



© Texas Instruments Incorporated 1981, 1982, 1983

All Rights Reserved, Printed in U.S.A.

The information and/or drawings set forth in this document and all rights in and to inventions disclosed herein and patents which might be granted thereon disclosing or employing the materials, methods, techniques or apparatus described herein, are the exclusive property of Texas Instruments Incorporated.

## MANUAL REVISION HISTORY

DNOS Text Editor Reference Manual (2270504-9701)

Original Issue .....	1 August 1981
Revision .....	1 October 1982
Revision .....	15 November 1983

The total number of pages in this publication is 118.

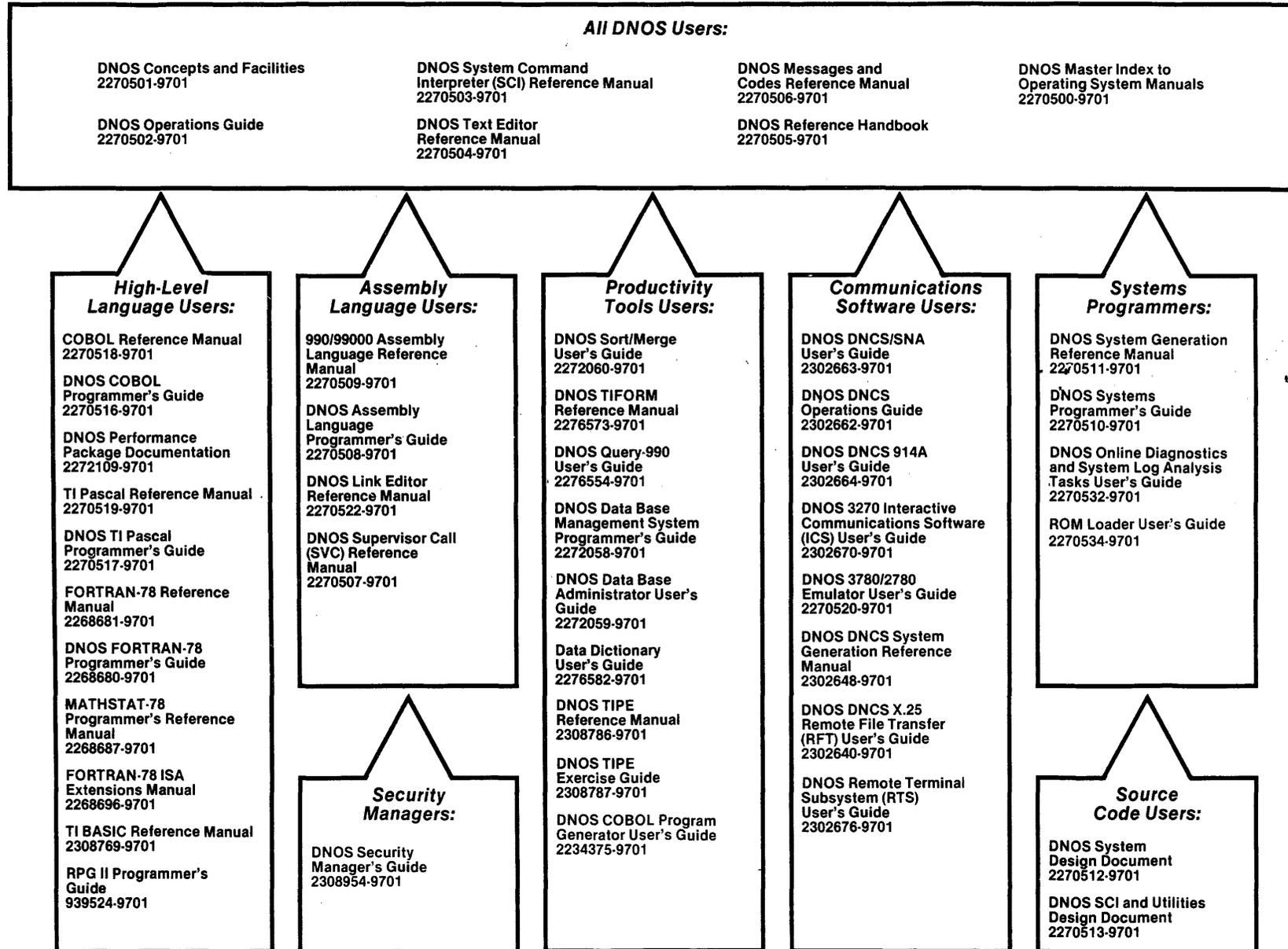
The computers offered in this agreement, as well as the programs that TI has created to use with them, are tools that can help people better manage the information used in their business; but tools—including TI computers—cannot replace sound judgement nor make the manager's business decisions.

Consequently, TI cannot warrant that its systems are suitable for any specific customer application. The manager must rely on judgement of what is best for his or her business.

# DNOS Software Manuals

This diagram shows the manuals supporting DNOS, arranged according to user type. Refer to the block identified by your user group and all blocks above that set to determine which manuals are most beneficial to your needs.

2270504-9701



# DNOS Software Manuals Summary

## **Concepts and Facilities**

Presents an overview of DNOS with topics grouped by operating system functions. All new users (or evaluators) of DNOS should read this manual.

## **DNOS Operations Guide**

Explains fundamental operations for a DNOS system. Includes detailed instructions on how to use each device supported by DNOS.

## **System Command Interpreter (SCI) Reference Manual**

Describes how to use SCI in both interactive and batch jobs. Describes command procedures and gives a detailed presentation of all SCI commands in alphabetical order for easy reference.

## **Text Editor Reference Manual**

Explains how to use the Text Editor on DNOS and describes each of the editing commands.

## **Messages and Codes Reference Manual**

Lists the error messages, informative messages, and error codes reported by DNOS.

## **DNOS Reference Handbook**

Provides a summary of commonly used information for quick reference.

## **Master Index to Operating System Manuals**

Contains a composite index to topics in the DNOS operating system manuals.

## **Programmer's Guides and Reference Manuals for Languages**

Contain information about the languages supported by DNOS. Each programmer's guide covers operating system information relevant to the use of that language on DNOS. Each reference manual covers details of the language itself, including language syntax and programming considerations.

## **Performance Package Documentation**

Describes the enhanced capabilities that the DNOS Performance Package provides on the Model 990/12 Computer and Business System 800.

## **Link Editor Reference Manual**

Describes how to use the Link Editor on DNOS to combine separately generated object modules to form a single linked output.

## **Supervisor Call (SVC) Reference Manual**

Presents detailed information about each DNOS supervisor call and DNOS services.

## **DNOS System Generation Reference Manual**

Explains how to generate a DNOS system for your particular configuration and environment.

## **User's Guides for Productivity Tools**

Describe the features, functions, and use of each productivity tool supported by DNOS.

## **User's Guides for Communications Software**

Describe the features, functions, and use of the communications software available for execution under DNOS.

## **Systems Programmer's Guide**

Discusses the DNOS subsystems and how to modify the system for specific application environments.

## **Online Diagnostics and System Log Analysis Tasks User's Guide**

Explains how to execute the online diagnostic tasks and the system log analysis task and how to interpret the results.

## **ROM Loader User's Guide**

Explains how to load the operating system using the ROM loader and describes the error conditions.

## **DNOS Design Documents**

Contain design information about the DNOS system, SCI, and the utilities.

## **DNOS Security Manager's Guide**

Describes the file access security features available with DNOS.

# Preface

---

This manual contains operating instructions for the use of the DNOS Text Editor on both video display terminals (VDTs) and hard-copy terminals. It is designed to be used as a tool for learning the Text Editor operation and as a reference manual. You need to have a basic understanding of DNOS directory structures and pathnames and know the log-on procedure for your system to use this manual.

Read Sections 1, 2, and 3 completely to familiarize yourself with the Text Editor commands and the proper use of the edit control keys. Perform the exercises, presented in Section 4 or 5 as appropriate for your terminal, to assist you in mastering all the Text Editor commands and functions. These exercises apply the information presented in the first three sections.

This manual contains five sections and one appendix as follows:

## Section

- 1 Text Editor Operation — Presents an overview of Text Editor usage.
- 2 Edit Control Functions — Describes each of the edit control function keys provided to assist you while using the Text Editor.
- 3 Text Editor Commands — Presents a detailed description of each Text Editor command and provides an example of its use.
- 4 Exercises Using a Video Display Terminal — Provides detailed instructions for use of the Text Editor on a VDT.
- 5 Exercises Using a Hard-Copy Terminal — Provides detailed instructions for use of the Text Editor on a hard-copy terminal.

## Appendix

- A Keycap Cross-Reference — Provides cross-references from generic names to the names on the keycaps of supported terminals.

100

1

1

1



# Contents

---

Paragraph	Title	Page
<b>1 — Text Editor Operation</b>		
1.1	Introduction . . . . .	1-1
1.2	Generic Keycap Names . . . . .	1-2
1.3	Activating the Text Editor . . . . .	1-2
1.4	Data Entry on a Video Display Terminal . . . . .	1-3
1.5	Data Entry on a Hard-Copy Terminal . . . . .	1-4
1.6	Command Selection and Error Recovery . . . . .	1-4
1.7	Access Rights on Secured Files . . . . .	1-5
1.8	Terminating the Text Editor . . . . .	1-5
<b>2 — Edit Control Functions</b>		
2.1	Introduction . . . . .	2-1
2.2	Edit Control Functions on a VDT Keyboard . . . . .	2-1
2.2.1	Return to Command Mode Function — Command Key . . . . .	2-3
2.2.2	Roll Up Function — F1 Key . . . . .	2-3
2.2.3	Roll Down Function — F2 Key . . . . .	2-3
2.2.4	Enable/Disable Word Wrap Function — F3 Key . . . . .	2-3
2.2.5	Duplicate to Tab Function — F4 Key . . . . .	2-3
2.2.6	Clear to Tab Function — F5 Key . . . . .	2-3
2.2.7	Display/Suppress Line Numbers Function — F6 Key . . . . .	2-4
2.2.8	Edit Mode/Compose Mode Function — F7 Key . . . . .	2-4
2.2.9	Restore Line Function — F8 . . . . .	2-4
2.2.10	New Line Function — Return Key . . . . .	2-4
2.2.11	Forward Tab Function — Forward Tab Key . . . . .	2-5
2.2.12	Erase Rest of Line Function — Skip Key . . . . .	2-5
2.2.13	Forward Space One Character Function — Next Character Key . . . . .	2-5
2.2.14	Backspace One Character Function — Previous Character Key . . . . .	2-5
2.2.15	Back Tab Function — Back Tab or Previous Field Key . . . . .	2-5
2.2.16	Right Margin Function — Next Field Key . . . . .	2-5
2.2.17	Left Margin Function — Enter Key . . . . .	2-5
2.2.18	Erase Characters on Line Function — Erase Field Key . . . . .	2-5
2.2.19	Delete Line Function — Erase Input Key . . . . .	2-5
2.2.20	Insert Line Function — Initialize Input Key . . . . .	2-6
2.2.21	Repeat Function — Repeat Key . . . . .	2-6
2.2.22	Insert Character Function — Insert Character Key . . . . .	2-6
2.2.23	Delete Character Function — Delete Character Key . . . . .	2-6
2.2.24	Move Cursor Up Function — Previous Line Key . . . . .	2-6

