents the first few words of the first line to be erased.</p> <p data-bbox="917 756 1372 819">text₂ represents the first few words of the last line to be erased.</p> <p data-bbox="860 840 1388 1039">This allows the user to verify that he has specified the correct lines. To actually erase the lines, the user should type the letter y (for yes) and then strike the RETURN key. To cancel the erase request (if the wrong lines have been specified), he should strike only the RETURN key.</p>										
FOOTING MODE	<p data-bbox="860 1081 1372 1165">This is a response to a command to enter the footing mode. It confirms that TEXT is in this mode.</p>										
FORMATTED MODE	<p data-bbox="860 1207 1356 1291">This is a response to a command to enter the formatted mode. It confirms that TEXT is in this mode.</p>										
HEADING MODE	<p data-bbox="860 1333 1372 1417">This is a response to a command to enter the heading mode. It confirms that TEXT is in this mode.</p>										
**ILLEGAL ACTION	<p data-bbox="860 1459 1388 1543">The user has typed an illegal command or command option. Command should be retyped correctly.</p>										
**ILLEGAL BACKSPACE OVER TAB CHARACTER	<p data-bbox="860 1585 1388 1753">The user has attempted to delete a TAB character. The backspacing procedure is ignored, and the line remains as it was before the backspacing commands. TEXT prints this line at the terminal, and permits the user to retry the operation.</p>										
**ILLEGAL LINE NUMBER	<p data-bbox="860 1795 1372 1879">The user has typed a command using a line number that does not exist. Command should be retyped with proper line number.</p>										

Table B-1. TEXT Messages (cont.)

Message	Meaning
LINE xxxx DELETED	This is a response to a command to delete a line. It confirms that the line has been deleted.
NEXT ACTION – R(EPLACE)/A(DD)/E(DIT)/C(LEAR)?	The screen editing is complete and the user has homed the cursor and pressed the BREAK key. TEXT requests further action. The general form of the response to this message is $\left\{ \begin{array}{c} R \\ A \\ E \\ C \end{array} \right\} \left\{ \begin{array}{c} N \\ S \end{array} \right\} \text{RET}$
NEXT LINE NUMBER xxxx	This is a response to a successful erase, move, or get command, where xxxx represents the line number that will be assigned to the next line of text that the user enters.
**NEXT LINE NUMBER TOO LARGE	The user has entered the last possible line in working storage; that is, the next line entered will have a number larger than 9999.999. Before any more text can be entered, the document in working storage should be stored and working storage should be cleared.
**NO STORAGE AVAILABLE	The user has tried to store a document that is too large for the permanent storage available. One or more documents should be deleted from permanent storage before attempting to store this document.
**NO SUCH LINE	The user has attempted to call out a line that does not exist.
**NO SUCH PHRASE	An attempt has been made to find or replace a nonexistent phrase on a specified line.
**NO SUCH TAG	An attempt has been made to find or replace a tag that does not exist.
**NOT FOUND	The user has typed a find command referencing a phrase that is not in the document.
**NOTHING TO DISPLAY	This message indicates that either working storage has been cleared or an empty range was specified for editing. The command action is terminated.
**NOTHING TO PRINT	The user has requested a printout, but there is no document in working storage.
**NOTHING TO STORE	The user has typed a store command, but there is no document in working storage.

Table B-1. TEXT Messages (cont.)

