

**TENET USERS MANUAL**

**EDITOR**

DEC 28 1970

## EDITOR COMMAND SET

- ENTER [record no.][STEP increment] p.15  
Causes automatic generation and prompting of record numbers.
- LIST [range]  $\left[ \begin{array}{l} \text{TEXT} \\ \text{NUM[BERS]} \end{array} \right]$  p.17  
Prints a listing of the specified records.
- DEL[ETE][range]  $\left[ \begin{array}{l} \text{LIST} \\ \text{TEXT} \\ \text{NUM[BERS]} \end{array} \right]$  p.18  
Deletes all records specified from the workspace.
- REP[LACE][range][STEP increment] p.19  
Enables the user to replace records within the workspace.
- REN[UMBER][start value][STEP increment] p.21  
Changes the record numbers of records within the workspace
- ALTER [range] p.22  
Enables the user to change record content by using a special set of control characters. The original record is printed before it can be edited.
- MOD[IFY][range] p.25  
Enables the user to change record content by using a special set of control characters. The original record is not printed at the terminal, only its record number appears.
- CH[ANGE]  $\left[ \begin{array}{l} \text{'string}_1 \text{'[position][ (occurrence) ]} \\ \text{position} \end{array} \right]$  TO  $\left[ \begin{array}{l} \text{'string}_2 \text{'[IN range]} \\ \left[ \begin{array}{l} \text{LIST} \\ \text{TEXT} \\ \text{NUM[BERS]} \end{array} \right] \end{array} \right]$  p.27  
Replaces one set of characters with another within the records specified.
- MOVE range [TO record no.][STEP increment] p.29  
Renumbers and moves the records specified to a new location in the workspace.
- DUP[LICATE][range][AT record no.][STEP increment][FROM 'filename'] p.30  
Duplicates and renumbers the records specified at another location in the workspace.
- LOAD [range] FROM 'filename [BASIC][CLEAR] p.32  
Retrieves a previously saved file from disc storage and loads it into the EDITOR workspace.

# **TENET USERS MANUAL EDITOR**

**OCTOBER 1970**

**TENET  
927 Thompson Place  
Sunnyvale, California 94086**

**Pub. No. 2003**

**Price: \$3.00**



## PREFACE

This document is a user's manual which describes in detail the features, vocabulary, and usage of EDITOR, a subsystem of the TENET 210 Time-sharing System. Other documents in this series are:

EXECUTIVE Users Manual  
TENET BASIC Users Manual  
FORTRAN IV Users Manual

*FIRST EDITION:*

*Specifications contained herein are subject to changes which will be reported in subsequent revisions. Copies of this and other TENET publications can be obtained through TENET branch offices. A form is provided at the back of this publication for readers' comments. If the form has been detached, please direct your comments to TENET, 927 Thompson Place, Sunnyvale, California 94086 (Attention: Software Publications Department).*



# CONTENTS

PREFACE	iii		
1 INTRODUCTION	1	DELETE Command	18
Files	2	REPLACE Command	19
2 USING EDITOR	3	RENUMBER Command	21
The Terminal	3	ALTER Command	22
Mode of Operation	3	MODIFY Command	25
Special Keys	3	CHANGE Command	27
LOGIN Sequence	5	MOVE Command	29
Correcting LOGIN Entries	5	DUPLICATE Command	30
Accessing EDITOR	6	LOAD Command	32
The Elements of EDITOR	6	SAVE Command	34
The EDITOR Workspace	6	FORM Command	36
Entering Information to EDITOR	6	LENGTH Command	38
Character Set	6	TABS Command	39
Record Length	7	QUIT Command	40
Blanks	7	Leaving the System LOGOUT	41
Syntax Error Checking and Recovery	7	4 SAMPLE TERMINAL SESSIONS	43
Glossary of EDITOR Terms	8	Example 1: Updating and Interrogating a Text File	43
3 EDITOR COMMANDS	13	Example 2: Preparation of a FORTRAN IV Program File	46
Introduction	13	Example 3: Bibliography Search by Topic	48
Conventions	13	APPENDIX A ANSI CHARACTER SET	51
ENTER Command	15	APPENDIX B MESSAGES	53
LIST Command	17	APPENDIX C MODEL 33 TELETYPE TERMINAL	57
		INDEX	61





# 1. INTRODUCTION

EDITOR is one of the subsystems of the TENET 210 Timesharing System. It consists of a set of sophisticated editing facilities which enable the user to create and manipulate the contents of text files.

- **Record Referencing by Record Number**  
Every record of information in the EDITOR subsystem is prefaced by a record number which may be used in any EDITOR command to reference a record or range of records within a file.
- **Record Reference by Content**  
The user may also reference a record or range of records within a file by specifying a sequence of characters within the records.
- **Records or ranges of records may be copied, moved, deleted, replaced, or inserted.**
- **Characters within records or ranges of records may be deleted, replaced, or inserted.**
- **File Concatenation and Segmentation**  
Files may be linked together with appropriate record renumbering, or individual files may be separated into multiple files.
- **Line Length Monitoring**  
The user may keep all records of a file within a prescribed length by using the LENGTH command which warns the user when he exceeds the specified maximum record length limit.
- **Tab Setting**  
The user may specify up to four tab stops to be set on the terminal.
- **Record Renumbering**  
The record numbers of a file may be resequenced, for example, after the file has been modified extensively.

## FILES

The TENET Timesharing System provides the user with a flexible means of creating, protecting, maintaining and using files. A file is composed of structured information sets, called records, maintained on a mass storage device (e.g., disc) external to the computer's central memory.

The TENET System offers the user considerable latitude in designing files that best suit his needs. When a file is created, the user determines its physical characteristics, usage within subsystems, and degree of privacy and protection. Text files acceptable to EDITOR may be created in the EXECUTIVE, TENET BASIC, FORTRAN IV, or EDITOR.

*For a complete description of file capacity allowed in the TENET system see the EXECUTIVE Users Manual.*

Throughout the TENET System, files are distinguished by content type (text or binary), record type (fixed- or variable-length), usage (as programs or data) and protection (private or shared). Although files created in any of the subsystems have certain general characteristics in common, there are special file functions which may be performed only in a particular subsystem. For example, file privacy and protection may be defined only at the EXECUTIVE level; only the language compilers TENET BASIC and FORTRAN IV can create and use files in binary format and use files as programs (i.e., to process information).

The files created and processed in the EDITOR subsystem have the following characteristics (regardless of how they are used outside the EDITOR workspace).

- All files must be text, i.e., character strings resembling teletypewriter input/output. Binary files may not be used in the EDITOR subsystem as they are written in internal machine language format and as such, cannot be printed at the teletypewriter terminal.
- Files may have fixed- or variable-length records but each record is limited to a maximum length of 256 characters.
- The EDITOR operates on files as data, independent of the ultimate usage of the file; namely, the EDITOR is unaware of whether the file is actually a FORTRAN IV program, a BASIC program, or data to be processed by FORTRAN IV or BASIC.

## 2. USING EDITOR

### THE TERMINAL

Before the user can access the EDITOR subsystem, he must first establish a connection with the TENET 210 computer. The computer receives and transmits information through a terminal such as the Model 33 Teletypewriter Terminal. The method of connecting the terminal to the computer varies according to the type of the terminal (direct-coupled or dial-in). Once the connection is established, the system prints out a header message such as:

```
TENET TIMESHARING 8/18/70
```

*The Model 33 Teletypewriter Terminal is described in Appendix C.*

*The content of the header message is variable and determined by individual installations.*

### Mode of Operation

The TENET 210 computer receives information from the terminal in full duplex mode only. Characters typed into the terminal are transmitted to the computer without simultaneous printing. Once the computer receives a character, it transmits a character back to the terminal for printing. This is a full duplex operation whereby the listing (terminal output) documents exactly what the computer receives, thus minimizing undetected transmission errors. Therefore, a character printed at the terminal is an echo of what the computer received. Certain control characters are not echoed. As a security precaution words denoting special permissions are never echoed.

### Special Keys

The Teletypewriter Terminal is used in much the same way as a standard typewriter. However, in addition there are several keys with special functions:

**CR**

The Carriage Return key signals an end of record. The teletypewriter print head is positioned at the beginning of the line. The system automatically supplies a line feed for every carriage return.

**LF**

The Line Feed key is used to enter records consisting of more than one line of information. Each time the Line Feed key is pressed, the paper is advanced one line. The system automatically supplies a carriage return for every line feed; this does not act as an end of record signal but as a continuation of the previous line. The last line of a multi-line record must be terminated by a carriage return supplied by the user.

**A<sup>c</sup>**

The A control key deletes the previous character entered by the user. It echoes a backspace arrow (←) and eliminates a character each time the key is pressed. For example:

TENT **A<sup>c</sup>** ET 222 **A<sup>c</sup>** **A<sup>c</sup>** 10

is printed at the terminal as:

TENT ← ET 222 ←← 10

The system will interpret it as:

TENET 210

**Q<sup>c</sup>**

The Q control key deletes the line of information currently being entered (i. e., before a **CR** is issued by the user). It echoes an upward arrow (↑), and a carriage return and line feed are automatically generated to position the teletypewriter print head to the beginning of a new line. The user can then enter a new or corrected line. If the **Q<sup>c</sup>** key is used while entering multi-line statements, only the text following the most recent Line Feed will be deleted.

**ESC** or  
**ALTMODE**

The Escape or ALTMODE key causes an interrupt of current operations. It causes all input/output operations at the terminal to be terminated; input not yet processed is lost. Since this key has a different significance in other situations (e. g., during ENTER mode), this key should not be used in place of the Q control key to delete the line currently being entered. Other uses of the **ESC** key are discussed where appropriate in this manual.

**EOT** or  
**D<sup>c</sup>**

The End of Text key terminates an input line and signals an end of record. However, unlike the Carriage Return or Line Feed keys, **EOT** does not cause the teletypewriter print head to be repositioned.

**BREAK** On dial-in terminals the Break key causes a transmission interrupt and disconnects the terminal from the computer. Pressing this key could cause a loss of the current program and data. On direct-coupled terminals this key is ignored.

**RUBOUT** This key is used to delete characters on paper tape; it is ignored but does not ring the bell.

**BELL** The bell sounds whenever unrecognizable characters are entered, and at line position 61 to warn the user that the 72-character teletypewriter line limit is approaching.

## LOGIN SEQUENCE

Once the terminal is connected to the computer and the header message is printed, the system is automatically placed in the LOGIN mode and the user is requested to identify himself.

-LOGIN account; name **CR**

where: account = user's account number (0-511); account number must be followed by a semicolon.

name = user's name (1-8 alphanumeric characters, % or \$, the first of which must be alphabetic, \$, or %).

If either of the items requested by the LOGIN prompt are omitted or invalid, the user is reprompted by ACCOUNT? or NAME?.

If the user has designated a password to be associated with his name, the LOGIN sequence will include a prompt for a password (after the user enters account number; name **CR** )

-PASSWORD?