Paragraph	Title	Page
2.2.25	Move Cursor Down Function — Next Line Key . . . . .	2-6
2.2.26	Move Cursor Right Function — Next Character Key . . . . .	2-7
2.2.27	Move Cursor Left Function — Previous Character Key . . . . .	2-7
2.2.28	Move Cursor Home Function — Home Key . . . . .	2-7
2.2.29	Enable/Disable Uppercase Function — Uppercase Lock Key . . . . .	2-7
2.2.30	Control Character Codes Function — Control Key . . . . .	2-7
2.3	Edit Control Functions on a Hard-Copy Terminal . . . . .	2-7
2.3.1	Roll Up Function — CTRL A . . . . .	2-9
2.3.2	Roll Down Function — CTRL B . . . . .	2-9
2.3.3	Enable/Disable Word Wrap Function — CTRL C . . . . .	2-9
2.3.4	Duplicate to Tab Function — CTRL D . . . . .	2-10
2.3.5	Clear to Tab Function — CTRL E . . . . .	2-10
2.3.6	Display/Suppress Line Numbers Function — CTRL F . . . . .	2-10
2.3.7	Move Printhead Left Function — BACKSPACE/CTRL H . . . . .	2-10
2.3.8	Forward Tab Function — CTRL I . . . . .	2-10
2.3.9	Move Printhead Down Function — CTRL J . . . . .	2-10
2.3.10	Erase Rest of Line (SKIP) Function — CTRL K . . . . .	2-10
2.3.11	Erase Entered Characters Function — RUB OUT/DEL . . . . .	2-11
2.3.12	Move Printhead Home Function — CTRL L . . . . .	2-11
2.3.13	New Line Function — RETURN/CTRL M . . . . .	2-11
2.3.14	Delete Line Function — CTRL N . . . . .	2-11
2.3.15	Insert Line Function — CTRL O . . . . .	2-11
2.3.16	Left Margin Function (733 Only) — CTRL S . . . . .	2-11
2.3.17	Back Tab Function — CTRL T . . . . .	2-11
2.3.18	Move Printhead Up Function — CTRL U . . . . .	2-11
2.3.19	Edit Mode/Compose Mode Function — CTRL V . . . . .	2-11
2.3.20	Restore Line Function — CTRL W . . . . .	2-12
2.3.21	Return to Command Mode Function — CTRL X . . . . .	2-12
2.3.22	Left Margin Function (783, 820 Only) — CTRL Y . . . . .	2-12
2.3.23	Right Margin Function — CTRL Z . . . . .	2-12
2.3.24	Backspace, Nonerase Function — CTRL 3 . . . . .	2-13
2.3.25	Enable/Disable Uppercase Function — UPPERCASE . . . . .	2-13
2.3.26	Control Character Codes Function — CTRL . . . . .	2-13

### 3 — Text Editor Commands

3.1	Introduction . . . . .	3-1
3.1.1	Field Prompt Responses . . . . .	3-1
3.1.2	Field Prompt Notation . . . . .	3-5
3.1.3	Command Description Format . . . . .	3-6
3.2	Text Editor Command Descriptions . . . . .	3-8
3.3	CL (Copy Lines) . . . . .	3-9
3.4	DL (Delete Lines) . . . . .	3-11
3.5	DS (Delete String) . . . . .	3-13
3.6	FS (Find String) . . . . .	3-15
3.7	IF (Insert File) . . . . .	3-17

Paragraph	Title	Page
3.8	MHR (Modify Horizontal Roll) .....	3-19
3.9	ML (Move Lines) .....	3-20
3.10	MR (Modify Roll) .....	3-22
3.11	MRM (Modify Right Margin) .....	3-23
3.12	MT (Modify Tabs) .....	3-24
3.13	QE (Quit Edit) .....	3-25
3.14	RE (Recover Edit) .....	3-29
3.15	RS (Replace String) .....	3-31
3.16	SL (Show Line) .....	3-33
3.17	SVL (Save Lines) .....	3-34
3.18	XE (Execute Text Editor) .....	3-36
3.19	XES (Execute Text Editor With Scaling) .....	3-39

#### 4 — Exercises Using a Video Display Terminal

4.1	Introduction .....	4-1
4.2	Activating the System Command Interpreter .....	4-1
4.3	Exercise 1 — Text Editing Session .....	4-2
4.3.1	Activating the Text Editor .....	4-2
4.3.2	Establishing a Format .....	4-3
4.3.3	Entering Data .....	4-4
4.3.4	Terminating the Text Editor .....	4-5
4.3.5	Printing the File .....	4-6
4.4	Exercise 2 — Revising an Existing File .....	4-7
4.4.1	Using the Modification Listing File .....	4-11

#### 5 — Exercises Using a Hard-Copy Terminal

5.1	Introduction .....	5-1
5.2	Activating the System Command Interpreter .....	5-1
5.3	Exercise 1 — Text Editing Session .....	5-2
5.3.1	Activating the Text Editor .....	5-2
5.3.2	Establishing a Format .....	5-3
5.3.3	Entering Data .....	5-4
5.3.4	Terminating the Text Editor .....	5-5
5.3.5	Printing the File .....	5-5
5.4	Exercise 2 — Revising an Existing File .....	5-6
5.4.1	Using the Modification Listing File .....	5-10

## Appendix

Appendix	Title	Page
A	Keycap Cross-Reference .....	A-1

## Index

## Illustrations

---

Figure	Title	Page
4-1	Input for Memorandum Using a VDT .....	4-5
4-2	Display of the Input File on a VDT .....	4-7
4-3	Revisions to Memorandum Using a VDT .....	4-8
4-4	Modification Listing File for the VDT Exercise .....	4-12
5-1	Input for Memorandum Using the 820 KSR .....	5-4
5-2	Display of the Input File on the 820 KSR .....	5-6
5-3	Text Editor Output File on the 820 KSR .....	5-9
5-4	Modification Listing File for the 820 Exercise .....	5-10

## Tables

---

Table	Title	Page
2-1	Edit Control Functions on a VDT .....	2-2
2-2	Edit Control Functions on a Hard-Copy Terminal .....	2-8
3-1	Response Type Indicators .....	3-2
3-2	Legal Device Names .....	3-3
3-3	Field Prompt Notation .....	3-6
3-4	Command Description Format .....	3-7

# Text Editor Operation

---

## 1.1 INTRODUCTION

The Text Editor allows you to create and modify files of data with minimum effort. It is designed primarily for use with video display terminals (VDTs) but also can be used on hard-copy terminals such as the Model 820 KSR and the Model 733 ASR/KSR.

This section of the manual contains a general description of the features and operation of the Text Editor. Throughout the manual, references to VDTs are applicable to Models 931 and 911 VDTs, Business System Terminals, and 940 Electronic Video Terminals (EVTs). Text Editor commands and functions operate in the same manner, regardless of which video terminal is used.

The Text Editor operates under the DNOS System Command Interpreter (SCI), the interface between the user and the operating system. SCI accepts commands entered from a terminal and interprets them to perform the desired functions. An SCI command activates the Text Editor.

When the Text Editor is active, you can use the edit control *keys* to enter, modify, delete, and display data. You also can select Text Editor *commands*, which affect the entire file, to display, move, and modify data.

Edit control functions are similar for both a VDT and a hard-copy terminal. However, the means of invoking a particular function can vary, since the VDT and hard-copy terminal keyboards are not identical. The term *hard-copy terminal* includes teleprinter devices (TPDs).

You can text edit any sequential or relative record file that contains 240 character positions or less per line (that is, 240-column records). An attempt to edit an existing file that has records exceeding 240 columns results in data being truncated and lost.

You specify the maximum line length when you begin the edit. Records longer than this are truncated. Records shorter than specified are blank filled. The specified line length is also used in creating the output file when you terminate the edit (if the output file does not already exist).

Although you can text edit a file as often as you want, you must terminate the Text Editor before you can edit a different file.

## 1.2 GENERIC KEYCAP NAMES

The names used in this manual for individual keys on keyboards are generic names and apply to all terminals. In some cases the names on the keycaps of the terminals match the generic key names, but in many cases they do not. You can tell which key the generic name refers to by consulting Table A-1 in Appendix A. There you can find cross references from the generic name to the keycap of each terminal supported by DNOS. Appendix A also contains keyboard diagrams showing the layout of each terminal, and a reverse reference table showing frequently used keys on the Model 911 VDT and the corresponding generic name.

## 1.3 ACTIVATING THE TEXT EDITOR

The Execute Text Editor (XE) SCI command activates the Text Editor. On a VDT, the Execute Text Editor with Scaling (XES) command also activates the Text Editor, and it displays a scale at the bottom of the VDT screen indicating column position. On a hard-copy terminal, the XES command writes a scaling line before writing the first line to be edited. When either of these commands is entered, field prompts are displayed as follows:

```
[ ]XE
EXECUTE TEXT EDITOR
      FILE ACCESS NAME:
      EXCLUSIVE EDIT?:   YES
      LINE LENGTH:      80
```

In response to the FILE ACCESS NAME prompt, you can:

- Enter the file name of an existing sequential or relative record file to be edited.
- Specify that the Text Editor is to create a sequential file, by leaving the response field blank and pressing the Return key. (On a hard-copy terminal, this is specified by entering two quotation marks (" ") and pressing the Return key.) When you quit text editing, the file is automatically created by the Text Editor with the file name you enter in response to one of the Quit Edit (QE) command field prompts.

In response to the EXCLUSIVE EDIT? prompt, you can:

- Enter YES (the default). In this case you have exclusive write access to the file specified.
- Enter NO. In this case other users share access to the file specified.

In response to the LINE LENGTH prompt, you can:

- Enter an even integer between 80 and 240. This number is the maximum line length of the file specified. Internally, this value is used to set the record length of a relative record scratch file used by the Text Editor.
- Accept the default record length of 80 column records by pressing the Return key.

You can suspend the Text Editor by pressing the Command key to return control to SCI. If you then reenter the same edit, the values for the EXCLUSIVE EDIT? and LINE LENGTH prompts are not displayed. You can specify different values for these prompts at the beginning of a new edit session. If you specify a longer record length than the file being edited, the records are padded with blanks.

#### NOTE

If you specify a shorter record length, the records are truncated at the point of the specified record length, which could result in lost data.

If you do not specify an input file or if the input file you have specified contains no data records, only the end-of-file mark (\*EOF) is displayed on your terminal.

If the input file contains data records, the beginning of that file is displayed on your terminal. Each line (record) of the input file is numbered relative to the beginning of the file. If you delete a line, add a new line, or change the characters on a line, these modifications are not incorporated into the original input file. They are placed in a work file called the modification file. The data displayed at the terminal reflects the changes made while editing because the Text Editor builds the display by temporarily applying the contents of the modification file to the input file. However, the contents of the input file are not changed until you enter a Quit Edit (QE) command and replace the input file with the edited file.