Message	Meaning
<p>PHRASE FOUND AND REPLACED ON LINE(S):</p> <p>xxxx. xxx</p> <p>xxxx. xxx</p> <p>.</p> <p>.</p> <p>.</p>	<p>A phrase has been found and replaced on the specified lines using the find command.</p>
<p>PHRASE FOUND ON LINES xxxx. xxx xxxx. xxx. . .</p>	<p>A phrase has been found (but not replaced by a new phrase) on the specified lines.</p>
<p>RESTORE SAVED DOCUMENT WIDTH, DEPTH, TABS?</p>	<p>The width, depth, and tab settings of a saved document differ from the working storage settings. A response of "y[Ⓜ]" resets working storage settings; any other response retains working storage settings.</p>
<p>SAVED DOCUMENT FORMAT RESTORED</p>	<p>Working storage width, depth, and tab settings have been reset to match those of the saved document.</p>
<p>SIGNAL WHEN READY</p>	<p>This is a response to a user request for a print-out. The user can begin actual printout by striking the ATTN key twice, or can cancel the printout by striking the ATTN key and then the RETURN key.</p>
<p>STORE COMMAND INTERRUPTED NUMBER OF LINES STORED: xxx CONTINUE STORE?</p>	<p>The store command was interrupted prior to completion. The user may respond "y[Ⓜ]" to continue the store operation.</p>
<p>TABLE MODE--PROVIDE FORMAT</p>	<p>This is a response to a command to enter the table mode. It confirms that TEXT is in this mode. The user must now specify column widths and whether columns are to be formatted or unformatted.</p>
<p>**TOO MANY LINES TO INSERT, DO YOU WISH TO REEDIT?</p>	<p>This message indicates that there is not enough room to insert the specified number of insert lines. A "y[Ⓜ]" response indicates the user wants to reedit the screen. Any other response will cause the screen information to be accepted as is except that the critical insert range will be deleted.</p>
<p>**TOO MANY LINES TO MOVE</p>	<p>The user has attempted to move too many lines between any two lines of the document in working storage.</p>
<p>UNFORMATTED MODE</p>	<p>This is a response to a command to enter the unformatted mode. It confirms that TEXT is in this mode.</p>

Table B-1. TEXT Messages (cont.)

Message	Meaning
<p>WIDTH xx DEPTH yy CONTINUOUS-FORM DEPTH zz ---t-----t---</p>	<p>This is a response to a ? command, where</p> <ul style="list-style-type: none"> xx is the current line width. yy is the current page depth. zz is the continuous-form page depth. t is the location of a tab setting. <p>This information is also printed in response to a successful get command and after log-on to indicate the previous or default settings.</p>
<p>WORKING STORAGE FORMAT RETAINED</p>	<p>The width, depth, and tab settings of working storage have not been changed to match those of the saved document.</p>

APPENDIX C. TERMINAL REQUIREMENTS

TEXT requires the use of the IBM 2741 Communications Terminal or its equivalent (Dated 30, UCC 1035, etc.). In describing the various TEXT features, this reference manual assumes that the above type terminal is being used.

TEXT will also operate however, with teletype compatible terminals. On those terminals the following key substitutions may be made:

<u>2741 Type Terminal</u>	<u>Teletype Terminal</u>
ⓐ	Control-D
TAB	Control-I
BACKSPACE	Control-H

When interrupting printout using a teletype terminal, the BREAK key (not the Control-D sequence) should be hit just once.

INDEX

Note: For each entry in this index, the number of the most significant page is listed first. Any pages thereafter are listed in numerical sequence.

? command, 69

A

absolute line number, 56
account number, 10, vi, 32, 34, 68
account storage report, 69
adding one document to another document, 35
administrative commands, 69
advanced commands, 38
as-entered printout, 21, 60
asterisk, 10
ATTN ATTN, 51, 63
ATTN key, 3, vi, 4, 7, 9, 20, 23, 49, 51, 54, 63
automatic page numbering, 41, 62
auxiliary files, 65-67, 70

B

BACKSPACE key, 3, 4
backspacing, 19, 20
backspacing over tab character, 20
basic TEXT commands, 13
batch operations, 80
blank insertion, 50
blank lines, 40
blank lines between printed pages, 62
blank table entry, 47
block insertion command, 50
block of text, 50
building a new document, 18

C

centered mode, 38
character, vi
character string, vi
clear command, 36, 67
clearing tabs, 17
clearing working storage, 18
CLP, 56-58
clp command, 57
column headings, 46
column width, 44-46
combining print commands, 63
combining store commands, 65
command, vi
commands to TEXT, 3
computer, vi
computer center, vi

computer operator, vi
conditional skip command, 53
connection with the computer, 10
continuous-form paper, 55
continuous-page depth, 62
Control Program-Five, vii
correcting commands, 19
correcting strikeouts and underlines, 20
correcting text, 19
correction of most of line, 25
corrections while entering a line, 19
current line pointer, 56-59