As a security precaution password entries are not echoed at the terminal.

## Correcting LOGIN Entries

The control characters **A<sup>c</sup>** and **Q<sup>c</sup>** may be used during the LOGIN sequence to correct entries as they are typed in and before the Carriage Return key is pressed. However, the Escape or Alt Mode key in the LOGIN mode causes the system to reissue the LOGIN prompt, thereby causing the LOGIN sequence to begin again.

The system allows the user three attempts to identify himself. After three unsuccessful attempts or three minutes without a response from the user, the terminal is automatically disconnected.

## ACCESSING EDITOR

Once the user has successfully completed the LOGIN sequence, he is at the EXECUTIVE level. From the EXECUTIVE level he may access the EDITOR subsystem by issuing the EXECUTIVE command "EDITOR" or "EDIT" in response to the EXECUTIVE prompt character "-". When the EDITOR subsystem is in control, it issues its own prompt character "@".

```
-EDITOR   
@
```

## THE ELEMENTS OF EDITOR

### The EDITOR Workspace

The EDITOR workspace is a temporary storage area into which the user may enter records to create files or copy existing files from disc storage. The size of the workspace is variable and is determined by individual installations. Files must be edited in segments when the editing operations to be performed will cause workspace capacity to be exceeded.

*Typical workspace size is  
112,000 characters.*

### Entering Information to EDITOR

In response to the EDITOR prompt character @, the user may enter any EDITOR command or any record of information. Files are created in the EDITOR workspace by entering records of information preceded by a record number and terminated by a Carriage Return. Multi-line records may be created by pressing the Line Feed key between multiple lines.

### Character Set

The following characters are recognized by EDITOR:

- Alphabet: A B C D E F G H I J K L M N O P Q R S T U V W  
X Y Z
- Digits: 1 2 3 4 5 6 7 8 9 0

- Special characters:

'	single quote	(	left parenthesis
"	double quote	)	right parenthesis
<	less than	,	comma
=	equal to	.	period
>	greater than	;	semicolon
+	plus	:	colon
-	minus		blank
*	asterisk	@	at
/	right oblique	\$	dollar sign
↑	up arrow	%	per cent
?	question mark	\	left oblique
[	left bracket	#	number sign
]	right bracket	&	ampersand
!	exclamation mark	←	backward arrow

## Record Length

Records can be a maximum of 256 characters in length. However, due to the limitation of the teletypewriter carriage width (72 characters) a Line Feed must be used to continue records from line to line.

## Blanks

It is necessary to separate EDITOR reserved words, file names, numeric values, and strings with one or more blanks. In general, a blank is necessary after any item which could have the first character of the next item as a potentially valid character.

*Blanks may not be embedded in any record number, file name, EDITOR reserved word, or number.*

## Syntax Error Checking and Recovery

Immediately after an EDITOR command is entered from the terminal, it is checked for correct syntax by the EDITOR subsystem. If an error is found, the user is notified with an error message. The command in error is not deleted, but the user is automatically placed in extended editing mode. At this time he can correct the command using the special control keys described under the ALTER and MODIFY commands or simply reenter the correct command followed by a Carriage Return.

*The ALTER command is discussed on p.22; MODIFY is discussed on p.25.*

Syntax error checking in the EDITOR subsystem applies only to EDITOR commands. Records entered as input to the EDITOR are not checked for errors relevant to other subsystems. Thus it is possible, for example, to

enter a record which is not flagged as an error by EDITOR, but is syntactically unacceptable to the FORTRAN IV compiler.

EDITOR also produces messages resulting from execution errors and warnings to the user. These are fully described with recovery procedures in Appendix B of this manual.

*Appendix B begins on p.53.*

## **Glossary of EDITOR Terms**

The following special terms may be used to reference information in the EDITOR workspace.

### Record Number

When records are entered into the EDITOR workspace, whether from a disc file or the teletypewriter terminal, they must be assigned (record) numbers. Record numbers are of the form:

xxxxx.xx

They must consist of a maximum of five digits preceding and two digits following a decimal point. Zeros before the integer part and zeros after the fractional part may be omitted. The decimal point may be omitted if there is no fractional part.

Record numbers may be generated in the following ways:

- The user types a record preceded by its record number.
- The ENTER command is used to automatically generate record numbers which prompt the user for record content.
- EDITOR automatically prefaces record numbers to records loaded or copied into the workspace (unless the file to be loaded or copied already has record numbers).

Although record numbers actually take up 8 character positions (plus one space between record number and record content), the first character position of a record is considered to start with the first character within the record. Care should be exercised in any of the commands which use a character position constraint since even though record content actually begins at character position 10, EDITOR considers it to begin at character position 1.

*The user may change the standard format of record numbers by using the FORM command, p.36.*

*The ENTER command is described on p.15.*

*A record number is not part of a record; records begin with the first character following the blank space after the record number.*



### Explicit Range

Records within the EDITOR workspace may be directly identified by their record numbers. An explicit range specification consists of one or more range elements separated by commas. There are two types of range elements:

- record no.
- record no. <sub>1</sub>:record no. <sub>2</sub>

The first range element refers to a single record; the second specifies a sequence of records starting with record no. <sub>1</sub> and ending with record no. <sub>2</sub> (record no. <sub>2</sub> must be greater than record no. <sub>1</sub>). Both range elements may be combined in a single reference. For example, 20:50, 230, 100:120 may be used to reference records 20 through 50, 230, and 100 through 120.

In addition to specifying explicit range by record numbers, the following special identifiers may be used.

FIRST or F = first record  
LAST or L = last record  
END = last record + 1  
ALL = FIRST:LAST

*END indicates a record number at which new records may be appended to the workspace.*

### Associative Range

An associative range specification references by record content rather than by record number. By specifying a set of characters (string) enclosed in quotation marks (single or double), the user may define a set of records which are associated by the presence of the specified string. For example, the string "ABC" may be used to reference all records containing "ABC". Similarly, by preceding an associative range specification by the word NOT, the user may specify a set of records which do not have the string attributes specified. For example, NOT "ABC" may be used to reference all records which do not contain "ABC". An associative range specification may optionally contain several constraints and is of the form:

[NOT] 'string' [position] [(occurrence)] [IN explicit range]

where

position	specifies the character position(s) of the record in which the string must occur. Position must be expressed as a single integer value, or as a range of integer values integer <sub>1</sub> :integer <sub>2</sub> ). Only one position constraint is allowed in a range specification. For example: "AB" 10:15 specifies that AB
----------	---

must occur between positions 10 and 15 of the record; "AB" 10 means that AB must start at position 10 (i. e. , occur in positions 10 and 11). If no position constraint is specified, the string may occur anywhere in the record.

**occurrence** specifies the nth record found containing the string. Occurrence may be expressed as a list of one or more integer values or a range of integer values (integer:integer) or any combination of these. Occurrence must be enclosed in parentheses to distinguish it from character position specification. For example, "AB"(3) specifies only the third record found with the string AB; "AB"(1,6:8,12) specifies the first, sixth, seventh, eighth, and twelfth records found with AB. If no occurrence constraint is specified, all records satisfying the other constraints specified are implied.

**IN explicit range** specifies the records of the file to be searched for the string. Using this constraint greatly expedites the search process since it confines the search to a portion of the file only. For example, "AB" IN 10:30 causes only records 10 through 30 to be searched for AB. If no explicit range is designated, all records are searched.

**Example:**

*list all records at the terminal*

```

@ LIST ALL
Δ5ΔNEW YORKΔΔΔΔΔΔΔΔΔΔΔΔNEW YORKΔΔΔΔΔΔΔΔΔΔΔΔ7,781,984
10 CHICAGO ILLINOIS 3,550,404
15 LOS ANGELES CALIFORNIA 2,479,015
20 PHILADELPHIA PENNSYLVANIA 2,002,512
25 DETROIT MICHIGAN 1,670,144
30 BALTIMORE MARYLAND 939,024
35 HOUSTON TEXAS 938,219
40 CLEVELAND OHIO 876,050
45 WASHINGTON DISTRICT-COLUMBIA 763,956
50 ST. LOUIS MISSOURI 750,026
55 SAN FRANCISCO CALIFORNIA 742,855
60 MILWAUKEE WISCONSIN 741,324
65 BOSTON MASSACHUSETTS 697,197
  
```

70	DALLAS	TEXAS	679,684
75	NEW ORLEANS	LOUISIANA	627,525
80	PITTSBURGH	PENNSYLVANIA	604,332
85	SAN ANTONIO	TEXAS	587,718
90	SAN DIEGO	CALIFORNIA	573,224
95	SEATTLE	WASHINGTON	557,087
← @ LIST "CALIFORNIA"			
15	LOS ANGELES	CALIFORNIA	2,479,015
55	SAN FRANCISCO	CALIFORNIA	742,855
90	SAN DIEGO	CALIFORNIA	573,224
← @ LIST "2" 41			
15	LOS ANGELES	CALIFORNIA	2,479,015
20	PHILADELPHIA	PENNSYLVANIA	2,002,512
← @ LIST "TEXAS" 21 (1:2)			
35	HOUSTON	TEXAS	938,219
70	DALLAS	TEXAS	679,684
← @ LIST "TEXAS" 21 (1) IN 45:95			
70	DALLAS	TEXAS	679,684

*list only the records containing the string CALIFORNIA*

*list all records containing "2" in character position 41*

*list first two records containing TEXAS starting in character position 21*

*list the first record within records 45-95 with TEXAS starting in character position 21*



## 3. EDITOR COMMANDS

### INTRODUCTION

The EDITOR commands operate on the current contents of the EDITOR workspace and may be issued only in response to the EDITOR prompt character @ .

Where the range of lines is not specified, the command will affect the entire content of the workspace. Range may be expressed as explicit range or associative range.

Any of the EDITOR commands may be terminated by using the Escape or Alt Mode key.

*The result of pressing the Escape key depends on the command currently in effect (see Messages, p. 53).*

### Conventions

The following conventions are used throughout the discussion of the EDITOR command set:

- Each statement discussion contains the following elements:
  - 1) format representation
  - 2) syntactic example
  - 3) explanation of purpose
  - 4) rules of usage (bulleted items)
  - 5) usage example
- Uppercase letters, digits, and special characters must appear exactly as shown in the format representation for all statements.
- Information in lowercase letters in the format representations is to be supplied by the user.
- Braces { } indicate that one of the items enclosed must be selected.
- Brackets [ ] indicate that one of the items enclosed may be optionally selected.
- Control characters are indicated by the superscript <sup>c</sup>.

- Special keys such as Line Feed and Carriage Return are signified in the text and examples by their initials enclosed in a circle; e.g., **CR** , **LF** .
- Underlined text in the examples indicates computer output or computer requests.
- In examples where spacing is critical, blanks are indicated by the character  $\Delta$  .
- Examples following the discussion of each command are for the purpose of demonstrating the usage of the command, and are not necessarily examples of good programming practice.
- Although a Carriage Return must be used to terminate every record of information entered from the terminal, Carriage Returns are not shown in the examples of the EDITOR commands except where they have special significance.