You can edit both sequential and relative record files with the Text Editor. To edit a relative record file that does not already exist, you must first create the file using the Create File (CF) SCI command. The file name of a relative record file must be specified when the Text Editor is activated and when it is terminated. You can terminate an edit to an existing relative record file also. (The input file could have been null, sequential, or relative record.)

Section 3 describes the XE and XES commands in detail.

### 1.4 DATA ENTRY ON A VIDEO DISPLAY TERMINAL

Data entered at the terminal keyboard is displayed on the VDT screen. The screen display consists of 24 lines, each with 80 column positions. The Model 940 VDT is capable of displaying 132 characters, but only 80 characters (the normal display mode) should be used in conjunction with the Text Editor. You can change any line appearing on the screen by positioning the cursor (a high-intensity block of light) over the character where the correction is to be made and making the change from the keyboard. You can insert, delete, or modify characters by typing over them. Changes can be made in any order and are displayed in high intensity. Special keys on the keyboard control cursor movement and activate functions that manipulate data in the file. These edit control keys are described in detail in Section 2. You can edit lines longer than 80 characters by scrolling the line left or right as necessary. The keys and conditions that cause this scrolling are also described in Section 2.

You can position the file contents for display with the Show Line (SL) or Find String (FS) command or by use of various edit control keys called the F1, F2, Next Line, Previous Line, Next Character and Previous Character keys.

Section 4 contains step-by-step exercises using the Text Editor on a VDT.

## **1.5 DATA ENTRY ON A HARD-COPY TERMINAL**

The hard-copy terminal prints one line (record) at a time on the terminal printer. Each line is numbered to indicate its position in the input file. Data is entered on a hard-copy terminal by typing or retyping the data, and then pressing the Return key at the end of each line on which you have changed data.

The edit control functions described in Section 2 and the Text Editor commands described in Section 3 are used to display, modify, and manipulate data in the file. For example, when you want to revise a file, print the entire file and mark the changes to be made. You then can use the Show Line (SL) or the Find String (FS) command to print and modify only those lines that require revision.

Section 5 contains step-by-step exercises using the Text Editor on a hard-copy terminal.

## **1.6 COMMAND SELECTION AND ERROR RECOVERY**

While text editing, you can use any of the SCI commands and also certain commands that are operative only when the Text Editor is active. The Text Editor commands perform operations that affect the entire file, such as copying, moving, or deleting blocks of text, and locating particular lines or character strings within the file. Each Text Editor command is presented in Section 3 with an example of its use and notes regarding special applications.

To execute a command, first press the Command key to place SCI in command mode and to cause the SCI prompt ([ ]) to be displayed. Then type the name of the desired command and press the Return key. Any field prompts associated with the command are displayed so that you can enter the parameters required for the operation to be performed.

If you enter a command or field prompt response incorrectly, an error message is displayed under the last prompt on the VDT screen (or printed, if you are using a hard-copy terminal). The error messages typically encountered while text editing are easy to understand and remedy. (Also, you can enter the Show Expanded Message (SEM) SCI command or a question mark (?) to obtain a further explanation of the error.)

After you have read the message, press the Return key and enter the correct command or response. Depending on the message, you may want to return to the position in the file where you were editing prior to the error message, to determine if any changes have been made. Do this by entering the XE or XES command to resume text editing. Further information about each error message and appropriate recovery procedures is contained in the *DNOS Messages and Codes Reference Manual*.

## 1.7 ACCESS RIGHTS ON SECURED FILES

If you are using the Text Editor to edit a file on a system that has file security support, you must have the appropriate access right to perform an operation involving that file. There are five possible types of access rights to a file.

- Read Access — the right to read or copy a file
- Write Access — the right to write data to a file
- Execute Access — the right to execute tasks, segments, procedures, and overlays within a program file
- Delete Access — the right to delete a file
- Control Access — the right to change access to a file by group or to change the access groups associated with a file

To text edit a file you need read, write, and/or delete access to that file. You can find which rights are needed for each text editor operation by consulting the command descriptions in Section 4 of this manual. If you cannot perform the desired operation with the access rights you have, consult the security manager of your system.

## 1.8 TERMINATING THE TEXT EDITOR

Use the Quit Edit (QE) command to terminate the text editing session. The following field prompt is displayed:

```
QUIT EDIT
  ABORT?:      NO
```

If your response to the ABORT? field prompt is Y or YES, the work file created by the Text Editor during the editing session is discarded, the Text Editor is terminated, and the SCI prompt ([ ]) is displayed. The file being edited is not changed.

If your response is N or NO to the ABORT? field prompt, additional field prompts are displayed as follows:

```
QUIT EDIT
  OUTPUT FILE ACCESS NAME:
                REPLACE?:      NO
  MOD LIST ACCESS NAME:
```

In response to the OUTPUT FILE ACCESS NAME prompt you must specify the destination for the output file. You can enter the:

- File name of the original input file if it is to be replaced by the output file
- File name for a file that the Text Editor is to create
- File name of another existing file if it is to be replaced by the output file

The output file is built by the Text Editor from the records in the work file and copies of records from the input file, if one was specified when the Text Editor was activated. The original input file is modified only if you use the input file pathname for the output file, then respond YES to the REPLACE? prompt.

If you specify a file name that the Text Editor is to create, a new output file is created and classified as sequential (if the input was sequential or null) or relative record (if the input was relative record). Any previously existing file having the specified output file name will be replaced by the output file.

If you specify another existing file, then its characteristics (logical record length, relative record or sequential, blank suppressed, and so forth) are used. If the output file does not exist but there was an input file, the input file's characteristics are used to create the output file (the logical record length is what was specified on the XE command). If neither file exists, then default values are used.

In response to the REPLACE? prompt you can enter Y or YES to specify that an existing file is to receive the output of the Text Editor. Respond N or NO to indicate that the output file is to be automatically created by the Text Editor. If an error message is returned when N or NO is entered, the output file you have specified already exists. This feature prevents accidental replacement of existing file contents.

The MOD LIST ACCESS NAME prompt gives you the option of retaining a copy of the Text Editor modification listing. Each of the records modified during the editing session is listed in its original form and then in its modified form. The response to this prompt can be one of the following:

- Name of the file (or device) that is to receive the modification listing. If the file name entered already exists, it is automatically replaced with the listing. If the file name specified does not exist, the Text Editor will automatically create a sequential file 80 characters wide regardless of the length you specified for the XE LINE LENGTH prompt. (You must have previously created any directories included in the file name.)
- A null response. Specify that no listing is to be provided by leaving the response field blank (or deleting any value that is supplied) and pressing the RETURN key.

The Quit Edit (QE) command is described in detail in Section 3.

# Edit Control Functions

---

## 2.1 INTRODUCTION

Edit control functions (implemented through specific terminal keys) can change the editing position in the file, insert and delete data, move the cursor, display or suppress line numbers, and change editing modes of operation. The functions described in this section are operational when the Text Editor has been activated. Some of these keys have different functions when the Text Editor is not active. Paragraph 2.2 describes the edit control functions provided on a VDT keyboard; paragraph 2.3 describes the functions provided on a hard-copy terminal keyboard.

## 2.2 EDIT CONTROL FUNCTIONS ON A VDT KEYBOARD

On the VDT keyboard, designated keys activate all of the edit control functions. The outcome resulting from the use of many of these keys depends on parameters set by one of the following Text Editor commands, which are explained in Section 3:

- Modify Horizontal Roll (MHR)
- Modify Right Margin (MRM)
- Modify Roll (MR)
- Modify Tabs (MT)

The left margin setting (the lowest numbered tab stop) and the right margin setting limit the action of some of the edit control keys and affect the positioning of the cursor.

The edit control functions are listed in Table 2-1 with the corresponding keys that control them. Each function is described in detail in a subsequent paragraph.

**Table 2-1. Edit Control Functions on a VDT**

Function	Generic Key Name
Return to command mode	Command
Roll (display) up	F1
Roll (display) down	F2
Enable/disable word wrap	F3 <sup>1</sup>
Duplicate to tab	F4
Clear to tab	F5
Display/suppress line numbers	F6 <sup>2</sup>
Edit mode/compose mode	F7 <sup>1</sup>
Restore line	F8
New line (next line)	Return
Forward tab	Forward Tab
Erase rest of line	Skip
Forward space one character	Next Character
Backspace one character	Previous Character
Back tab	Back Tab or Previous Field
Right margin	Next Field <sup>3</sup>
Left margin	Enter <sup>3</sup>
Erase characters on line	Erase Field
Delete line	Erase Input
Insert line	Initialize Input
Repeat	Repeat
Insert character	Insert Character
Delete character	Delete Character
Move cursor up	Previous Line
Move cursor down	Next Line
Move cursor right	Next Character
Move cursor left	Previous Character
Move cursor home	Home
Enable/disable uppercase	Uppercase Lock <sup>1</sup>
Control character codes	Control

**Notes:**

<sup>1</sup> Alternates modes each time key is pressed.

<sup>2</sup> Alternates display of line numbers (74 data characters per line with no display of line numbers (80 data characters per line) each time key is pressed.

<sup>3</sup> The Next Field key and Enter key also function as horizontal scroll keys.

### 2.2.1 Return to Command Mode Function — Command Key

The System Command Interpreter (SCI) is the interface between the terminal user and the operating system. The SCI command mode is active when you log on to the system. When you enter the Execute Text Editor (XE) command, control passes to the Text Editor. You can reactivate SCI by pressing the Command key.

If an edit session is in progress, reactivating SCI causes the Text Editor to be suspended. You then can execute most SCI commands as well as Text Editor commands. Entering any Text Editor command restores control to the Text Editor.

### 2.2.2 Roll Up Function — F1 Key

The F1 key activates the roll up function, which repositions the display by rolling it up (toward the end of the file) a specified number of lines. The number is determined by the roll parameter, which can be changed using the Modify Roll (MR) Text Editor command.

### 2.2.3 Roll Down Function — F2 Key

The F2 key activates the roll down function, which repositions the display by rolling it down (toward the beginning of the file) a specified number of lines. The number is determined by the roll parameter, which can be changed by the Modify Roll (MR) Text Editor command.

### 2.2.4 Enable/Disable Word Wrap Function — F3 Key

The F3 key is an alternate action key that enables or disables the word wrap function if you are in compose mode. (The F3 key has no effect if you are not already in the compose mode.) This feature aids in the entry of large volumes of data. When you activate this feature, the Text Editor automatically inserts a blank line following the current line if you attempt to type past the end of the current line. It also moves any nonblank characters (word) at the end of the current line to the leftmost tab position of the new line. Specifically, upper and lowercase alphabetic characters, numbers, and the \$ character will be moved to the next line. The characters (, [, and < will also be copied, but characters immediately preceding these characters will not be copied. The cursor is positioned after the last character to await further input.