D

d command, 36
defining special characters, 54
delete command, 36
deleting a document, 36
deleting a line, 19
deleting words or phrases, 25
depth, 11, 15, 41, 55, 62
determining next line number, 30
display/oriented editing, 77
document, vi
document formatting for printout, 13
document identification, 32, 33, 35
document name, 35
document storage, 4
duplicate document names, 33

E

e command, 27
Edit-compatible files, 64
editing, vi
editing a document, 23
editing commands, 56
ejecting a page, 54
embedded commands, 38-55, vi, 3, 21, 66
end command, 36
ending a terminal session, 36
entering a document, 19, vi
erase command, 27
erased line numbers, 31
erasing a single line, 26
erasing multiple lines, 27
error correction, 19
error messages, 93

Note: For each entry in this index, the number of the most significant page is listed first. Any pages thereafter are listed in numerical sequence.

F

f command, 58
fa command, 58
find command, 58
finding a phrase, 58
font, vi
footing mode, 39-41
form command, 63
form letter, 63, 50, 60, 64
form-letter mode, 63
format-mode command, 13
format-mode commands, 38
formatted mode, 6, 13
formatted mode, 19
formatted printout, 22

G

g command, 34, 35, 65
get command, 34, 35, 65
gx command, 65

H

heading and footing modes, 39-42
high-speed printer, 60, 63, 69

I

i command, 27
indentation, 61
index command, 70
index tag, 49
index-word auxiliary file, 66, 67
insert command, 27
inserted line number, 27
inserting blanks, 52
inserting/moving lines, 27
introduction, 1

J

justified right margin, 60, 61

K

keeping lines together, 48, 49
keyboard, 9

L

length of phrases, 25
line corrections, 23
line deletion, 19
line erasures, 26
line number, 5, vi, 19, 21, 27, 30, 55, 57
line width, 11, 15, 38, 55
line width setting, 16
lines per page, 11
LOCAL-REMOTE switch, 9
log-on procedure, 10, vi
logging off, 11, 37

M

m command, 29-31
magnetic tape, 67, 68
maximum line length, 65
message to computer operator, 69
middle of the page, 38
move command, 28-30, 51
moving a single line, 28, 30
moving embedded commands, 31
moving groups of existing lines, 29
moving multiple lines, 28
moving paragraphs, 31

N

name/address files, 72
new paragraph, 17
number of characters per line, 16
number of corrections, 20
number of lines printed per page, 16

O

OFF command, 37
ON-OFF switch, 9
organization of manual, 2
overview, 3

P

p command, 60-63, 20
pa command, 60
page code, 39, 40
page depth, 11, 15, 41, 55, 61
page depth setting, 16
page number, 41, 42
page eject command, 54

Note: For each entry in this index, the number of the most significant page is listed first. Any pages thereafter are listed in numerical sequence.

pan command, 21
paragraph indentation, 17, 18
parentheses, 16
parentheses in commands, 3
password, 10, vi, 32-35, 66-68
pc command, 62
pd command, 61
period, 34
permanent storage, 5, vi, 34, 65, 66
phrase, vii
phrase-substitution command, 23, 24
physical page depth, 55
physical tab settings, 11, 16, 34
pj command, 61
pn command, 61, 22
pnw command, 61
pr command, 63
precautions in moving groups of lines, 31
print as-entered, 21
print command, 20, 21, 60-62
print continuous, 62
print double-spaced, 61
print options, 63, 64
print triple-spaced, 61
print with line numbers, 61, 22
print without underlines, 61
printing copies with line numbers, 21
printing parts of a document, 61
printout, vii
printout beginning at specified page number, 62
printout commands, 60
pt command, 61

R

rd command, 66, 67
reading in index-word auxiliary files, 67
reading in special-character auxiliary files, 66
reel number, 67, 68
relative line number, 56, 57
renumbering a document, 64
replacement phrase, 28
replacing short words, 27
report command, 69
reprint of last page, 63
retrieving a document, 34, 35
retrieving a document from magnetic tape, 68
retrieving edit files, 64
RETURN key, 3, 4, 10, 23
rough drafts, 61
rs command, 67
rx command, 67