## ENTER Command

`ENTER [record no.] [STEP increment]`

ENTER

ENTER 400

ENTER 400 STEP 0.5

Normally the user must type in a record number for each record when creating a file. However, the ENTER command can be used to direct the system to generate record numbers automatically as the user creates a program or inserts or adds records to an existing file. The ENTER command causes the system to prompt the user with the record number of the record to be entered. EDITOR continues to prompt with record numbers until the user either presses the Escape key or enters an End of Text (D<sup>c</sup>) in response to a record number prompt.

- The initial record number to be generated may be specified by the user.  
  
The default value for this option is 1.0 if there are no records currently in the workspace or the integer part of the last current record number +1.0 if there are records currently in the workspace when this command is issued.
- The STEP (increment) option may be used to specify a value by which successive record numbers are incremented. The default value for this option is 1.0.
- ENTER may be used to insert records within a file as long as any record number generated by the ENTER command does not exceed that of the record number before which the new records are inserted. Thus, ENTER may not be used to replace existing records by duplicating record numbers, or to interleave new records with old.

Example:

```
@ LIST 200:400
ΔΔ200.00ΔEQLIMO
 300.00 EQLIM1
 400.00 EQLOM1
@ ENTER 350
 350.00 EQLOMO
 351.00 [EOT]
```

*The record number specifications used in this command should be compatible with any FORM command currently in effect. The FORM command is described on p. 36.*

*A single record may be inserted between two existing records by typing the new record with a record number within the bounds of of the existing record numbers.*

```

@ LIST 200:400
200.00 EQLIMO
300.00 EQLIM1
350.00 EQLOMO
400.00 EQLOM1
@ ENTER 200.01 STEP .01
200.01 ΔΔΔΔΔΔ
200.02 ADDRESS EQUAL, INSTRUCTION FETCH, UNMAPPED
200.03 [EOT]
@ ENTER 200.10 STEP .01
200.10 *EQLIMO=
200.11 + */PSCLO.PSCHO.UNMAPI/
200.12 [EOT]
@ ENTER 200.50 STEP .01
200.50 LOADS:
200.51 *EQLINS-001,X02,03N,01
200.52 ADDEQL-102,X02,02P,01
200.53 [EOT]
@ ENTER 200.30 STEP .01
200.30 IMPLEMENTATION
200.31 101,X02,02R,12
200.32 SCHEMATIC-1000013,SH4,B4
200.33 [EOT]
@ LIST 200:400
200.00 EQLIMO
200.01
200.02 ADDRESS EQUAL, INSTRUCTION FETCH, UNMAPPED
200.10 *EQLIMO=
200.11 + */PSCLO.PSCHO.UNMAPI/
200.30 IMPLEMENTATION
200.31 101,X02,02R,12
200.32 SCHEMATIC-1000013,SH4,B4
200.50 LOADS:
200.51 *EQLINS-001,X02,03N,01
200.52 ADDEQL-102,X02,02P,01
300.00 EQLIM1
350.00 EQLOMO
400.00 EQLOM1
@

```



## LIST Command

LIST [range] [TEXT NUM[BERS]]
----------------------------------

LIST

LIST 16:90.6

LIST 'AB' IN 100:200 TEXT

LIST 'Y'30 IN ALL NUMBERS

The LIST command causes the records in the workspace specified by the range to be printed at the terminal with their associated record numbers.

- If range is not specified, all records are listed.
- The TEXT option in the LIST command causes the contents of each record to be printed without associated record numbers.
- The NUMBERS option causes only the record number of each record specified in the range to be printed at the terminal. This option enables the user to easily ascertain, for instance, all occurrences of a certain string within a particular file.
- If neither TEXT nor NUMBERS is specified, the content of each record is listed including both the record number and textual content.

Examples:

```
@ LIST "18" 20 IN 1:130
10.00 JONES 18423-C
24.09 MAYNARD 18238
45.45 MERRITT 18976-D
50.00 RYDER 18247
78.40 SMALL 18935-H
84.03 TINDLE 18263
101.00 TELLER 18200
129.76 DIXON 18003-P
@ LIST "18" 20 IN 1:130 NUM
10.00
24.09
45.45
50.00
78.40
84.03
101.00
129.76
@
```

## DELETE Command

DEL[ETE] [range] <span style="font-size: 2em; vertical-align: middle;">[</span> LIST TEXT NUM[BERS] <span style="font-size: 2em; vertical-align: middle;">]</span>
--

DELETE

DEL ALL

DELETE 100:200.5

DELETE 'ARE' IN ALL

The DELETE command deletes all records of a file within the range specified from the workspace. This command affects only the copy of a file maintained in the EDITOR workspace. Files may be deleted from a user's directory at the EXECUTIVE level only.

*For information about deleting files from the user's file directory, see the EXECUTIVE Users Manual.*

- If range is not specified, the entire file is deleted from the workspace.
- The LIST option causes the record number and content of each deleted record to be printed.
- The TEXT option causes only the content of each deleted record to be printed.
- The NUMBERS option causes only the record number of each deleted record to be printed.
- If neither LIST, TEXT, nor NUMBERS is specified, no part of the deleted records is printed.

Example:

```
@ LIST 20:50
△△△20.00△JONES, CASEY
  30.00 1234 NONESUCH PLACE
  40.00 HANNIBAL, MISSOURI
  50.00 23409
@ DELETE 30:40
@ LIST 20:50
  20.00 JONES, CASEY
  50.00 23409
@
```

## REPLACE Command

`REP[LACE] [range] [STEP increment]`

```
REP 10:30 STEP .1
REPLACE 'Y3' 30:33 IN ALL
REPLACE
```

The REPLACE command enables the user to replace records within the workspace. The operation performed by the REPLACE is equivalent to issuing a DELETE command followed by an ENTER command. The REPLACE command deletes the records specified and then prompts the user with record numbers for replacement records starting with the record number of the first deleted record. The user may then enter any number of records as long as no generated record number exceeds the record number of the record before which the new records are inserted. After each replacement record is entered, the user is prompted for the next record with the next sequential record number.

- If range is not specified, the entire content of the workspace is deleted.
- The STEP (increment) option may be used to specify a value by which successive record numbers are incremented. The default value for this option is 1.0.

*A single record may be replaced or changed by entering its record number followed by the new record content.*

### Examples:

```
@ LIST
10 TWAS THE NIGHT BEFORE CHRISTMAS
15 AND WE HAVE MET THE ENEMY
20 NOT EVEN A MOUSE
@ REP 12:15 STEP 3
12 AND ALL THROUGH THE HOUSE
15 NOT A CREATURE WAS STIRRING
18 EOT
@ LIST
10 TWAS THE NIGHT BEFORE CHRISTMAS
12 AND ALL THROUGH THE HOUSE
15 NOT A CREATURE WAS STIRRING
20 NOT EVEN A MOUSE
@
```

```
@ LIST 140.10:140.19
140.10
140.11 ADDEQL=
140.12 + EQLIMO
140.13 + EQLIMI
140.14 + EQLOMO
140.15 + EQLOMI
@ REP 140.12
140.12 + PSCLO.PSCHO.UNMAP I
@ LIST 140.10:140.19
140.10
140.11 ADDEQL=
140.12 + PSCLO.PSCHO.UNMAP I
140.13 + EQLIMI
140.14 + EQLOMO
140.15 + EQLOMI
@
```

## RENUMBER Command

```
REN[UMBER] [start value] [STEP increment]
```

```
RENUMBER  
RENUMBER 100  
REN STEP 10.0  
REN 500 STEP .5
```

The user may change the record numbers of the records within the workspace by using the RENUMBER command. RENUMBER causes the entire contents of the EDITOR workspace to be renumbered.

It is convenient to use this command after the content of the workspace has been substantially modified within the EDITOR subsystem or before a file with duplicate record numbers is appended to the file currently within the workspace.

- The start value option allows the user to specify the record number at which renumbering begins. The default value for this option is 1.0.
- The STEP (increment) option may be used to specify the value by which successive record numbers are incremented. The default value for this option is 1.0.

*The record number specifications used in this command should be compatible with any FORM command currently in effect. The FORM command is described on p. 36.*

Example:

```
@ LIST  
ΔΔΔ20.00Δ30ΔACCEPT I  
22.00 FACT = 1  
25.00 20 IF I<1 THEN GO TO 10  
30.00 FACT = FACT*I  
32.00 I = I-1  
33.00 GO TO 20  
35.00 10 DISPLAY FACT  
37.00 GO TO 30  
@ RENUMBER 100 STEP 100  
@ FORM 4  
@ LIST  
Δ100Δ30ΔACCEPT I  
200 FACT = 1  
300 20 IF I<1 THEN GO TO 10  
400 FACT = FACT*I  
500 I = I-1  
600 GO TO 20  
700 10 DISPLAY FACT  
800 GO TO 30
```

## ALTER Command

ALTER [range]

ALTER  
ALTER 10:300  
ALTER "RATE"

The ALTER command enables the user to change the content of any record or set of records by means of a set of special editing control characters. When the ALTER command is executed, the user is placed in extended editing mode. The first record specified by range is printed at the terminal (preceded by its record number). The user then may alter the content of the record by the characters described below. After the user completes work on a line, the next line in the range specification is printed at the terminal and editing continues until the last record is edited or until the user types the ESC key, thus terminating extended editing mode.

The following control characters may be used in conjunction with the ALTER command:

*These control characters may be used to recover from syntax errors in any EDITOR command. (See the discussion of syntax error checking and recovery, p. 7.)*

<u>Character</u>	<u>Function</u>
$\boxed{C^c}$	Copies the next character from the old record to the new record; the copied character is echoed at the terminal.
$\boxed{S^c}$	Skips the next character in the old record; a % is echoed at the terminal.
$\boxed{D^c}$	Copies the remainder of the old record to the new line and terminates editing the old record; all copied characters are echoed at the terminal.
$\boxed{E^c}$	Enter/Exit insert mode. Echoes '<' and '>', respectively. The text typed in the insert mode is entered into the new record and does not affect the old record.

Character

Function

A<sup>c</sup>

When the backspace key is used in extended editing mode, it affects only the content of the new record.

For example:

if the old record is: A B C D E F G

and the user types: 

C <sup>c</sup>	C <sup>c</sup>	A <sup>c</sup>	X	Y	D <sup>c</sup>
----------------	----------------	----------------	---	---	----------------

the system will echo: A B ← X Y E F G

the new record is: A X Y E F G

Q<sup>c</sup>

When the key for deleting records being entered is used in extended editing mode, only the contents of the new record are affected. For example:

if the old record is: V E N I V I D I

and the user types: 

C <sup>c</sup>	C <sup>c</sup>	C <sup>c</sup>	C <sup>c</sup>	Q <sup>c</sup>	D <sup>c</sup>
----------------	----------------	----------------	----------------	----------------	----------------

the system will echo: V E N I ↑

V I D I

the new record is: V I D I

Other

Other characters entered from the terminal replace corresponding characters in the old record.