### 2.2.5 Duplicate to Tab Function — F4 Key

Pressing the F4 key causes the data from the previous line to be copied into the current line, from the cursor position up to the next tab stop to the right, then repositions the cursor at that tab stop. For example, if the cursor is in column 35 and the next tab stop is in column 65, characters from columns 35 through 64 on the previous line are copied into the corresponding columns of the current line.

If there is no tab stop to the right of the cursor position, the data from the previous line is copied from the cursor position to the right margin setting. This can cause horizontal scrolling.

### 2.2.6 Clear to Tab Function — F5 Key

Pressing the F5 key inserts blanks from the current cursor column position to the next tab stop to the right, then repositions the cursor at that tab stop. For example, if the cursor is at column 35 and the next tab stop is column 65, the clear to tab function replaces the data from columns 35 through 64 with blanks.

If there is no tab stop to the right of the cursor position, the data is erased from the cursor position to the right margin setting. This can cause horizontal scrolling.

### **2.2.7 Display/Suppress Line Numbers Function — F6 Key**

The F6 key is an alternate action key that causes line numbers to be displayed or suppressed at the left of the screen. For example, if line numbers are currently displayed and the F6 key is pressed, the display reappears without line numbers and the text is shifted to the left margin. If the key is pressed again, the display reappears with line numbers. Consecutive renumbering does not occur during the text editing session. Therefore, new lines inserted during the current editing session are displayed without line numbers. Deleted lines are indicated by nonconsecutive numbering.

When line numbers are displayed, only 74 data characters per line (record) appear on the screen, but up to 240 characters are written into the output file, depending on the record length specified. Six additional columns can be displayed and changed by suppressing the line numbers with the F6 key.

### **2.2.8 Edit Mode/Compose Mode Function — F7 Key**

The F7 key alternately activates edit mode or compose mode. Successively pressing the key alternates the selection between these modes.

In edit mode, the cursor moves to the first tab stop position of the next existing line when the Return key is pressed. If you specify an existing input file when you execute the Text Editor, it is activated in edit mode.

In compose mode, a new blank-filled line is generated each time the Return key is pressed, and all lines above the new line are moved up one line on the display. The cursor moves to the first tab stop position of the new line. Therefore, compose mode is used to enter text when creating a new file or to insert a large volume of data into an existing file. If you do not specify an existing input file when the Text Editor is executed, it is activated in compose mode. All other edit control functions operate the same in either edit or compose mode.

### **2.2.9 Restore Line Function — F8**

Pressing the F8 key nullifies the result of pressing the Erase Input, Erase Field, or Skip key. You must press the F8 key immediately after you enter one of these keys to restore the line to its previous value. This is useful for restoring a line of data after accidentally pressing an erase or skip key.

### **2.2.10 New Line Function — Return Key**

Pressing the Return key moves the cursor from the line where it is positioned to the first tab stop on the next line. This function operates differently in edit mode than it does in compose mode (use the F7 key to toggle between modes).

If edit mode is active when the Return key is pressed, the cursor moves to the first tab stop position of the next line (no line is inserted). If the cursor is in the bottom line of the display screen, all lines roll up one line, and the next line in the file is displayed as the bottom line.

If compose mode is active when the Return key is pressed, a blank-filled line is inserted after the line that contains the cursor. The line with the cursor and all lines above it in the display are rolled up one line, and the cursor is repositioned in the first tab stop position of the inserted line.

**2.2.11 Forward Tab Function — Forward Tab Key**

Pressing the Forward Tab key positions the cursor at the tab stop to the right of the current cursor position within the line. If no tab stop is set between the current cursor position and the right margin setting, the cursor moves back to the first tab stop on the same line.

**2.2.12 Erase Rest of Line Function — Skip Key**

Pressing the Skip key replaces with blanks all the data on a line from the current cursor position to the right margin. The cursor remains in the same column position.

This function affects characters that are both on and off the display screen.

**2.2.13 Forward Space One Character Function — Next Character Key**

Each time you press the Next Character key, the cursor moves one column to the right. The cursor cannot be moved further to the right than the right margin setting. Pressing this key when the cursor is at the right edge of the screen causes horizontal scrolling one column at a time if the right margin is not on the screen.

**2.2.14 Backspace One Character Function — Previous Character Key**

Pressing the Previous Character key moves the cursor one character to the left, within the line. If the cursor is in column 1 of the line, it remains there when you press this key. Pressing this key when the cursor is at the left edge of the screen causes horizontal scrolling one column at a time if column 1 is not on the screen.

**2.2.15 Back Tab Function — Back Tab or Previous Field Key**

Pressing the Back Tab or Previous Field key moves the cursor to the first tab stop to the left of its current position within the line. If the cursor is positioned at the beginning of the line or if no tab stop is set between the current cursor position and the beginning of the line, the cursor moves to the rightmost tab stop (preceding the margin setting) on the same line.

**2.2.16 Right Margin Function — Next Field Key**

Pressing the Next Field key moves the cursor to the right margin position on the same line or to the last character position on that line visible on the display screen, whichever comes first. If you press the Next Field key and the cursor is at the rightmost column of the screen, the screen rolls a number of columns to the left, or until the right margin is encountered. The number of columns rolled depends on the value entered in the MHR command (default is 40 columns).

**2.2.17 Left Margin Function — Enter Key**

Pressing the Enter key moves the cursor to the left edge of the screen on the same line. If the Enter key is pressed and the cursor is in the leftmost column of the screen, the screen rolls to the right a number of columns or until column 1 is encountered. The number of columns rolled depends on the value entered via the MHR command (default is 40 columns).

**2.2.18 Erase Characters on Line Function — Erase Field Key**

Pressing the Erase Field key replaces with blanks all characters in the line between the first character position and the right margin. The cursor is repositioned at the first character position of the same line. This key affects all characters, both on and off the display screen.

**2.2.19 Delete Line Function — Erase Input Key**

Pressing the Erase Input key deletes the line that contains the cursor. All lines following the deleted line roll up one line, and the cursor moves to the left screen edge on the next line.

### **2.2.20 Insert Line Function — Initialize Input Key**

Pressing the Initialize Input key causes the line containing the cursor and all lines following it to roll down. A blank line is inserted, and the cursor is repositioned in the same column on the new blank-filled line generated by this action. If the cursor is on the bottom line of the screen, it appears in the same column on the blank-filled line generated at the bottom of the screen.

### **2.2.21 Repeat Function — Repeat Key**

You activate the repeat function by holding down the Repeat key and momentarily pressing the key having the character, symbol, or function you want to repeat. The character or function selected is repeated until you release the Repeat key.

### **2.2.22 Insert Character Function — Insert Character Key**

You activate the insert character function by pressing the Insert Character key. Once the function has been selected, any character typed is inserted at the current cursor position. The cursor, the character at the current cursor position, and any characters to the right of the cursor shift to the right as each inserted character is entered. You can insert characters until you press another one of the edit control keys or until the cursor reaches the right edge of the screen. If the right margin of the line is not on the screen, the screen will scroll until it is in view. However, note that *data characters at the right margin of the line are lost* as they are shifted to the right past the line margin.

When you select this function, characters not displayed on the screen also shift right as you insert characters and can be lost if shifted past the right margin. The word wrap function operates when you insert characters, if the cursor moves past the right margin.

### **2.2.23 Delete Character Function — Delete Character Key**

Pressing the Delete Character key deletes the character at the current cursor position. All characters on that line to the right of the deleted character (between the deleted character and the right margin) are shifted left one column position. The rightmost column affected by this displacement is blank filled.

The delete character function *affects all characters on the line out to the right margin*. If there are any characters beyond the right margin (caused by changing the margin setting) they are retained in their current positions.

### **2.2.24 Move Cursor Up Function — Previous Line Key**

Pressing the Previous Line key moves the cursor to the same column position in the previous line. If the cursor is currently in the top line of the screen, all lines are rolled down one line, and the cursor remains in the same column position within the newly displayed top line. If the cursor is in the first line of the file, it remains there.

### **2.2.25 Move Cursor Down Function — Next Line Key**

Pressing the Next Line key moves the cursor to the next line, in the same column position as in the previous line. If the cursor is in the bottom line of the display when the function is called, all lines are rolled up one line. The cursor remains in the same column position within the newly displayed bottom line. If the cursor is in the last line of the file, it remains there.

**2.2.26 Move Cursor Right Function — Next Character Key**

Each time you press the Next Character key, the cursor moves one column to the right. You cannot move the cursor further to the right than the right margin setting. Pressing this key when the cursor is at the right edge of the screen causes horizontal scrolling if the right margin is off the screen. The text scrolls left one column each time you press the key.

**2.2.27 Move Cursor Left Function — Previous Character Key**

Each time you press the Previous Character key, the cursor moves one column to the left. Pressing this key when the cursor is at the left edge of the screen causes horizontal scrolling if column 1 is not on the screen. Text will scroll right one column each time you press the key.

**2.2.28 Move Cursor Home Function — Home Key**

Pressing the Home key positions the cursor in the first tab stop position of the first line of the display (the home position).

**2.2.29 Enable/Disable Uppercase Function — Uppercase Lock Key**

The Uppercase Lock key is a special alternate action locking key. If you place this key in the down (locked) position, the keyboard cannot generate lowercase letters. Only the keys A through Z are affected. If the key is in the up (unlocked) position, all keys operate the same as a standard typewriter keyboard.

**2.2.30 Control Character Codes Function — Control Key**

When you press and hold the Control key while pressing certain other keys, ASCII control character codes are generated. For the 911 VDT, some of the useful printable symbols are as follows:

CONTROL 8	(broken vertical bar)
CONTROL 9	' (back apostrophe)
CONTROL 0	~ (tilde)
CONTROL =	\ (back slash)
CONTROL ;	{ (left brace)
CONTROL '	} (right brace)

**2.3 EDIT CONTROL FUNCTIONS ON A HARD-COPY TERMINAL**

You activate some edit control functions on a hard-copy terminal with designated keys. Other functions require that you hold down the CTRL key and simultaneously press the key that indicates the particular edit control function to be executed. This action generates a unique control character and is represented in this manual by the word CTRL followed by the character.

The generic keycap names on the chart in Appendix A are valid for hard-copy terminals. However, most hard-copy terminals have the same keycap designation for each function; in this case, there is no advantage in using generic names. Therefore, the rest of Section 2 uses keycap names instead of generic names.

The outcome resulting from the use of many of these keys depends on parameters set by one of the following Text Editor commands, which are explained in Section 3:

- Modify Horizontal Roll (MHR)
- Modify Right Margin (MRM)
- Modify Roll (MR)
- Modify Tabs (MT)

The left margin setting (the lowest numbered tab stop) and the right margin setting limit the action of some of the edit control keys and affect the positioning of the printhead. The current column position in the file is indicated on the column counter of an 820 terminal.

The edit control functions are listed in Table 2-2 with the corresponding keys that control these functions; they are described in detail in subsequent paragraphs.