S

s command, 32, 33, 64, 65
semicolon, 16, 25, 26, 33, 65

semicolon within data to be edited, 26
setting tabs, 16, 17
setting up the format for printout, 18
shutdown procedure, 11
SIGNAL WHEN READY, 7, 20
skip command, 53
skipping lines, 53, 54
special characters, 54
special-character auxiliary file, 65
sr command, 64
standard prefixes, 73
standard suffixes, 74
start procedure, 9
stop code, 51
storage and retrieval commands, 64
storage report, 69
store and renumber command, 64
store command, 31, 32, 64, 65
storing a document, 30-32
storing a document on magnetic tape, 67
storing a nonexistent document, 32
strikeover, 20
string, vii
summary of TEXT commands, 83
summary of TEXT messages, 93
sx command, 65
system commands, 55-68, vii, 3
system tab settings, 11, 16, 35

T

t(command, 48, 49
t+ command, 53
t+99 command, 54
t) command, 48, 49
t# command, 54, 65
tab setting, 11, 15-17
table command, 43, 47
table entry by columns, 47
table entry by rows, 43
table mode, 43-47
table of contents, 50, 68
tag command, 49, 50
tags, 49, 50
tape reel number, 67, 68
tb command, 50
tc command, 50
te command, 50
tend command, 44
terminal, vii
terminal status, 69
text, vii
TEXT, vii
TEXT command summary, 83
tf command, 13
tg command, 49
ti command, 49
tt command, 50

Note: For each entry in this index, the number of the most significant page is listed first. Any pages thereafter are listed in numerical sequence.

tu command, 14
tuc command, 38
tuf command, 40-42
tuh command, 39-42
typing, vii

U

underline, 20
unformatted, 19
unformatted column, 46
unformatted mode, 6, 14
user, vii
user identification word, 10

W

w command, 55, 62
widow suppression, 17
width, 11, 16, 38, 55
width command, 55, 62
working storage, 5, vii, 35, 36, 67

X

xo command, 69
xp command, 63
xreport command, 69
xt command, 67, 68
xta command, 68

HONEYWELL INFORMATION SYSTEMS
Technical Publications Remarks Form

CUT ALONG LINE

TITLE

XEROX TEXT
LANGUAGE AND OPERATIONS
REFERENCE MANUAL

ORDER NO.

XM23-00

DATED

MAY 1976

ERRORS IN PUBLICATION

[Empty box for reporting errors in publication]

SUGGESTIONS FOR IMPROVEMENT TO PUBLICATION

[Empty box for providing suggestions for improvement to publication]



Your comments will be investigated by appropriate technical personnel and action will be taken as required. Receipt of all forms will be acknowledged; however, if you require a detailed reply, check here.

FROM: NAME _____

DATE _____

TITLE _____

COMPANY _____

ADDRESS _____

PLEASE FOLD AND TAPE—
NOTE: U. S. Postal Service will not deliver stapled forms



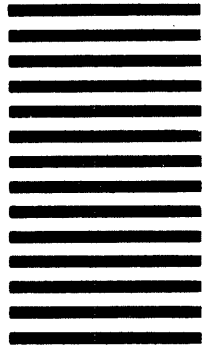
NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 39531 WALTHAM, MA02154

POSTAGE WILL BE PAID BY ADDRESSEE

HONEYWELL INFORMATION SYSTEMS
200 SMITH STREET
WALTHAM, MA 02154

ATTN: PUBLICATIONS, MS486



CUT ALONG LINE

FOLD ALONG LINE

FOLD ALONG LINE

Honeywell

Honeywell Information Systems

In the U.S.A.: 200 Smith Street, MS 486, Waltham, Massachusetts 02154

In Canada: 155 Gordon Baker Road, Willowdale, Ontario M2H 3N7

In the U.K.: Great West Road, Brentford, Middlesex TW8 9DH

In Australia: 124 Walker Street, North Sydney, N.S.W. 2060

In Mexico: Avenida Nuevo Leon 250, Mexico 11, D.F.

22228, 483, Printed in U.S.A.

XM23-00