- If a range or set of statements is specified in the ALTER command, statements are printed one at a time and the next sequential statement is printed after the previous statement is edited.
- Multi-line statements are edited on a segment basis, i.e., between lines feeds to a terminating carriage return. The control character 

D <sup>c</sup>
----------------

 will copy only up to the next line feed or carriage return character.
- Although it is possible to type in a new record number when editing an old record, the system retains the old number and any subsequent usage of listing of the record will cause the original record number to be used.

Example:

@ ALTER 20

Δ Δ 2 0 Δ L S T Δ A + L B \* C

C <sup>c</sup>	C <sup>c</sup>	C <sup>c</sup>	C <sup>c</sup>	C <sup>c</sup>	C <sup>c</sup>
----------------	----------------	----------------	----------------	----------------	----------------

 E 

C <sup>c</sup>	C <sup>c</sup>	C <sup>c</sup>
----------------	----------------	----------------

 = 

S <sup>c</sup>	EOT
----------------	-----

 (as typed)

Δ Δ 2 0 Δ L E T Δ A = % B \* C (as echoed)

@ ALTER 20

Δ Δ 2 0 Δ L E T Δ A = B \* C

C	C	C	C	C	S	S	S	S	C	E
---	---	---	---	---	---	---	---	---	---	---

 (I,J) 

E	C	C	C
---	---	---	---

 + D/F (as typed)

Δ Δ 2 0 Δ % % % % A < (I,J) > = B + D/F (as echoed)

@ LIST

ΔΔ20ΔA(I,J)=B+D/F



## MODIFY Command

MOD[IFY] [range]

MOD  
MODIFY  
MODIFY 10:300  
MOD ALL

The MODIFY command is identical to the ALTER command except that only the record numbers of the records to be edited are printed at the terminal.

The following special control characters may be used with the MODIFY command.

<u>Character</u>	<u>Function</u>
<code>C<sup>c</sup></code>	Copies the next character from the old record to the new record; the copied character is echoed at the terminal.
<code>S<sup>c</sup></code>	Skips the next character in the old record; a % is echoed at the terminal.
<code>D<sup>c</sup></code>	Copies the remainder of the old record to the new record and terminates editing the old record; all copied characters are echoed at the terminal.
<code>E<sup>c</sup></code>	Enter/Exit insert mode echoes a '<' and '>', respectively. The text typed between the first <code>E<sup>c</sup></code> and the next <code>E<sup>c</sup></code> is inserted into the new record and does not affect the old record.
<code>A<sup>c</sup></code>	When the backspace key is used in extended editing mode, it affects only the content of the new record. For example: if the old record is: A B C D E F G and the user types: <code>C<sup>c</sup>C<sup>c</sup>A<sup>c</sup></code> X Y <code>D<sup>c</sup></code> the system will echo: A B ← X Y E F G the new record is: A X Y E F G

*These control characters may be used to recover from syntax errors in any EDITOR command. (See the discussion of syntax error checking and recovery, p. 7).*

Character

Function

**Q<sup>c</sup>**

When the key for deleting records being entered is used in extending editing mode, only the contents of the new record are affected. For example:

if the old record is: V E N I V I D I

and the user types: 

C <sup>c</sup>	C <sup>c</sup>	C <sup>c</sup>	C <sup>c</sup>	Q <sup>c</sup>	D <sup>c</sup>
----------------	----------------	----------------	----------------	----------------	----------------

the system will echo: V E N I ↑  
V I D I

the new record is: V I D I

**Other**

Other characters entered from the terminal replace corresponding characters in the old record.

- If a range or set of statements is specified in the MODIFY command, statements are printed one at a time and the next sequential statement is printed after the previous statement is edited.
- Multi-line statements are edited on a segment basis, i.e., between lines feeds to a terminating carriage return. The control character **D<sup>c</sup>** will copy only up to the next line feed or carriage return character.
- Although it is possible to type in a new record number when editing an old record, the system retains the old number and any subsequent usage of listing of the record will cause the original number to be used.

**Example:**

@ LIST 40

ΔΔ40ΔSMITH,WM.JR

@ MODIFY 40

ΔΔ40Δ

C <sup>c</sup>	C <sup>c</sup>	C <sup>c</sup>	C <sup>c</sup>	C <sup>c</sup>	C <sup>c</sup>	C <sup>c</sup>	C <sup>c</sup>	C <sup>c</sup>
----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------

 Δ 

S <sup>c</sup>	C <sup>c</sup>
----------------	----------------

 (as typed)

ΔΔ40Δ S M I T H , W M Δ S R (as echoed)

@ LIST 40

ΔΔ40ΔSMITH,WMSR

## CHANGE Command

CH[ANGE]	[string <sub>1</sub> [position][(occurrence)]] position	[TO'string <sub>2</sub> '] [IN range]	[LIST TEXT NUM[BERS]]
----------	--	---------------------------------------	-----------------------------

```
CHANGE 40:45 TO '    '  
CH 'X' 30 TO '$' IN 400:450 TEXT  
CHANGE 'BETA' (2) TO 'ALPHA' NUMBERS
```

The CHANGE command replaces one set of characters with another throughout the range specified by the user. There are several options available with this command which permit the user a considerable variety of intraline editing facilities.

- 'string<sub>1</sub>' is the string of characters to be replaced; 'string<sub>2</sub>' is the replacement string.
- 'string<sub>1</sub>' specification may be qualified by specifying the character position(s) in which the string may occur. Character position within a line may be expressed as a single integer value or as a pair of integer values (integer:integer). If the character position constraint consists of a single integer, the string being replaced (string<sub>1</sub>) must begin in the character position specified. If the character position constraint consists of a pair of integer values, a search is made for string<sub>1</sub> contained in any of the positions specified and the replacement string<sub>2</sub> is inserted in string<sub>1</sub>'s place. If string<sub>2</sub> is smaller than string<sub>1</sub>, the length of the record is shortened accordingly. If string<sub>2</sub> is larger than string<sub>1</sub>, it is inserted and the remainder of the record is moved to the right to accommodate string<sub>2</sub> (the length of the record is increased accordingly).
- String<sub>1</sub> may also be qualified by an integer value (occurrence) specifying the nth occurrence of the string within a record. For example, an occurrence value of two causes the CHANGE command to affect only the second string found in each record.
- If string<sub>1</sub> is not specified, the character position constraint must be, and the character positions specified will be changed regardless of their original content. If only a single position is specified, the replacement string will be inserted in front of the specified position.

*Occurrence specification in the CHANGE command is different from occurrence in a range specification. In the latter, occurrence specifies the nth record found containing the string.*

- The CHANGE command may be used to delete character strings in a record, by specifying string<sub>2</sub> as a null record (' ' or " " with no blanks between quotes). If string<sub>2</sub> is specified as a string of blanks ('ΔΔ...ΔΔ '), an equivalent string of blanks will appear in the changed record.
- If IN range is not specified, the entire workspace is assumed.
- If the LIST option is used, the record and record number of each changed record is printed after each record is changed.
- If the TEXT option is used, only the content of the record (i. e., without record numbers) is printed after each record is changed.
- If the NUMBERS option is used, only the record number of each statement affected by the CHANGE command is printed after each record is changed.

Example:

```

@ 1 THAT IS TEXT DATA
@ 2 THAT IS ALSO TEXT DATA
@ CHANGE "THAT" TO "THIS" IN ALL TEXT
THIS IS TEXT DATA
THIS IS ALSO TEXT DATA
@ CHANGE "IS" (2) TO "WAS" IN ALL LIST
ΔΔΔΔ1.00ΔTHIS WAS TEXT DATA
2.00 THIS WAS ALSO TEXT DATA
@ CHANGE 1 TO "*" IN ALL LIST
1.00Δ*THIS WAS TEXT DATA
2.00 *THIS WAS ALSO TEXT DATA
@

```

## MOVE Command

MOVE range [TO record no.] [STEP increment]

### MOVE

MOVE 400:600 TO 900 STEP 1.0

MOVE TO 400

The MOVE command moves and renumbers all or a range of records within the workspace to a new location within the workspace. (The original records are deleted.)

*To copy records and not delete the originals, use the DUPLICATE command, p.30.*

- If a new location (TO record no.) is not specified, the records will be moved to the end of the workspace.
- If increment value is not specified, a value of 1.0 is assumed.
- MOVE may not be used to replace existing records (by duplicating record numbers).

### Example:

```
@ LIST 100:200,300:340
100.00  A = B + 3
120.00  GO TO 89
140.00  IF (A=B) THEN GO TO 103
160.00  A3 = MIN(A,B)
300.00  84 A2 = 6.4D-3
320.00  B = A17+A24
340.00  A = A24/B17
@ MOVE 300:320 TO 125 STEP 5
**** 2 RECORDS AFFECTED

@ LIST 100:200,300:340
100.00  A = B + 3
120.00  GO TO 89
125.00  84 A2 = 6.4D-3
130.00  B = A17+A24
140.00  IF (A=B) THEN GO TO 103
160.00  A3 = MIN(A,B)
340.00  A = A24/B17
@
```

## DUPLICATE Command

```
DUP[LICATE] [range] [AT record no.] [STEP increment] [FROM 'filename']
```

### DUPLICATE

```
DUP "X1" 20:22 IN ALL AT 300 BY 0.1 FROM 'MYFILE'  
DUPLICATE AT 400 FROM '98;AUER;AFILE'
```

The DUPLICATE command duplicates and renumbers a set of records or an entire file at another location within the workspace. The original records are not deleted by this operation.

*To copy records and delete the originals, use the MOVE command, p. 29.*

- If range specification is omitted, the entire current content of the EDITOR workspace (or 'filename' if specified) will be duplicated.
- If the AT option is not used, the duplicated records will be added to the end of the current content of the workspace (numbers will begin at LAST+1.0).
- If no filename is specified, the records in the workspace are used.
- The STEP option allows the user to specify a value by which successive record numbers are incremented. The default value for this option is 1.0.
- The FROM 'filename' option allows the user to duplicate all or part of a saved file into the EDITOR workspace. If 'filename' is owned by the user, the name consists of a string of 1 to 8 alphanumeric characters, including the symbols \$ and % enclosed in quotation marks. The first character must be alphabetic, % or \$. If 'filename' is not owned by the user but shared, the name must be of the form:

filename = 'a;un;fn'

where

a = account number (0-511) of the file's creator

un = name of the file's creator (1-8 alphanumeric characters including % and \$; the first character must be alphabetic, % or \$).

*Shared files are described in the EXECUTIVE Users Manual.*

fn = name of the file (1-8 alphanumeric characters including \$ and %; the first character must be alphabetic, % or \$).