**Table 2-2. Edit Control Functions on a Hard-Copy Terminal**

Edit Control Function	Hard-Copy Terminal Key(s) Pressed
Roll Up	CTRL A
Roll Down	CTRL B
Enable/Disable Word Wrap	CTRL C <sup>1</sup>
Duplicate to Tab	CTRL D
Clear to Tab	CTRL E
Display/Suppress Line Numbers	CTRL F <sup>2</sup>
Move Printhead Left and Erase Character	CTRL H
Backspace Printhead	BACKSPACE <sup>3</sup>
Forward Tab	CTRL I
Move Printhead Down	CTRL J
Erase Rest of Line (Skip)	CTRL K
Erase Entered Characters	RUB OUT (733) or DEL (820) <sup>4</sup>
Move Printhead (to first tab on line)	CTRL L
New Line (Next Line)	CTRL M, RETURN
Delete Line	CTRL N
Insert Line	CTRL O
Left Margin (733 only)	CTRL S <sup>5</sup>
Back Tab (Left Field)	CTRL T
Move Printhead Up	CTRL U
Edit Mode/Compose Mode	CTRL V <sup>1</sup>
Restore Line	CTRL W
Return to Command Mode	CTRL X

**Table 2-2. Edit Control Functions on a Hard-Copy Terminal (Continued)**

Edit Control Function	Hard-Copy Terminal Key(s) Pressed
Left Margin (820 only)	CTRL Y <sup>5</sup>
Right Margin	CTRL Z <sup>6</sup>
Enable/Disable Uppercase	UPPERCASE <sup>1</sup>
Backspace Without Erasing Character	CTRL 3 <sup>7</sup>
Control Character Codes	CTRL

**Notes:**

- <sup>1</sup> Alternates modes each time the key is pressed.
- <sup>2</sup> Alternates display of line numbers (74 data characters) with no display of line numbers (80 data characters) each time key is pressed.
- <sup>3</sup> CTRL H on 783 KSR terminal.
- <sup>4</sup> Erases characters entered by user. It does not erase the whole line, as the Erase Field key does on a VDT.
- <sup>5</sup> Can cause horizontal scrolling if the starting position is the leftmost column of the line.
- <sup>6</sup> Can cause horizontal scrolling if starting position is the rightmost column of the line.
- <sup>7</sup> Unlike CTRL H, CTRL 3 backspaces for preexisting as well as user entered characters.

**2.3.1 Roll Up Function — CTRL A**

CTRL A activates the roll up function, which repositions the file by rolling it up (toward the end of the file) a specified number of lines and printing the line that is at the current file position after the roll. The number of lines rolled is determined by the roll parameter, which can be changed using the Modify Roll (MR) Text Editor command.

**2.3.2 Roll Down Function — CTRL B**

CTRL B activates the roll down function, which repositions the file by rolling it down (toward the beginning of the file) a specified number of lines and printing the line that is at the current file position after the roll. The number of lines rolled is determined by the roll parameter, which can be changed by the Modify Roll (MR) Text Editor command.

**2.3.3 Enable/Disable Word Wrap Function — CTRL C**

The CTRL C key is an alternate action key that enables or disables the word wrap function if you are in compose mode. (This key has no effect if you are not already in compose mode.) This feature aids in the entry of large volumes of data. When activated, the Text Editor automatically inserts a blank line following the current line when an attempt is made to type past the end of the current line. It also moves any non-blank characters at the end of the current line to the leftmost tab position of the new line. Specifically, upper and lowercase alphabetic characters, numbers, and the \$ character will be moved to the next line. The characters (, [, and < will also be copied, but no characters occurring immediately prior to them will be copied. The printhead is positioned after the last character to await further input.

### **2.3.4 Duplicate to Tab Function — CTRL D**

Pressing CTRL D causes the data from the previous line in the file to be copied into the current line, from the printhead position up to the next tab stop to the right. The printhead is repositioned at that tab stop. The printhead moves to the right margin if no tab stop is found. For example, if the printhead is in column 35 and the next tab stop is at column 65, characters from columns 35 through 64 on the previous line in the file are copied into the corresponding columns of the current line.

### **2.3.5 Clear to Tab Function — CTRL E**

Pressing CTRL E inserts blanks from the current printhead column position to the next tab stop to the right, then repositions the printhead at that tab stop. The printhead moves to the right margin if no tab stop is found. For example, if the printhead is at column 35 and the next tab stop is at column 65, the data is replaced with blanks from columns 35 through 64.

### **2.3.6 Display/Suppress Line Numbers Function — CTRL F**

CTRL F is an alternate action key that causes line numbers to be displayed or suppressed at the left of the printed line. Each time you press the CTRL F key, this function alternates between printing line numbers and not printing line numbers.

For example, if line numbers are currently displayed and you activate this function, the lines are printed without the line numbers and the text is shifted to the left margin. If CTRL F is pressed again, the lines are printed with line numbers. Consecutive renumbering does not occur during the text editing session. Therefore, new lines inserted during the current editing session are displayed without line numbers and any lines deleted are indicated by nonconsecutive numbering.

When line numbers are displayed, only 74 data characters per line are printed, but up to 240 characters per line (depending on line length specified for XE prompt) are written into the output file, padded with blanks.

### **2.3.7 Move Printhead Left Function — BACKSPACE/CTRL H**

The BACKSPACE key moves the printhead left one column position. Pressing CTRL H also moves the printhead left one column and erases the character from the file. CTRL H only backspaces over (and erases) characters you have entered. It does not backspace over characters written on the line by the Text Editor. Use CTRL 3 to backspace (without erasure) over characters you have entered, as well as over those written on the line by the Text Editor.

### **2.3.8 Forward Tab Function — CTRL I**

Pressing CTRL I causes the data on that line to be printed from the current printhead position to the next tab stop to the right. If there is no tab stop between the current printhead position and the end of the line, the line up to the right edge of the terminal is printed (that is, retained in the file), and the printhead is positioned to the first tab position on the same line. The tab stops are set by the Modify Tabs (MT) Text Editor command.

### **2.3.9 Move Printhead Down Function — CTRL J**

Pressing CTRL J moves the printhead to the next line in the file, prints the data, then repositions the printhead at the beginning of the line.

### **2.3.10 Erase Rest of Line (SKIP) Function — CTRL K**

This function replaces all of the characters with blanks, from the current printhead position through the right end of the line. The printhead does not move when CTRL K is selected.

**2.3.11 Erase Entered Characters Function — RUB OUT/DEL**

Pressing the RUB OUT key (on the Model 733 ASR) or the DEL key (on the Model 820) erases characters you enter. It does not erase the whole line, as the Erase Field key does on a VDT.

**2.3.12 Move Printhead Home Function — CTRL L**

Pressing CTRL L positions the printhead to the first tab stop position of the display line.

**2.3.13 New Line Function — RETURN/CTRL M**

Pressing the RETURN key or CTRL M moves the printhead from the line on which it is positioned to the next line and prints that line. This function operates differently in edit mode than in compose mode (CTRL V).

If edit mode is active when the RETURN (or CTRL M) key is pressed, the printhead moves to the first tab stop position of the next line (no line is inserted).

If compose mode is active when the RETURN (or CTRL M) key is pressed, a blank-filled line is inserted after the line currently containing the printhead. The printhead is repositioned in the first tab stop position of the inserted line.

**2.3.14 Delete Line Function — CTRL N**

Pressing CTRL N deletes the current line (the line just printed) from the file and causes the next line in the file to be printed. The printhead is positioned to the left edge of the display area.

**2.3.15 Insert Line Function — CTRL O**

Pressing CTRL O causes a blank-filled line to be inserted into the file at the file position immediately preceding that of the current printed line. The printhead position does not change. (It will be on the new blank line.)

**2.3.16 Left Margin Function (733 Only) — CTRL S**

Pressing CTRL S on the Model 733 ASR causes the printhead to be repositioned to the leftmost position of the current line. If it is already in the leftmost position, then the line will scroll horizontally to the left a specified number of columns or until the printhead is in column 1. The number of columns scrolled is set via the MHR command (default is 40 characters). CTRL S functions like the Enter key on a VDT.

**2.3.17 Back Tab Function — CTRL T**

On a hard-copy terminal, CTRL T causes the current line to be reprinted up to the previous tab stop to the left of the current printhead position. You must have at least one tab stop set. If there is no previous tab stop, the current line is reprinted up to the rightmost tab stop (that is, the print wraps around as it does on a VDT).

**2.3.18 Move Printhead Up Function — CTRL U**

Pressing CTRL U causes the previous line in the file to be printed.

**2.3.19 Edit Mode/Compose Mode Function — CTRL V**

CTRL V alternately activates edit mode or compose mode. Successively entering CTRL V alternates the selection between these modes.

When the RETURN key is pressed in edit mode, the printhead moves to the first tab stop position of the next existing line after printing the line. If you specify an existing input file when you execute the Text Editor, it is activated in edit mode.

In compose mode, a new blank-filled line is generated each time the RETURN key is pressed. The printhead moves to the first tab stop position of the new line. Therefore, compose mode is used to enter text when creating a new file or to insert a large volume of data into an existing file. If you do not specify an existing input file when the Text Editor is executed, it is activated in compose mode. All other edit control functions operate the same in either edit or compose mode.

### **2.3.20 Restore Line Function — CTRL W**

Pressing CTRL W allows you to negate the result of CTRL N, RUBOUT (or DEL), or CTRL K. You must use CTRL W immediately after entering one of these other functions to restore the line to its previous value. This is useful for restoring the line after accidentally entering one of the erase keys.

### **2.3.21 Return to Command Mode Function — CTRL X**

The System Command Interpreter (SCI) is the interface between the terminal user and the operating system. SCI command mode is active when you log on to the system and can be reactivated by pressing CTRL X on the keyboard.

If an edit session is already in progress, activating SCI causes the Text Editor to be suspended. You can then execute most SCI commands as well as the Text Editor commands. Enter XE, or any Text Editor command, to restore control to the Text Editor.

### **2.3.22 Left Margin Function (783, 820 Only) — CTRL Y**

Pressing CTRL Y on the Model 783 or 820 KSR causes the printhead to be repositioned to the leftmost position of the current line. If it is already in the leftmost position, then the line will scroll horizontally to the left a specified number of columns or until the printhead is in column 1. The number of columns scrolled is set via the MHR command (default is 40 characters). CTRL Y functions like the Enter key on a VDT.

### **2.3.23 Right Margin Function — CTRL Z**

Pressing CTRL Z causes the printhead to move to the right margin position or the last displayed character on that line, whichever comes first. Horizontal scrolling will occur if this function is entered when the printhead is in the rightmost column position of the display line and the right margin is off the display line. The number of columns scrolled is set via the MHR command (default is 40 characters).

**2.3.24 Backspace, Nonerase Function — CTRL 3**

Pressing CTRL 3 backspaces the printhead without erasing the character. This function works over characters you have entered, as well as characters written by the printer.

**2.3.25 Enable/Disable Uppercase Function — UPPERCASE**

The UPPERCASE key is a special, alternate-action locking key. If you place this key in the down (locked) position, the keyboard cannot generate lowercase letters. Only the keys A through Z are affected. If the key is in the up (locked) position, all keys operate the same as a standard typewriter keyboard.