- If during a DUPLICATE operation the EDITOR workspace capacity is filled, the system prints a warning message and the record number of the last record copied:

```
WORKSPACE FULL
xxxxx.xx LAST RECORD AFFECTED
```

Example:

```
@ 1 THREE BLIND MICE
@ 2 SEE HOW THEY RUN
@ 3 THEY ALL JUMPED OVER THE FARMER'S WIFE
@ DUPLICATE 2 AT 2 STEP .5
@ DUPLICATE 1 AT 1 STEP .5
@ LIST ALL
1.00 THREE BLIND MICE
1.50 THREE BLIND MICE
2.00 SEE HOW THEY RUN
2.50 SEE HOW THEY RUN
3.00 THEY ALL JUMPED OVER THE FARMER'S WIFE
@
```

## LOAD Command

```
LOAD [range] FROM 'filename' [BASIC] [CLEAR]
```

```
LOAD 15:30 FROM 'AFILE' CLEAR  
LOAD 'AB' 10:20 (20:100) FROM 'MYFILE'  
LOAD ALL FROM '$$FILE$' BASIC CLEAR
```

*Files loaded into the EDITOR workspace are only copies. The original files are always maintained on the disc.*

The **LOAD** command retrieves all or a portion of a previously saved file from disc storage and loads it into the EDITOR workspace. The **LOAD** command may be used to access any text file created by the user in EXECUTIVE, TENET BASIC, FORTRAN IV or EDITOR (and any shared files if the appropriate passkeys were entered at the EXECUTIVE level).

- If 'filename' is owned by the user, the name consists of a string of 1 to 8 alphanumeric characters, including the symbols \$ and %, enclosed in quotation marks. The first character must be alphabetic, %, or \$.
- If 'filename' is not owned by the user but shared, the name must be of the form:

filename = 'a;un;fn'

where

a = account number (0-511) of the file's creator

un = name of the file's creator (1-8 alphanumeric characters including % and \$; the first character must be alphabetic, %, or \$)

fn = name of the file (1-8 alphanumeric characters including \$ and %; the first character must be alphabetic, %, or \$)

- If the **BASIC** option in the **LOAD** statement is used, **BASIC** statement numbers are removed from the text of each record and used by EDITOR as record numbers. If this option is not used when loading a **BASIC** program file, **BASIC** program statement numbers will remain as part of the content of the file. Thus, the results of an editing operation which references file content with column constraints will vary depending on whether or not the **BASIC** option was used when a **BASIC** program file was loaded.



- The CLEAR option causes the current contents of the EDITOR workspace to be deleted before the file specified in the LOAD command is loaded. If this option is not used and the workspace does currently contain information, an error message is issued and the LOAD command is not executed.
- If during a LOAD operation the EDITOR workspace is filled to capacity, the system prints a warning message and the record number of the last record loaded into the workspace:

```

WORKSPACE FULL
xxxxx.xx LAST RECORD AFFECTED

```

Example:

```

@ LOAD "X1" IN ALL FROM 'BASICFIL' CLEAR
@ LIST ALL
402.10 20 REAL X1
420.90 90 LET X1 = X1/A*BETA
421.00 95 PRINT X1
@

```

## SAVE Command

SAVE [range]	<table border="1"><tr><td>NEW 'filename'</td></tr><tr><td>OLD 'filename'</td></tr><tr><td>WITH 'filename'</td></tr></table>	NEW 'filename'	OLD 'filename'	WITH 'filename'	[BASIC] [REN[UMBER]] [(no.)]
NEW 'filename'					
OLD 'filename'					
WITH 'filename'					

SAVE NEW 'MARFILE' RENUMBER (60)

SAVE 400:600 WITH 'FILE12'

SAVE OLD 'MYFILE'

The SAVE command enables the user to save part or all of the contents of the EDITOR workspace. The file is stored on the disc and maintained as part of the user's file directory.

- If range is not specified, the entire workspace will be saved.
- If 'filename' is owned by the user, the name consists of a string of 1 to 8 alphanumeric characters including the symbols % and \$, enclosed in quotation marks. The first character must be alphabetic, %, or \$.
- If 'filename' is not owned by the user but shared, the name must be of the form:

filename = 'a;un;fn'

where

a = account number (0-511) of the file's creator

un = name of the file's creator (1-8 alphanumeric characters, including % and \$; the first character must be alphabetic, %, or \$).

fn = name of the file (1-8 alphanumeric characters, including % and \$; the first character must be alphabetic, % or \$).

- NEW indicates that the file does not exist in the user's file directory.
- OLD causes the file to be saved as a replacement for a file of the same name already in the user's file directory. The file being replaced must be compatible with the file being saved, i. e., text with fixed- or variable-length records.

- WITH 'filename' saves the file at the end of an existent (previously saved) file.
- RENUMBER (if used without WITH) causes the file to be renumbered starting at 1.0 in increments of 1.0 as the specified records are stored on the disc file. If the WITH option is used with RENUMBER, the record numbers of the records being saved are changed so that the first record is assigned a number equivalent to the last record number +1.0 of 'filename'. Subsequent records are assigned numbers in increments of +1.0.
- The user may specify that the records are to be saved as fixed-length records by specifying record length (no.). Record length may be an integer value of from 1 to 256 characters. This option will always cause renumbering when saving. If used with the WITH option, this number must be the same as the record length specified for the old file.
- The BASIC option causes the record numbers used in the EDITOR workspace to be stored with the content of each record (i. e., as BASIC statement numbers). Thus, the file may be used by the BASIC subsystem as a program. However, the existing record numbers must be integer values (no fractional part) or an error message will be issued when the SAVE command is executed.

## FORM Command

FORM integer<sub>1</sub> [. integer<sub>2</sub>]

FORM 5

FORM 5.1

FORM 0.2

When record numbers in the EDITOR workspace are printed, they normally have the following format:

xxxxx. xx

On output the EDITOR automatically reserves a total of nine character positions (eight for the record number and one blank space) before the start of the text of each record. Thus, the text always starts at the tenth character position from the left side of a teletypewriter line. The user may change the standard amount of space allowed for record numbers (and implicitly, the magnitude range of record numbers) by using the FORM command. The FORM command may be used at any time and subsequent record number format specifications override any previous versions.

- integer<sub>1</sub> is the integer part of the record number and must be an integer value from 0 to 5 indicating the number of places preceding the decimal point (if specified).
- integer<sub>2</sub> is the fractional part of the record number and must be an integer value from 0 to 2 indicating the number of places following the decimal point. A specification of 0 for the fractional part is equivalent to no fractional specification, e.g., FORM 4.0 is equivalent to FORM 4 and no decimal point will appear in the record number.
- Record numbers are situated in the format specified as follows:
  - the integer part is right-justified with leading blank fill
  - the fractional part is left-justified with trailing blank fill
- Records may be generated in the workspace whose record numbers do not fit the current form. These records and their record numbers are entered into the workspace and may be addressed in a normal manner. However, when these records are printed, an "\*" will re-

place the blank that normally follows record numbers to indicate truncation (on the left or right end of the record number).

- Any command that creates or prints record numbers that do not fit the current form will cause the message

xxxxx.xx FORM ERROR

to be printed once; xxxxx.xx is the record number of the first record encountered whose record number did not fit the form.

Example:

```
@ ENTER 100 STEP 100
ΔΔ100.00ΔΔΔΔΔCOMMON A,B
 200.00 ACCEPT I,J
 300.00 CALL POLY(I,J)
 400.00 DISPLAY A,B
 500.00 EOT
@ FORM 2
@ LIST
```

```
***** 100.00 FORM ERROR
```

```
00*ΔΔΔΔCOMMON A,B
00* ACCEPT I, ESQ
@ FORM 3
@ LIST
100ΔΔΔΔΔCOMMON A,B
200 ACCEPT I,J
300 CALL POLY(I,J)
400 DISPLAY A,B
@
```

## LENGTH Command

LEN[GTH] integer

LEN 40

LENGTH 200

The LENGTH command allows the user to keep all records entered into the workspace within a specific length by causing a message to be printed whenever the user enters or loads a record exceeding the length specified in the LENGTH command. The message serves as a warning only; the record is not deleted.

- Record length must be an integer value from 1 to 256.
- Since the teletypewriter carriage width is limited to 72 characters per line, a line feed must be used to write lines greater than 72 characters even when the LENGTH command is in effect.
- Although lines exceeding the record length specified in the LENGTH command are not truncated in the EDITOR workspace, they may be truncated when the user saves the file using the fixed-length record option (i. e., if the actual record length exceeds the length specified in the SAVE command). The user is warned by the message:

LENGTH ERROR

Example:

```
@ LENGTH 10
@ ENTER
 $\Delta\Delta\Delta 1.00 \Delta A = B$ 
 $\underline{\quad 2.00 \quad} C = D$ 
 $\underline{\quad 3.00 \quad} E = A*\text{SQRT}(C)+3.14159$ 

****  $\underline{\quad 3.00 \quad}$  LENGTH ERROR

 $\underline{\quad 4.00 \quad}$  PRINT E
 $\underline{\quad 5.00 \quad}$  [EOT]
@ LIST
 $\underline{\quad 1.00 \quad} A = B$ 
 $\underline{\quad 2.00 \quad} C = D$ 
 $\underline{\quad 3.00 \quad} E = A*\text{SQRT}(C)+3.14159$ 
 $\underline{\quad 4.00 \quad}$  PRINT E
@
```

## TABS Command

TABS integer<sub>1</sub> [, integer<sub>2</sub> [, integer<sub>3</sub> [,integer<sub>4</sub>]]]

TABS 0

TABS 20, 40, 60

TABS 20, 15, 60, 50

The TABS command establishes tab settings on the teletypewriter in a way similar to tab key operation on a standard typewriter. The user may specify up to four tab stops across the width of the teletypewriter print line. The print head is moved from any position of the line to the next tab stop by pressing the  $I^C$  control key.

- Tab stop specification must be any integer value from 1 to 256.
- Tab stops may be specified in any order.
- Tab positions not specified are cleared.
- The command TABS 0 clears all previous tabs.

Example:

```
@ TABS 20
@ FORM 2
@ ENTER 1
1 JOHN DOE  $I^C$  408-245-8751
2 CARLTON PERKINS  $I^C$  415-323-2159
3 HARRY JUST  $I^C$  216-324-6987
4 SAM WONDERFUL  $I^C$  415-328-3246
5  $EOT$ 
@ LIST ALL
 $\Delta$ 1 $\Delta$ JOHN DOE $\Delta\Delta\Delta\Delta\Delta\Delta\Delta\Delta\Delta\Delta$ 408-245-8751
2 CARLTON PERKINS 415-323-2159
3 HARRY JUST 206-324-6987
4 SAM WONDERFUL 415-328-3246
```

## QUIT Command

Q[UIT] [CLEAR]

QUIT

Q

QUIT CLEAR

QUIT is the only command which enables the user to leave the EDITOR subsystem and return to EXECUTIVE control. When the user returns to EXECUTIVE, the current contents of the workspace are lost (unless the SAVE command is used prior to QUIT). To prevent the user from accidentally losing a file the EDITOR requires that the workspace is empty before the QUIT command can be executed. If the QUIT command is used when the workspace is not empty, EDITOR will print the message

WORKSPACE NOT EMPTY

If the CLEAR option is used in the QUIT statement, the current content of the EDITOR workspace will be cleared before control is returned to EXECUTIVE.

Example:

```
@ QUIT
**** WORKSPACE NOT EMPTY

@ SAVE WITH 'EDIT.04'
@ QUIT CLEAR
=
```



## Leaving the System - LOGOUT

After the user completes his work at the terminal, he may leave the system by issuing the EXECUTIVE command LOGOUT. This command causes the following accounting information to be printed at the terminal:

```
ttt                mm/dd/yy
CPU MINS - xx.xx
TERMINAL MINS - xx.xx
FILE MODULES - zzzz
```

where:

ttt	time in 24-hour clock units
mm/dd/yy	month/day/year
xx.xx	minutes
zzzz	number of file modules associated with account number and user name

After this message the system returns to the LOGIN mode and prompts with "-LOGIN". If there is no response from the user within three minutes or if the user turns the LINE/OFF/LOCAL knob to the OFF position, the terminal is disconnected.



## 4. SAMPLE TERMINAL SESSIONS

### EXAMPLE I: Updating and Interrogating a Text File

The EDITOR is a powerful tool for maintaining and updating text files. This example illustrates using EDITOR to update and interrogate a file containing a list of parts and information about the assemblies in which they are used. Each time a part is changed or obsoleted, this file is changed to reflect the current status of the part as well as its repercussions on other items of information in the file, e. g., cost per unit and quantity per assembly. The file used in this sample terminal session is organized as follows:

Records 1 - 2	describe the format of successive records in the file	
Remaining records	(with integer record numbers):	
	Character positions 1 - 12	part number
	13 - 22	part description
	remainder	not used (blank)
	(with non-integer record numbers):	
	Character positions 1 - 22	not used (blank)
	23 - 36	assembly on which part is used
	37 - 47	assembly description
	48 - 54	number of (featured) parts used in each assembly
	55 - 62	assembly revision level
	63 - 66	unit cost per assembly

```

- EDITOR ← Access the EDITOR subsystem
@ LIST 1,2 ← List the first two records containing format information
△△△△1.00 △PART△△△△△△△PART△△△△△△△USED ON△△△△△△△ASSEMBLY△△△△# PTS/△ASSEMB△△UNIT
  2.00 NUMBER   DESCRIPT  ASSEMB#   DESCRIPT  ASSEMB REV LEV COST
@ LIST "1002020000" 1:10 NUM ← Ascertain the record number
  343.00                                     of the record containing
@ LIST 343.00:343.99 ← By listing all records between record number
  343.00 1002020000 ROD                       343 and the next integer record number, a
  343.10                2003456000  TC GUAGE   1      G2      3.312
  
```

```

343.40          5003423100  RTR ASSY  3    L4    .820
-----
343.50          3002130000  T/M ASSY  2    C4    1.181
-----

```

@ DELETE 343.40 ← Delete reference to assembly 5003423100

@ LIST LAST NUM ← Ascertain last record in file

4005.40

@ ENTER 4006.00 STEP .10 ← Have EDITOR generate record numbers at the end of the file

4006.00 1002021000 ROD ← Update file by adding new records

```

4006.10          5003423100  RTR ASSY  2    L5    .950
-----

```

4006.20 EOT ← Terminate ENTER mode

@ LIST NOT '^' 1 IN 3:10 ← List all records in the range 3:10 that are not blank in character position 1; this produces a list of parts only

```

 3.00 100781000  SPRING CP
-----
 4.00 100802000  SPRING AD
-----
 5.00 100987200  MOTOR FR
-----
 7.00 100763800  SPRING
-----
 8.00 100234000  PIN
-----
 9.00 100235000  ROD BER
-----

```

@ LIST '2003456000' 23 IN ALL NUMBERS ← List all record numbers of records containing assembly number 2003456000. By truncating these record numbers to their integer parts, the user obtains a list of record numbers for each part used in this assembly.

```

98.30
107.20
224.50
343.10
394.30
402.30
481.70
497.40
503.20
706.80
842.80
875.10

```

@ LIST 98,107,224,343,394,402,481,497,503,706,842,875 TEXT

List the records containing  
the parts used in assembly  
2003456000

1002349000 LOCK

1002349800 WASHER

1003498400 SCREW

1002020000 ROD

1002934890 PLATE

1002347000 RTAINER

1029388000 HOOK

1002938400 HOUSING

1002938700 CONNCT

1029384000 LMT SW

1002936400 GLASS

1002930000 TRANST

@ QUIT CLEAR

Leave the EDITOR

-

## EXAMPLE 2: Preparation of a FORTRAN IV Program File

EDITOR is frequently used to prepare files which will be subsequently used as program files by the TENET BASIC and FORTRAN IV subsystems. This sample terminal session demonstrates the creation and editing of a (program) file to be used by the FORTRAN IV subsystem.

```

- EDITOR ← Access the EDITOR subsystem
@ FORM 4 ← Specify record number format to
@ TABS 5,10 ← Set tab stops at character positions
@ ENTER 10 STEP 10 ← Have EDITOR generate
  record numbers
ΔΔ10Δ I C ΔΔΔΔ INTEGER TT1,TT2,TT3
  20 I C ACCEPT DAT1,DAT2
  30 I C IF (DAT1.EQ.DAT2)GO TO PT1
  40 I C OPEN (6,'TTLIST')
  50 I C READ (6'16)TT1,TT2,TT3
  60 I C FACT=FUNCTION (TT1,TT2,TT3)*DAT2
  70 I C DISPLAY FACT
  80 I C GO TO PT2
  90Δ PT1ΔOPEN (7,'RRLIST')
  100 I C READ (7'16)TT1,TT2,TT3
  110 I C FACT=FUNCTION (TT1,TT2,TT3)
  120 I C DISPLAY FACT
  130Δ PT2ΔSTOP XXXX
  140 I C END
  150 I C ΔΔΔΔ I C ΔΔΔΔΔ INTEGER FUNCTION (ALPH,BETA,ROND)
  160 I C I C FUNCTION=(ALPH*BETA)**3/ROND
  170 I C I C RETURN
  180 EOT ← Terminate the ENTER command
@ CHANGE 'TT1,TT2,TT3' TO 'RR1,RR2,RR3' IN 100:110

**** 4 RECORDS AFFECTED

@ CHANGE 'FUNCTION' TO 'ROBB' ← Change the contents of
  various records throughout
  the workspace

**** 4 RECORDS AFFECTED

```

Enter FORTRAN IV program statements. Several dummy values are used here, such as PT1 and PT2 which will be changed when the program is completed. Note: when control character I is used, it is not considered a text character and as such does not take up a character position in the line being entered.

@ CHANGE 'ROBB' TO 'FUNCTION ROBB' IN 150

@ CHANGE 'PT1' TO '100'

\*\*\*\* 2RECORDS AFFECTED

@ CHANGE 'PT2' TO '200'

\*\*\*\* 2RECORDS AFFECTED

@ 15 INTEGER RR1,RR2,RR3 ← *Add a new record*

@ MOVE 10:140 ← *Move the program's function definition to the beginning of the program*

@ LIST

△150△△△△△△△△△△INTEGER FUNCTION ROBB (ALPH,BETA,ROND)

160 ROBB=(ALPH\*BETA)\*\*3/ROND

170 RETURN

171△△△△△INTEGER TT1,TT2,TT3

172 △INTEGER RR1,RR2,RR3

173△△△△△ACCEPT DAT1,DAT2

174 IF (DAT1.EQ.DAT2)GO TO 100 ← *Since the renumbering option was not used when the MOVE command was issued, record numbers are incremented by 1.0 ( 1 since the FORM command is in effect).*

175 OPEN (6,'TTLIST')

176 READ (6'16)TT1,TT2,TT3

177 FACT=ROBB (TT1,TT2,TT3)\*DAT2

178 DISPLAY FACT

179 GO TO 200

180 100△OPEN (7,'RRLIST')

181 READ (7'16)RR1,RR2,RR3

182 FACT=ROBB (RR1,RR2,RR3)\*DAT1

183 DISPLAY FACT

184 200 STOP XXXX

185 END

*Edit 172 to make it line up with the rest of the file (I<sup>c</sup> was not used when this record was created)*

@ CHANGE 1 TO "△△△△" IN 172 ←

@ CHANGE 'XXXX' TO '5050' IN 184

@ SAVE NEW 'FORTFIL' RENUMBER ← *Renumber and save the file for use by FORTRAN IV*

@ Q CLEAR ← *Clear the workspace and leave the EDITOR subsystem*

-





@ LIST 'INTRODUCT' 1:45 IN 11:LAST TEXT ← Search for titles with the word INTRODUCT

ACCOUNTING THEORY AND PRACTICE, INTRODUCTORY MILROY, R.G. 60HTMFL

@ LIST LAST NUMBERS ← Determine the record number of the last record in the workspace  
234

@ 235 BASIC ACCOUNTING - PART 1 PACE, H. 59P&P

@ 236 BASIC ACCOUNTING - PART 2 PACE, H. 59P&P ← Add new records to the end of the workspace

@ 237 COLLEGE ACCOUNTING SHERWOOD, J.R. 57SW

@ CHANGE 'BEGINNING' IN 1:10 TO 'BASIC, COLLEGE' TEXT ← Update keyword list to delete BEGINNING and add BASIC and COLLEGE  
BASIC, COLLEGE, ELEMENTARY