**2.3.26 Control Character Codes Function — CTRL**

When the CTRL key is held in the down position while certain other keys are pressed, ASCII control character codes are generated that activate assigned functions.



# Text Editor Commands

---

## 3.1 INTRODUCTION

The commands used exclusively while text editing are described in detail in this section. To execute any Text Editor command, press the Command key to place SCI in command mode. Then type the command name and press the Return or Skip key. The following information is applicable to all Text Editor commands.

### 3.1.1 Field Prompt Responses

Text Editor commands have associated field prompts that are displayed at your terminal after you enter a command. They request information needed to execute the operation properly. Field prompts that are followed by a question mark require a YES (Y) or NO (N) response. Some prompts supply an initial value or character string that represents a common response or that is determined by some previously entered response. These are supplied simply to assist you, and they can be replaced by typing another value. Also, the operating system supplies default values for many of the response fields. Press the Return key after you have typed the response. Use the Previous Field key if you need to return to a previous response field.

You can respond to the command prompts in uppercase or lowercase characters for all commands except Delete String (DS), Find String (FS), and Replace String (RS). These three commands require you to specify precisely your intended string in whatever combination of uppercase and lowercase characters you need. If your SCI accepts lowercase characters (if you have set .OPTION LOWERCASE = YES), you must enclose your intended string in a set of quotation marks (“ ”) if it contains lowercase characters.

If you do not want to enter a value (null response) and the response field is blank, press the Return key. When an initial value is displayed, press the Skip key instead. (Enter “ ” on a hard-copy terminal.)

A feature provided for VDT operation allows you to accept all of the field prompt response values displayed. When you press the Enter key, the responses for all field prompts are accepted and the command executes.

Each command description specifies the valid response type that you can enter for a particular prompt. Table 3-1 lists and defines these response types. Table 3-2 lists the valid device names that can be entered for the device name response type.

**Table 3-1. Response Type Indicators**

<b>Response Type</b>	<b>Definition</b>
Pathname	Name used to access an I/O resource. A pathname can include channel name, device name, file name or station name. You can also specify a pathname by a synonym, a synonym followed by a pathname (synonym.pathname), a logical name, or a logical name followed by a pathname (logical.name.pathname). Legal characters in pathnames include uppercase alphabetic characters (A through Z), numerals (0 through 9), period (.), dollar sign (\$), left bracket ([), right bracket (]), and back slash (\). The name must start with an alphabetic character.
Device name	Name of a device (for example, DS01 and ST01).
Filename	Pathname of a file. A concatenation of a disk or volume name, any directory-level names leading to a file, and a final component name of a file. You can also specify a file by a synonym, a synonym followed by a file name (synonym.filename), a logical name, or a logical name followed by a file name (logical.name.filename).
Site	Name of a computer installation in a DNOS network. A site name can be from 1 to 8 alphanumeric characters beginning with an alphabetic character. You must have the Distributed Network I/O package installed to be able to use a site name.
Station name	A station ID in the range of ST01 through ST99. You can obtain the station ID of your station by entering the Show Terminal Information (STI) command.
YES/NO	The response to a prompt can be YES, NO, Y, or N.
Integer	Hexadecimal or decimal digits. Precede hexadecimal numbers by entering a right angle bracket (>) or a leading zero.
Integer exp	Decimal or hexadecimal value or expression composed of decimal or hexadecimal integers and the operators add (+), subtract (-), multiply (*), and divide (/).
Integer exp list	A list of decimal or hexadecimal values or expressions, separated by commas.
Full exp	Integer expression with the additional operators ( ) to indicate evaluation order and < > to show contents of indicated memory address registers. String operands are also permitted. In the debugger controlled mode, symbolic names and the symbols #PC, #WP, #ST, and #R0, #R1, #R2,...,#R15 are permitted. This type is unique to the SCI debugger.
Full exp list	More than one full expression separated by commas.
Alphanumeric	A string of alphabetic and/or numeric characters or a dollar sign (\$), starting with an alphabetic character. (Used with user IDs, volume names, and so on.)
Character(s)	Letters, digits, punctuation marks, or other symbols restricted in format or content by the context.
Line number	Numeric expression composed of an integer or integers and the operators plus (+) and minus (-).

Table 3-2. Legal Device Names

Device Name	Device
CMxx	A communication device specified by a device name in the range of CM01 through CM99.
CRxx	A card reader specified by a device name in the range of CR01 through CR99.
CSxx	A cassette unit of a Model 733 ASR terminal specified by a device name in the range of CS01 through CS99.
DSxx	A disk unit or an FD1000 diskette unit, specified by a device name in the range of DS01 through DS99.
ME	The terminal at which a command is entered. You can find the station ID of your terminal by entering the Show Terminal Information (STI) command.
MTxx	A magnetic tape unit specified by a device name in the range of MT01 through MT99.
LPxx	A line printer specified by a device name in the range of LP01 through LP99.
SDxx	A user-configured special device in the range of SD01 through SD99.
STxx	A terminal specified by a device name in the range of ST01 through ST99. You can find the station ID of your terminal by entering the Show Terminal Information (STI) command. The device name STxx, where xx is your terminal ID number, is equivalent to entering ME.

Two of these response types, *filename* and *line number*, are used frequently while text editing and must be thoroughly understood.

**File Names.** The following paragraphs summarize file names and file structure in relation to the Text Editor. Detailed descriptions of DNOS file structure, file names, and synonyms are contained in the *DNOS System Command Interpreter (SCI) Reference Manual*.

You can store one or more files under a group heading, called a directory. You can group one or more directories (and one or more files) under another directory heading. A directory is created by the Create Directory File (CFDIR) SCI command. You must create a directory before you create other directories or files that are to be grouped under that directory. The Text Editor cannot automatically create a directory.

Files are created by the Create File (CF) SCI command. You must create relative record files prior to the text editing session; you can create sequential files prior to the editing session or the Text Editor can create them automatically.

A file is accessed by its file name. The file name can consist of the volume name of the disk on which the file is stored, followed by one or more directory names under which the files are grouped, then the particular file name. These file name components are separated by periods and are composed of alphanumeric characters. The first character of each component must be alphabetic. Each component is eight or fewer characters in length. The total length of the file name must not exceed 48 characters, including periods. No spaces are permitted. Examples of valid file names are as follows:

```
.MYFILE  
VOL1.MYFILE  
VOL1.MAINDIR.MYFILE1  
VOL1.MAINDIR.SUBDIR1.MYFILE1  
VOL1.MAINDIR.SUBDIR1.MYFILE2
```

When DNOS is used in a network of systems, disk file names can have an additional component. This component is a site name which appears as the first part of the file name. The site name is followed by a colon (:) and the file name, as described above. For example, DALLAS:MYDIR.MYDIRA is a directory at site DALLAS on the volume MYDIR. The file TOLEDO:.OUTPUT.LISTING.RECORDS resides on the system disk at the site named TOLEDO. If no site name is used, the user's own system is assumed to be the intended site. The user may specify his own site name; DNOS will use that as a local name.

All Text Editor commands permit the use of synonyms to represent one or more components of the file name. A synonym is a character string that you define to represent another character string. You assign the synonym by using the Assign Synonym (AS) SCI command. Usually, the synonym is much shorter than the character string it represents and is used for convenience to specify a long file name.

*Line Numbers.* Many Text Editor command field prompts request line number values. Once you have entered data into a file, consecutive line numbers are displayed for each line (record) whenever you edit that file. However, lines that you insert, copy, or move during the current edit session are unnumbered. There are several ways you can specify the location of a line in response to a prompt that requests a line number. You can position the cursor anywhere in the line and then enter a null response to the field prompt when you execute the command. If the line is the beginning or ending line in the file, you can specify the character B or E to represent these respective lines. Or, you can identify the line by its line number or by an integer expression that describes the location of the line relative to an existing numbered line. For example, the relative line reference for a new line preceding line number 12 is the expression 12 - 1; and the relative line reference for the second of two new lines following line number 15 is 15 + 2.

Any one of the following is a valid response type to a field prompt that requests a line number:

<b>Response Type</b>	<b>Description</b>
B or E	The character B designates the beginning line in the file; E designates the ending line in the file.
i	The integer that is the number of the line in the current file. This value refers to an existing line. A zero value is not permitted.
i ± n	A line referenced by its offset value (n) from a numbered line (i). This value is used to reference a newly entered line. Neither i nor n can be zero.
± n	A line referenced by its offset value from the line where the cursor is positioned.
default	A line referenced by the cursor position. This is specified by entering a null response (leaving the response field blank and pressing the Return key). The line where the cursor is positioned is assumed to be the designated line.

Values for the START LINE and END LINE field prompts are optional. If you enter no value, the value of the line on which the cursor is positioned is used. If the value you entered for the starting line is greater than the value you enter for the ending line, the starting line value is also used as the ending line value.

### 3.1.2 Field Prompt Notation

Notation symbols enclose some field prompt responses in the command descriptions, as shown in Table 3-3, and are used in the descriptions of prompt responses throughout this manual to define how DNOS expects you to enter the response type.

Table 3-3. Field Prompt Notation

Notation	Meaning
Uppercase	Enter the response as listed.
Lowercase	Enter a response of this type.
No marks	The response is required.
[ ]	The response is optional.
{ }	The response must be exactly one of the enclosed items or must be of a type that is one of the enclosed items (the choices are separated by a slash).
Item...item	More than one item of this type can be entered to the response. Separate items by commas.
@	Synonyms are allowed as responses.
( )	Represents the initial value. If (*) is shown, the value can be supplied from a synonym set by a previously used command procedure.
/	Indicates alternate response types
-	When a hyphen (-) precedes a valid field prompt type, the initial and user responses are not echoed to the screen. In a batch stream the response is replaced by four dashes.

### 3.1.3 Command Description Format

The format for the presentation of the command descriptions in this section is consistent from command to command. Each description is divided into categories to aid you in locating particular command information. The format categories are listed in Table 3-4.

**Table 3-4. Command Description Format**

Category	Description
Command	This category includes the command keyword and its full descriptive name.
Prompts	This category includes the set of prompts for the command, the types of responses accepted, and an indication of any initial values supplied by the command procedure.
Purpose	This category includes a statement of the function performed by the command, its general uses, and rules of use.
Prompt Details	For each of the command prompts, this category describes what is required for the command, how the response can be specified, and any default values used when you do not supply a response.
Example	This category presents an example of how to use the command in a common way, show responses to the prompts, and describes the outcome of the command.
Special Cases	This category is included for commands that have nonobvious applications, such as using the Copy Directory command to copy program files. Few commands need this category of explanation.
Assumptions	This category includes assumptions that are not obvious from the statement of purpose or from the prompts of the command. The fact that a certain LUNO must be assigned or that the file or device must be in a certain state are examples of assumptions.
Messages	This category documents special completion messages, output formats, or other nonerror messages. This category does <i>not</i> include standard error messages but it does include a description of other messages you must notice.
Notes	This category includes special warnings, cautions, or reminders about the command.
Related Commands	This category lists the commands that perform opposite operations from the command being described (for example, Create File versus Delete File), perform similar operations to the same types of data, or in some fashion are similar to the described command.