@ SAVE OLD 'ACCOUNT' ← Save the file ACCOUNT as updated in EDITOR

@ QUIT CLEAR ← Leave EDITOR

= ← EXECUTIVE is in control

## APPENDIX A. ANSI CHARACTER SET

ANSI Hex Code	Character	Teletype Key	Teletype/Printer Graphic	Hollerith Card Code	ANSI Hex Code	Character	Teletype Key	Teletype/Printer Graphic	Hollerith Card Code
00	NUL	P <sup>CS</sup>		12-0-9-8-1	25	%	5 <sup>S</sup>	%	0-8-4
01	SOH or DEL	A <sup>C</sup>		12-9-1	26	&	6 <sup>S</sup>	&	12
02	STX	B <sup>C</sup>		12-9-2	27	'	7 <sup>S</sup>	'	8-5
03	ETX	C <sup>C</sup>		12-9-3	28	(	8 <sup>S</sup>	(	12-8-5
04	EOT	D <sup>C</sup>		9-7	29	)	9 <sup>S</sup>	)	11-8-5
05	ENQ	E <sup>C</sup>		0-9-8-5	2A	*	: <sup>S</sup>	*	11-8-4
06	ACK	F <sup>C</sup>		0-9-8-6	2B	+	;: <sup>S</sup>	+	12-8-6
07	BEL	G <sup>C</sup>		0-9-8-7	2C	,	,	,	0-8-3
08	BS	H <sup>C</sup>		11-9-6	2D	-	-	-	11
09	HT	I <sup>C</sup>		12-9-5	2E	.	.	.	12-8-3
0A	LF	Line Feed		0-9-5	2F	/	/	/	0-1
0B	VT	K <sup>C</sup>		12-9-8-3	30	0	0	0	0
0C	FF	L <sup>C</sup>		12-9-8-4	31	1	1	1	1
0D	CR	Return		12-9-8-5	32	2	2	2	2
0E	SO	N <sup>C</sup>		12-9-8-6	33	3	3	3	3
0F	SI	O <sup>C</sup>		12-9-8-7	34	4	4	4	4
10	DLE	P <sup>C</sup>		12-11-9-8-1	35	5	5	5	5
11	DC1	Q <sup>C</sup>		11-9-1	36	6	6	6	6
12	DC2	R <sup>C</sup>		11-9-2	37	7	7	7	7
13	DC3	S <sup>C</sup>		11-9-3	38	8	8	8	8
14	DC4	T <sup>C</sup>		9-8-4	39	9	9	9	9
15	NAK	U <sup>C</sup>		9-8-5	3A	:	:	:	8-2
16	SYN	V <sup>C</sup>		9-2	3B	;	;	;	11-8-6
17	ETB	W <sup>C</sup>		0-9-6	3C	<	,: <sup>S</sup>	<	12-8-4
18	CAN	X <sup>C</sup>		11-9-8	3D	=	-: <sup>S</sup>	=	8-6
19	EM	Y <sup>C</sup>		11-9-8-1	3E	>	.: <sup>S</sup>	>	0-8-6
1A	SUB	Z <sup>C</sup>		9-8-7	3F	?	/ <sup>S</sup>	?	0-8-7
1B	ESC	K <sup>CS</sup>		0-9-7	40	@	p <sup>S</sup>	@	8-4
1C	FS	L <sup>CS</sup>		11-9-8-4	41	A	A	A	12-1
1D	GS	M <sup>CS</sup>		11-9-8-5	42	B	B	B	12-2
1E	RS	N <sup>CS</sup>		11-9-8-6	43	C	C	C	12-3
1F	US	O <sup>CS</sup>		11-9-8-7	44	D	D	D	12-4
20	Blank	Space Bar			45	E	E	E	12-5
21	!	1 <sup>S</sup>	!	12-8-7	46	F	F	F	12-6
22	"	2 <sup>S</sup>	"	8-7	47	G	G	G	12-7
23	#	3 <sup>S</sup>	#	8-3	48	H	H	H	12-8
24	\$	4 <sup>S</sup>	\$	11-8-3	49	I	I	I	12-9
					4A	J	J	J	11-1

ANSI Hex Code	Character	Teletype Key	Teletype/Printer Graphic	Hollerith Card Code	ANSI Hex Code	Character	Teletype Key	Teletype/Printer Graphic	Hollerith Card Code
4B	K	K	K	11-2	66	f		F	12-0-6
4C	L	L	L	11-3	67	g		G	12-0-7
4D	M	M	M	11-4	68	h		H	12-0-8
4E	N	N	N	11-5	69	i		I	12-0-9
4F	O	O	O	11-6	6A	j		J	12-11-1
50	P	P	P	11-7	6B	k		K	12-11-2
51	Q	Q	Q	11-8	6C	l		L	12-11-3
52	R	R	R	11-9	6D	m		M	12-11-4
53	S	S	S	0-2	6E	n		N	12-11-5
54	T	T	T	0-3	6F	o		O	12-11-6
55	U	U	U	0-4	70	p		P	12-11-7
56	V	V	V	0-5	71	q		Q	12-11-8
57	W	W	W	0-6	72	r		R	12-11-9
58	X	X	X	0-7	73	s		S	11-0-2
59	Y	Y	Y	0-8	74	t		T	11-0-3
5A	Z	Z	Z	0-9	75	u		U	11-0-4
5B	[	K <sup>S</sup>	[	12-8-2	76	v		V	11-0-5
5C	\	L <sup>S</sup>	\	0-8-2	77	w		W	11-0-6
5D	]	M <sup>S</sup>	]	11-8-2	78	x		X	11-0-7
5E	†	N <sup>S</sup>	†	11-8-7	79	y		Y	11-0-8
5F	—	O <sup>S</sup>	—	0-8-5	7A	z		Z	11-0-9
60	∅ or ¢		@	8-1	7B	{			12-0
61	a		A	12-0-1	7C				12-11
62	b		B	12-0-2	7D	}			11-0
63	c		C	12-0-3	7E	⌋ or ~			11-0-1
64	d		D	12-0-4	7F	RUBOUT			12-9-7
65	e		E	12-0-5					

## APPENDIX B. MESSAGES

### FILE ALREADY EXISTS

Commands: SAVE (NEW option)

Cause: An attempt was made to save a file using a name which already exists in the user's file directory.

Action: 1. Use the OLD option to overwrite the existing file.  
2. Change the name of the file specified as NEW.

### FILE CANNOT BE SHARED

Commands: DUPLICATE, SAVE, LOAD

Cause: The command has specified a file which is not shared, or for which the user has not entered the correct passkey values at the EXECUTIVE level.

Action: Return to the EXECUTIVE level and enter the correct passkey values if the file is shared.

### FILE DOES NOT EXIST

Commands: DUPLICATE, SAVE (OLD option), LOAD

Cause: A non-existent file was referenced by the command.

Action: 1. Correct the file name.  
2. Use the NEW option with the SAVE command.

### FILE TOO BIG

xxxx.xx LAST RECORD AFFECTED

Commands: LOAD

Cause: The file is too large for the EDITOR workspace. .

Action: None. The file must be edited in the workspace in segments.

### FILE TYPE INCORRECT

Commands: COPY, LOAD, SAVE (WITH option)

Cause: 1. The file specified is a binary file.  
2. The WITH option was used in a SAVE command which caused an intermixing of fixed- and variable-length records in the same file.  
3. A fixed-length record specification was used in a SAVE command (with the WITH option) which does not agree with the record length originally specified for the file.

Action: 1. Reenter proper file name.  
2. If adding records to a fixed-length record file specify record length to be compatible with original declaration of record length for the file.

## REARRANGING WORKSPACE

Commands: None  
Cause: Internal conditions necessitate some operation that may cause a delay.  
Action: None. EDITOR will prompt the user when it is ready to accept information.

## RECORD NUMBER OVERLAP

Commands: SAVE (WITH option), ENTER, REPLACE  
Cause: Record numbers generated by the command overlap the range specified.  
Action: Change STEP size or change record numbers in the destination area to provide more room for incoming records.

## RECORD NUMBER OVERLAP

xxxxx.xx LAST RECORD AFFECTED

Commands: DUPLICATE, MOVE  
Cause: Record numbers generated by the command overlap the range specified. The user is informed of the last source record affected by the command (before the overlap condition).  
Action: Change STEP size or change record numbers in the destination area to provide more room for incoming records.

## SYNTAX ERROR

Commands: All  
Cause: The command is syntactically incorrect. An upward arrow (↑) points to the section of the command in error. The terminal is automatically placed in extended editing mode.  
Action: Reenter the command correctly or use the special editing facilities described under the ALTER and MODIFY commands.

## WORKSPACE FULL

Commands: ENTER, REPLACE, ALTER, MODIFY  
Cause: The EDITOR workspace has been filled to capacity; the record currently being entered cannot be accommodated.  
Action: SAVE the contents of the workspace and reenter the file into the workspace on a segment basis.

## WORKSPACE FULL

xxxxx.xx LAST RECORD AFFECTED

Commands: DUPLICATE, MOVE, CHANGE  
Cause: The EDITOR workspace has been filled to capacity; the record currently being entered cannot be accommodated. The user is informed of the last source record affected by the command (before the workspace full condition).  
Action: SAVE the contents of the workspace and reenter the file into the workspace on a segment basis.

## WORKSPACE NOT EMPTY

Commands: LOAD, QUIT

Cause: 1. An attempt was made to load a file into the workspace when it already contains records.  
2. An attempt was made to leave the EDITOR subsystem without clearing the contents of the workspace (by DELETE or SAVE).

Action: 1. Reissue the LOAD command using the CLEAR option.  
2. Reissue the QUIT command using the CLEAR option.

## xxxxx.xx FORM ERROR

Commands: DUPLICATE, MOVE, ENTER, REPLACE, LOAD, RENUMBER

Cause: The command has generated a record number that does not fit the form specification currently in effect (by default, or by use of the FORM command). This message is generated only at the first occurrence of a form error. Subsequent occurrences are signaled by an asterisk (\*) in the character position between the end of the record number and before the beginning of the text of a record.

Action: Change the current record number specification by using the FORM command.

## xxxxx.xx LAST RECORD AFFECTED

Commands: LOAD, MOVE, DUPLICATE, SAVE, CHANGE

Cause: The Escape (or Alt Mode) key was used to terminate the current operation. The user is informed of the last source record affected by the command (before ESC or Alt Mode was used).

Action: None

## xxxxx.xx LENGTH ERROR

Commands: ENTER, DUPLICATE, MOVE, REPLACE, MODIFY, CHANGE

Cause: The command has generated a record whose length exceeds the current record length specification.

Action: The record has been entered into the EDITOR workspace in its entirety. However, if the user attempts to save the record as part of a fixed-length record file, and the record exceeds the length specification in the SAVE command, the record will be truncated.

## xxxxx RECORDS AFFECTED

Commands: MOVE, DUPLICATE, DELETE, CHANGE

Cause: The system is informing the user of how many records were affected by the command.

Action: None. This message is informative only.



# APPENDIX C. MODEL 33 TELETYPEWRITER TERMINAL

The Model 33 Teletypewriter Terminal used by the TENET Timesharing System consists of a control unit, keyboard, paper tape punch, and paper tape reader mechanism.

## CONTROL UNIT

The configuration of the control unit ( see Figure C-1 ) depends on whether the terminal is direct or acoustically coupled to the computer. The control unit on a direct-coupled terminal consists of only a LINE/OFF/LOCAL knob.

- LINE** If the control is in the LINE position, the terminal should be on and connected to the computer (on-line) unless the terminal had been automatically disconnected. In this case turn the knob first to the OFF position, and then to the LINE position to establish a connection.
- OFF** The terminal is off and incapable of communicating with the computer.
- LOCAL** The terminal is on but not connected to the computer. When the terminal is in this mode (off-line), operations such as punching paper tape may be performed.

Acoustically coupled terminals require a headset mechanism which is used to hold a telephone receiver. The LINE/OFF/LOCAL knob for acoustically coupled terminals is the same as for direct-coupled terminals except that when the knob is in the LINE position, the terminal is not automatically connected to the computer, but is capable of being connected to the computer. To establish a connection, the user must first turn the knob to the LINE position, phone the computer site, wait for a high-pitched tone, and place the telephone receiver into the headset mechanism.