If there is no need for a category to be included for a particular command description, that category has been excluded.

### 3.2 TEXT EDITOR COMMAND DESCRIPTIONS

Read the Text Editor command descriptions thoroughly to become familiar with each command. For ease in referencing the commands, they are presented in alphabetical order by the command name as follows:

CL	Copy Lines
DL	Delete Lines
DS	Delete String
FS	Find String
IF	Insert File
MHR	Modify Horizontal Roll
ML	Move Lines
MR	Modify Roll
MRM	Modify Right Margin
MT	Modify Tabs
QE	Quit Edit
RE	Recover Edit
RS	Replace String
SL	Show Line
SVL	Save Lines
XE	Execute Text Editor
XES	Execute Text Editor with Scaling

### 3.3 CL (COPY LINES)

*Prompts:*

**COPY LINES**

**START LINE:** [BEGINNING/END/line number]  
**END LINE:** [BEGINNING/END/line number]  
**INSERT AFTER LINE:** [BEGINNING/END/line number]

*Purpose:*

The CL command copies a designated set of lines in the current file to a new position in the file. The original lines remain intact in their original position in the current file. Approximately 400 80-character lines can be copied by each CL command, depending on the level of operating system activity. Correspondingly fewer lines can be copied for lines with a record length between 80 and 240 characters.

*Prompt Details:*

**START LINE:**

The integer or expression that specifies the first line of the set of lines to be copied to a new position in the file. The character B or E can be entered to specify the beginning or ending line, respectively, in the file. If no response is entered, the line on which the cursor is positioned, whether it is numbered or unnumbered, is assumed to be the first line of the set to be copied.

**END LINE:**

The integer or expression that specifies the last line of the set of lines to be copied to a new position in the file. The character B or E can be entered to specify the beginning or ending line, respectively, in the file.

This value should be the same as or greater than the START LINE value. If the value is less than the START LINE value, the START LINE value is used. If no response is entered, the line on which the cursor is positioned, whether it is numbered or unnumbered, is assumed to be the last line of the set to be copied.

**INSERT AFTER LINE:**

The integer or expression that specifies the line number after which the set of copied lines is to be placed. The character B or E can be entered to specify the beginning or ending line, respectively, in the file. If no response is entered, the line on which the cursor is positioned, whether it is numbered or unnumbered, is the line after which the copied lines are inserted.

**Example:**

The following example indicates lines 33 through 56 as the lines to be copied and inserted immediately after line 108 in the file.

```
COPY LINES
      START LINE: 33
      END LINE: 56
INSERT AFTER LINE: 108
```

When the lines have been copied, the file is positioned with the first line of the newly copied set displayed as the top line of the VDT screen (printed on the hard-copy terminal).

**Special Cases:**

A set of lines can be copied to the beginning of the current file by responding to the INSERT AFTER LINE prompt with the character B.

If the line number specified for START LINE or END LINE has been previously moved or deleted, an error message is displayed. If this occurs, enter the XE or XES command to return to the display that existed when the CL command was entered. If the END LINE value specified is greater than the last existing line number in the file, E (ending line) is assumed.

If the INSERT AFTER LINE number is greater than the last line number in the file, E (ending line) is assumed, and the set of lines is inserted immediately after the last existing line of the file.

You will get an error message if the Text Editor is unable to get system resources to copy the lines.

**Notes:**

Any modifications made to the lines prior to the copy process are included at the new position of the lines.

**Related Commands:**

DL	(Delete Lines)
IF	(Insert File)
ML	(Move Lines)
SL	(Show Line)
SVL	(Save Lines)

### 3.4 DL (DELETE LINES)

*Prompts:*

```
DELETE LINES
  START LINE: [BEGINNING/END/line number]
  END LINE:   [BEGINNING/END/line number]
```

*Purpose:*

The DL command deletes the designated set of lines from the current file. All lines from the START LINE value through the END LINE value are deleted.

*Prompt Details:*

**START LINE:**

An integer or expression that specifies the first line of the set of lines to be deleted. The character B or E can be entered to specify the beginning line or ending line, respectively, in the file. If no value is specified, the line on which the cursor is positioned is assumed to be the first line of the set of lines to be deleted.

**END LINE:**

An integer or expression that specifies the last line of the set of lines to be deleted. The character B or E can be entered to specify the beginning line or ending line, respectively, in the file. This value should be the same as or greater than the START LINE value. If the value is less than the START LINE value, the START LINE value is assumed. If no value is specified, the line on which the cursor is positioned is assumed to be the last line of the set to be deleted.

*Example:*

The following example indicates lines 33 through 56 as lines to be deleted from the current file. Upon completion of the command, the file is positioned so that the line immediately following the deleted set of lines contains the cursor and is displayed as the top line on the VDT screen (printed on a hard-copy device).

```
DELETE LINES
  START LINE: 33
  END LINE:   56
```

*Special Cases:*

If either of the line numbers specified has been deleted or previously moved, an error message is displayed. Enter an XE or XES command to return to the display that existed when the command was entered. If the END LINE value specified is greater than the last existing line in the file, E (ending line) is assumed.

*Related Commands:*

CL	(Copy Lines)
ML	(Move Lines)
SL	(Show lines)
SVL	(Save Lines)

### 3.5 DS (DELETE STRING)

*Prompts:*

```

DELETE STRING
  NUMBER OF OCCURRENCES: integer           (1)
                        START COLUMN: integer      (*)
                        END COLUMN: integer        (*)
                        STRING: [character(s)]     (*)

```

*Purpose:*

The DS command deletes a designated number of occurrences of a specified string from the file being edited. The command activates a search of each line of the file for the specified string. The search begins at the start column of the line containing the cursor. When the string is found, it is deleted, then found and deleted again until the specified number of occurrences is deleted. A null value is acceptable as a character string. If a null value is specified, the DS command deletes all characters between the specified columns (regardless of the characters in those columns) for the specified number of occurrences.

*Prompt Details:*

**NUMBER OF OCCURRENCES:**

An integer that specifies the number of deletions to be made of the specified string.

**START COLUMN:**

An integer that specifies the column number of each line at which the search for the specified string will begin.

**END COLUMN:**

An integer that specifies the last column of each line at which the search for the specified string will end.

**STRING:**

The character string that is to be deleted. Leading and trailing blanks are not recognized as part of the string unless the string and any leading and trailing blanks are enclosed in a set of double quotation (" ") marks. When you have set .OPTION LOWERCASE = YES, a string that includes lowercase characters also must be enclosed in a set of double quotation marks. If it is not, all lowercase characters are translated to uppercase.

*Example:*

In the following example, columns 1 through 70 are searched for the string, *preliminary*. The search begins with the current line and continues until four occurrences of the string are deleted.

```
DELETE STRING
  NUMBER OF OCCURRENCES: 4
    START COLUMN: 1
      END COLUMN: 70
        STRING: "preliminary"
```

After the fourth deletion of the string, the file is repositioned so that the line in which the last deletion occurred is displayed on the line on which the cursor is positioned (printed on a hard-copy device). The end-of-file mark is displayed if less than four occurrences are found and deleted or if the string is not found as specified.

*Notes:*

When selecting a character string, choose a unique set of characters to avoid modifying other data that contains the same character string. Leading and/or trailing blanks are useful in this respect.

If the character string contains two or more separate words, only those occurrences in which the entire string appears on one line are recognized and deleted. You must scan the margin of your file to locate any occurrences in which parts of the string may have been typed on two separate lines.

This command only searches forward to delete the specified string. To delete a string in an entire file, use the Show Line (SL) command to return to the beginning of the file, then perform the Delete String (DS) operation.

If you want to delete every occurrence of the string in the file, specify an excessively large number. When every occurrence of the string is deleted, the \*EOF is displayed.

When the string is found and deleted, all characters to the right of the string (and to the left of the right margin) are shifted left to fill the area occupied by the string. The remainder of the line is blank filled.

*Related Commands:*

FS	(Find String)
RS	(Replace String)

### 3.6 FS (FIND STRING)

*Prompts:*

```

FIND STRING
  OCCURRENCE NUMBER: integer      (1)
    START COLUMN: integer         (*)
      END COLUMN: integer         (*)
        STRING: character(s)     (*)

```

*Purpose:*

The FS command searches forward in the current file to locate a particular occurrence of a specified string. The command causes a search between specified columns of each line of the file, beginning with the line on which the cursor is positioned. The FS command positions the cursor on the string found.

*Prompt Details:*

**OCCURRENCE NUMBER:**

An integer that specifies the occurrence of the string to be located.

**START COLUMN:**

An integer that specifies the number of the column at which the search for the specified string will begin on each line.

**END COLUMN:**

An integer that specifies the last column of each line at which the search for the specified string will end.

**STRING:**

The character string that is to be found. Leading and trailing blanks are not recognized unless the entire string and all leading and/or trailing blanks are enclosed in a set of double quotation (" ") marks. When you have set .OPTION LOWERCASE = YES, a string that includes lowercase characters also must be enclosed in a set of double quotation marks. If it is not, all lowercase characters are translated to uppercase.

*Example:*

In the following example, columns 1 through 80 are searched for the string, REF, beginning with the current line and continuing until the fourth occurrence of the string is found.

```

FIND STRING
  OCCURRENCE NUMBER: 4
    START COLUMN: 1
      END COLUMN: 80
        STRING: REF

```

When the fourth occurrence of the string is found, the search terminates and the file is repositioned so that the line containing the string is displayed on the line on which the cursor is positioned (or printed on a hard-copy device). The end-of-file mark is displayed if less than four occurrences of the string are encountered within the limits of the search.

*Notes:*

The command only searches forward in the file. To search the entire file, use the Show Line (SL) command to return to the beginning of the file, then enter the Find String (FS) command.

When selecting a character string to be found, choose a unique set of characters. Leading and/or trailing blanks are useful in this respect.

If the character string contains two or more separate words, only those occurrences in which the entire string appears on one line are recognized. You must scan the margins of your file to locate any occurrence in which parts of the string may have been typed on two separate lines.

When searching for more than one occurrence of a string, move the cursor down one line from where the string has been located, press the Command key, then enter the command name followed by a period (FS.). This causes the entire set of Find String field prompts to be accepted with the parameters as previously entered.

*Related Commands:*

DS	(Delete String)
RS	(Replace String)

### 3.7 IF (INSERT FILE)

*Prompts:*

```
INSERT FILE
  INSERT AFTER LINE: [BEGINNING/END/line number]
  FILE PATHNAME: [site:]filename@
```

*Purpose:*

The IF command causes a copy of the contents of a second file to be inserted following a specified line within the file being edited.