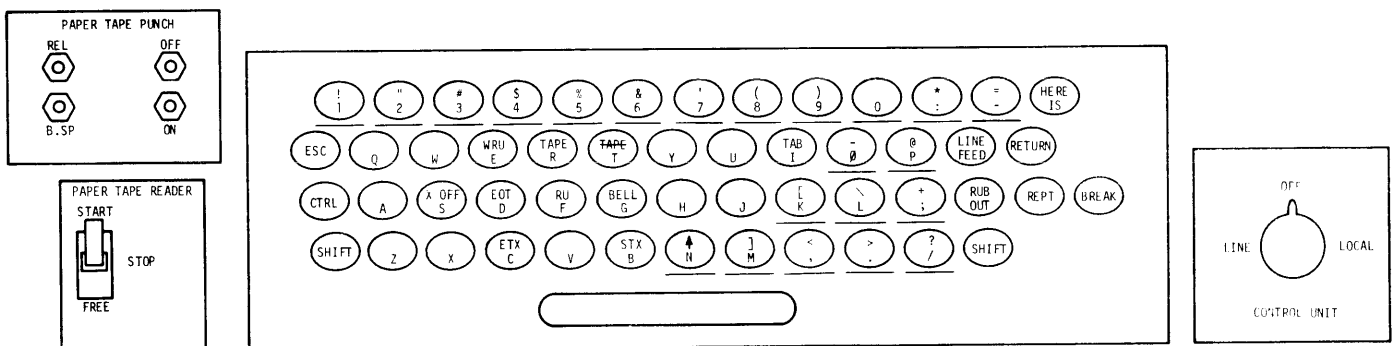


Figure C-1. Model 33 Teletypewriter Terminal Keyboard and Controls



## THE KEYBOARD

The teletype keyboard ( see Figure C-1 ) is used as a standard typewriter keyboard with the exception of the keys described below.

SHIFT	<p>Only those keys underlined in Figure C-1 have a shift position. The shift key is non-locking and must be depressed when typing. Characters are printed as they appear on the upper half of the key. However, on some terminals:</p> <p style="padding-left: 40px;">K shift is not marked but appears as a [</p> <p style="padding-left: 40px;">L shift is not marked but appears as a \</p> <p style="padding-left: 40px;">M shift is not marked but appears as a ]</p> <p>The keyboard locks whenever an attempt is made to use the shift with a key with no shift position.</p>
CTRL (Control)	<p>Any alphabetic character may be pressed in conjunction with CTRL. ( CTRL is non-locking. ) The resulting control character is not always printed at the terminal. Characters used with the CTRL are discussed in section 9 of this manual. They are designated by the subscript <sup>c</sup>. Control characters not recognized by the system are ignored but cause the bell to ring once for each ignored character.</p>
ESC or ALT MODE	<p>This key terminates any input/output operation in progress and causes a program interrupt.</p>
LINE FEED	<p>Each time the Line Feed key is pressed, the paper is advanced one line. ( When the terminal is connected to the computer, the system automatically generates a Carriage Return for each Line Feed. )</p>
RETURN or CARRIAGE RETURN	<p>This key positions the print head at the beginning of a line. When the terminal is connected to the computer ( on-line ), the system automatically generates a Line Feed for every Carriage Return.</p>
RUBOUT	<p>This key is used to delete characters on paper tape. It is always ignored, but does not ring the bell.</p>
REPT (Repeat)	<p>This key causes any character key pressed while the REPT is pressed to be repeated for as long as the REPT key is pressed.</p>
HERE IS	<p>Transmits and prints whatever is on the answerback drum.</p>
BREAK	<p>On a direct coupled terminal this key is ignored; on an acoustically coupled terminal, pressing this key disconnects the terminal from the computer.</p>

## PAPER TAPE PUNCH

The paper tape punch is used to produce a perforated tape which can be used as input to the computer instead of input from the teletypewriter terminal keyboard.

OFF and ON	The ON button initiates and continues paper tape punching until the OFF button is pressed. Information punched on paper tape is also printed at the terminal.
REL	The release button frees the paper tape so that the user can manually pull blank tape through the punch mechanism.
BKSP	The backspace button moves the paper tape backwards one frame each time the button is pressed. It is used in conjunction with the RUBOUT key to delete paper tape entries.

## PREPARING PAPER TAPE OFF-LINE

The user can save information in paper tape form. The TAPE command, for example, is used by the BASIC subsystem to access programs saved on paper tape. Similarly, the contents of data files may be punched on paper tape and later read into a file by the EXECUTIVE. To prepare paper tape off-line, turn the terminal control dial to LOCAL, depress the Punch ON button, and enter data from the keyboard. Since this is an off-line operation, the user will find it convenient to follow a Line Feed with a Carriage Return and vice versa. When paper tapes are read by the system, Line Feed / Carriage Return and Carriage Return / Line Feed combinations are treated as Line Feed and Carriage Return respectively.

The control keys  $\textcircled{A}^c$  ( delete previous character ) and  $\textcircled{Q}^c$  ( delete current line ) may be used to edit paper tape entries.

## PAPER TAPE READER

START	This control initiates and continues paper tape reading.
STOP	This key terminates paper tape reading.
FREE	This control frees the reader mechanism so that the tape can be pulled through the reader manually.

# INDEX

- A control key 4, 23, 25
- Accessing EDITOR 6
- Accessing saved files 32
- Account numbers 5
- ALL 9
- Alt Mode key 4, 13
- ALTER command 22
- ANSI character set 51
- Associative range 9
  
- BASIC files 32, 34
- Bell 5
- Binary files 2
- Blanks 7
- Break key 5
  
- C control key 22, 25
- Carriage Return key 3
- CHANGE command 27
- Character set
  - ANSI 51
  - EDITOR 6
- Clearing the workspace 33, 40
- Copying records 30
- Creating files 34
  
- D control key 4, 15, 22, 25
- Data files 2
- DELETE command 18
- Deleting
  - characters 4
  - entries 4
  - records 18
- Diagnostics 53
- Disconnecting the terminal 41
- DUPLICATE command 30
  
- E control key 22, 25
- Echoing 3
- Editing control characters 22, 25
- EDITOR
  - accessing 6
  - character set 6
  - commands 13
  - features 1
  - glossary 8
  - input to 6
  - syntax error checking 7
  - workspace 6
- END 9
- End of record 4
- End of text 15
- ENTER command 15
- EOT key 15, 22, 25
- Error message 7, 53
- ESC key 4, 13, 15
- Escape key 4, 13, 15
- EXECUTIVE 6, 40
- Explicit range 9
- Extended editing mode 7, 22, 25
  
- F 9
- Files 2
  - binary 2
  - content 2
  - data 2
  - names 32
  - program 2
  - protection 2
  - record type 2
  - retrieval 32
  - text 2
- FIRST 9
- Fixed-length record files 34
- FORM command 36
- Full duplex mode 3

Generating line numbers 15  
Glossary of EDITOR terms 8

I control key 39  
Intraline editing 27

Keys

- A control 4, 23, 25
- Alt Mode 4, 13
- Break 5
- Carriage Return 4
- D control 4, 15, 22, 25
- EOT 4, 15, 22, 25
- ESC 4, 13, 15
- Escape 4, 13, 15
- Line Feed 4
- Q control 4, 23, 25
- Rubout 5

L 9  
LAST 9  
Leaving EDITOR 40  
Leaving the system 41  
LENGTH command 38  
Line Feed key 4  
Linking files 34  
LIST command 17  
LOAD command 32  
LOGIN

- prompts 5
- sequence 5

LOGOUT sequence 41

Messages 53  
Mode of operation 3  
Model 33 teletypewriter terminal 57  
MODIFY command 25  
MOVE command 29

Occurrence constraints 10  
Overwriting files 34

Password 5  
Position constraint 10  
Program files 2

Prompts

- EDITOR 6
- EXECUTIVE 6
- LOGIN 5
- record number 15

Q control key 4, 23, 26  
QUIT command 40

Range

- associative 9
- constraints 9
- elements 9
- explicit 9
- specification in commands 13

Record

- constraint 10
- copying 30
- length 7, 38
- number format 8, 36
- number generation 15
- numbers, renumbering 21

RENUMBER command 21  
REPLACE command 19  
Rubout key 5

S control key 22, 25  
Sample terminal sessions 43  
SAVE command 34  
Shared file names 32  
Signing off 41  
Signing on 5  
Special characters for editing 22, 25  
Storing files 34  
Strings 9  
Syntax error checking 7

TABS command 39  
Teletypewriter keys 3  
Terminal 3, 57

- keyboard 57
- sample sessions 43

Text files 2  
Transmission interrupt 5  
User identification 5  
Workspace 6



FOLD

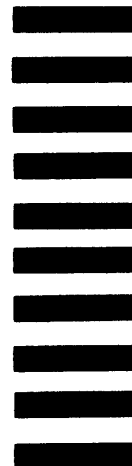
FIRST CLASS  
PERMIT NO. 480  
SUNNYVALE, CA.

BUSINESS REPLY MAIL  
NO POSTAGE STAMP NECESSARY IF MAILED IN U.S.A

POSTAGE WILL BE PAID BY

**TENET**  
927 THOMPSON PLACE  
SUNNYVALE, CA. 94086

ATTENTION: SOFTWARE PUBLICATIONS



FOLD

FROM: NAME \_\_\_\_\_

POSITION \_\_\_\_\_

ADDRESS \_\_\_\_\_

SAVE [range] 

NEW 'filename'
OLD 'filename'
WITH 'filename'

 [BASIC][REN[UMBER]][(no.)] p.34

Saves all or part of the contents of the workspace on disc storage.

FORM integer<sub>1</sub> [.integer<sub>2</sub>] p.36

Changes the standard format for record numbers.

LEN[GTH] integer p.38

Enables the user to maintain a fixed record length for records of less than 256 characters.

TABS integer<sub>1</sub>[,integer<sub>2</sub>[,integer<sub>3</sub>[,integer<sub>4</sub>]]] p.39

Establishes tab stops on the teletypewriter print line.

Q[UIT][CLEAR] p.40

Enables the user to leave the EDITOR subsystem and return to EXECUTIVE.

### EXTENDED EDITING MODE CHARACTERS

CHARACTER	FUNCTION
C <sup>c</sup>	Copies the next character from the old record to the new record; the copied character is echoed at the terminal.
S <sup>c</sup>	Skips the next character in the old record; a % is echoed at the terminal.
D <sup>c</sup>	Copies the remainder of the old record to the new record and terminates editing the old record; all copied characters are echoed at the terminal.
E <sup>c</sup>	Enter/Exit insert mode. Echoes '<' and '>', respectively. The text typed in the insert mode is entered into the new record and does not affect the old record.
A <sup>c</sup>	When the backspace key is used in extended editing mode, only the contents of the new record is affected.
Q <sup>c</sup>	When the delete current record key is used in extended editing mode, only the contents of the new line are affected.
Other	Other characters entered from the terminal replace corresponding characters in the old record.

**TENET, Inc. / 927 Thompson Place / Sunnyvale, California 94086 / (408) 245-8751**