*Prompt Details:*

**INSERT AFTER LINE:**

The integer or expression that specifies the line number in the file being edited after which the contents of the second file will be placed. The character B can be entered to specify the beginning line in the file, or the character E can be entered to specify the ending line in the file. If no line is specified, the line on which the cursor is positioned is assumed as the line after which to insert the file.

**FILE PATHNAME:**

The file name that specifies the file that is to be copied and inserted into the file being edited. The file that is copied must be relative record or sequential, and it remains intact after copying. If it is a file in a secure system, you must have read access rights to it or an error message appears on the screen when you enter the file name.

*Example:*

In the following example, a copy of the file .EDIT.INSERT is inserted following line 108 in the current file.

```
INSERT FILE
  INSERT AFTER LINE: 108
  FILE PATHNAME: .EDIT.INSERT
```

After this command executes, the last line of the inserted file and the cursor appear as the top line on the VDT screen (printed on the hard-copy terminal).

*Special Cases:*

If the file to be copied and inserted does not exist or cannot be accessed for any other reason, an error message is displayed. Reactivate the Text Editor using the XE or XES command.

If the line number specified for the INSERT AFTER LINE prompt is greater than the largest existing line number in the file, E (ending line) is assumed. The contents of the specified second file are inserted following the last existing line in the file being edited.

If the record length of the inserted file is less than the file being edited, the inserted file records are padded with blanks. If the record length of the inserted file is greater than the file being edited, the inserted file records are truncated at the point that they exceed the record length of the file being edited. This could cause loss of data in the columns truncated.

*Notes:*

If you use the IF command on a system with file security, you must have read access rights to the file which is specified in response to the FILE PATHNAME prompt. If you do not have access to the file and still want to perform the operation, see your security manager.

*Related Commands:*

**SVL** (Save Lines)

### 3.8 MHR (MODIFY HORIZONTAL ROLL)

*Prompts:*

```
MODIFY HORIZONTAL ROLL
NUMBER OF COLUMNS TO ROLL: integer (*)
```

*Purpose:*

The MHR command specifies the number of columns the file being edited will roll, right or left, when the horizontal roll function is activated. From 1 to 240 columns can be selected to be incremented or decremented from the current column position.

*Prompt Details:*

**NUMBER OF COLUMNS TO ROLL:**

The integer that specifies the number of columns to roll when a horizontal roll function is activated. A single column or any number of columns up to and including 240 can be specified. The command will accept a value greater than 240; however, the roll will be terminated when the right margin is encountered. If no value is assigned, the initial value of 40 is used.

*Example:*

The following example causes 80 columns of data to be rolled left or right when the horizontal roll function occurs.

```
MODIFY HORIZONTAL ROLL
NUMBER OF COLUMNS TO ROLL: 80
```

*Notes:*

When either the first column of the file or the last column (right margin) of the file is encountered during a horizontal roll operation, the roll terminates.

The MHR command affects only rolling of files being edited by the Text Editor; it does not affect roll functions in other SCI commands.

*Related Commands:*

MR	(Modify Roll)
MRM	(Modify Right Margin)
MT	(Modify Tabs)

### 3.9 ML (MOVE LINES)

*Prompts:*

**MOVE LINES**

START LINE: [BEGINNING/END/line number]  
END LINE: [BEGINNING/END/line number]  
INSERT AFTER LINE: [BEGINNING/END/line number]

*Purpose:*

The ML command moves a designated set of lines from one position to another position within the file being edited. You will get an error message if the Text Editor is unable to get the system resources to move the block. Approximately 400 80-character lines can be moved by each ML command, depending on the level of operating system activity. Correspondingly fewer lines can be moved for lines having a record length of 80 to 240 characters.

*Prompt Details:*

**START LINE:**

The integer or expression that specifies the first line of the set of lines to be moved. The character B or E can be entered to specify the beginning line or ending line, respectively, in the file. If no response is entered, the line on which the cursor is positioned (whether numbered or unnumbered) is assumed to be the first line of the set to be moved.

**END LINE:**

The integer or expression that specifies the last line of the set of lines to be moved to the new position in the file. The character B or E can be entered to specify the beginning line or ending line, respectively, in the file. The END LINE number should be equal to or greater than the START LINE number.

If the value specified is less than the START LINE value, the START LINE value is assumed. If no response is entered, the line on which the cursor is positioned (whether numbered or unnumbered) is assumed to be the last line of the set of lines to be moved.

**INSERT AFTER LINE:**

The integer or expression that specifies the line number after which the set of moved lines will be placed. The character B or E can be entered to specify the beginning line or ending line, respectively, in the file. If no response is entered, the moved lines are inserted after the line on which the cursor is positioned (whether numbered or unnumbered).

**Example:**

In the following example, lines 33 through 56 are moved to a new position following line 108 in the file.

```
MOVE LINES
          START LINE: 33
          END LINE: 56
          INSERT AFTER LINE: 108
```

After this command has executed, the first line of the set of moved lines contains the cursor and is displayed as the top line on the VDT screen (printed on a hard-copy device).

**Special Cases:**

If the line specified for the START LINE or END LINE prompt has been previously moved or deleted, an error message will be displayed. When this occurs, enter the XE or XES command and the same display that existed prior to entering the ML command will return.

If the INSERT AFTER LINE number specified is greater than the last line number existing in the file, E (ending line) is assumed, and the moved lines are inserted immediately following the last existing line in the file.

If an attempted ML operation is unsuccessful, divide the lines into several sets and repeat the operation for each set.

**Notes:**

Any modifications made to the lines prior to the move process are included at the new position of the lines.

**Related Commands:**

```
CL    (Copy Lines)
DL    (Delete Lines)
```

### 3.10 MR (MODIFY ROLL)

*Prompts:*

```
MODIFY RIGHT MARGIN
  RIGHT MARGIN POSITION: integer          (*)
```

*Purpose:*

The MR command specifies the number of lines the file being edited will roll up (toward the end of the file) or roll down (toward the beginning of the file) from the current line when a roll function (F1 key or F2 key) is activated. From 1 to 32,767 lines can be selected to be rolled up or down from the current line.

*Prompt Details:*

**NUMBER OF LINES TO ROLL:**

The integer that specifies the number of lines to roll when a roll function is activated. A single line or any number of lines up to and including 32,767 lines can be specified. (When entering an integer, do not include a comma.) A value greater than 32,767 causes the default of 12 lines to a roll.

*Example:*

The following example causes 22 lines of data to be rolled forward or rolled back when one of the roll functions executes.

```
MODIFY ROLL
  NUMBER OF LINES TO ROLL: 22
```

*Notes:*

When either the beginning of the file or the end-of-file mark is encountered during a roll operation, the roll terminates.

The MR command affects only the roll of files being edited by the Text Editor; it does not affect other roll functions.

*Related Commands:*

```
MRM    (Modify Right Margin)
MT     (Modify Tabs)
```

**3.11 MRM (MODIFY RIGHT MARGIN)***Prompts:*

```

MODIFY RIGHT MARGIN
  RIGHT MARGIN POSITION: integer (*)

```

*Purpose:*

The MRM command establishes the column that is to be used as the right margin. Data cannot be entered or modified to the right of the column specified.

*Prompt Details:***RIGHT MARGIN POSITION:**

The integer that specifies the column to be used as the right margin by the Text Editor. Any column from column 1 through column 240 can be specified. If the margin exceeds the LENGTH prompt specified in the XE command, the LENGTH specified is used as the right margin. If no right margin is specified, the default of 80 is assigned.

*Example:*

This example specifies column 132 as the rightmost column on a line at which data can be entered.

```

MODIFY RIGHT MARGIN
  RIGHT MARGIN POSITION: 132

```

*Special Cases:*

This command can be used in conjunction with the first tab stop set by the Modify Tabs command to establish the margins for centered text that is to be printed. For a standard pica printer, 10 characters equal one inch; therefore, the first tab stop set to 10 and the right margin set to 74 produces a six and one-half inch print area with one inch margins.

*Related Commands:*

```

MR      (Modify Roll)
MHR     (Modify Horizontal Roll)
MT      (Modify Tabs)

```

### 3.12 MT (MODIFY TABS)

*Prompts:*

```
MODIFY TABS
      TAB COLUMNS: [integer exp list]          (*)
```

*Purpose:*

The MT command specifies the columns in which the cursor is to be positioned when the tab function is called. The lowest number tab stop specified becomes the left margin for the cursor.

*Prompt Details:*

**TAB COLUMNS:**

A list of integers that specifies columns to be accessed when the tab function is executed. The tab stops are entered without intervening spaces, separated by commas. The column numbers can be entered in any order. When new values are entered, they become the initial values that are displayed. To clear all tabs, enter only the integer that will serve as the left margin for the cursor. If no tabs have been set, tabs 1, 8, 13, 26, and 31 are assigned by the XE and XES commands.

*Example:*

In the following example, columns 1, 7, 12, 17, 22, 27, 72, and 121 are set as tabs. The cursor returns to column 1 when moved to the beginning of a line and will reposition at column 7, 12, 17, 22, 27, 72, and 121 as each successive tab function is called.

```
MODIFY TABS
      TAB COLUMNS: 1,7,12,17,22,27,72,121
```

*Notes:*

There must be at least one tab stop set to serve as the left margin for the cursor (1 is the default). Tabs set past the right margin are ignored.

*Related Commands:*

```
MRM    (Modify Right Margin)
MR      (Modify Roll)
MHR     (Modify Horizontal Roll)
```

**3.13 QE (QUIT EDIT)***Prompts:*

```

QUIT EDIT
      ABORT?:  YES/NO                (NO)

```

If N or NO (NO is the initial value supplied) is entered in response to the ABORT? prompt, the following is displayed:

```

QUIT EDIT
  OUTPUT FILE ACCESS NAME: [site:]filename@
                        REPLACE?: YES/NO                (NO)
  MOD LIST ACCESS NAME:   [site:][pathname]@

```

*Purpose:*

The QE command terminates Text Editor processing. It allows the data generated during the editing session to be retained or discarded. If retained, the modified version of the input file is written to the output file. In addition, a listing of all the modifications made during the editing session can be printed to a file, to a device, or to the user's terminal.

*Prompt Details:*

**ABORT?:**  
Valid responses are Y, YES, N, or NO.

If the edit is to be discarded, an entry of Y or YES is required. The Text Editor is immediately terminated, and the work file created during the editing session is destroyed. No other prompts are displayed for this command.

If the data entered during the editing session is to be retained, a response of N or NO is required. The following prompts are then displayed.

**OUTPUT FILE ACCESS NAME:**  
The file name under which the results of the current editing session are to be stored. The default entry is the file name entered when the XE prompt was processed. If you enter a file name for a secure file, you must have delete access rights to the file in order to replace it with the edited file. If you do not, an error message appears on your screen, and you must create another file to accept the data entered in the current editing session.

**REPLACE?:**  
Respond with Y or YES to specify that an existing file is to receive the output of the Text Editor. The existing file can be either the input file or another file. Respond with an N or NO to indicate that the output file is to be automatically created by the Text Editor. An error message is returned if an N or NO is entered and the output file already exists. This feature prevents accidental replacement of existing file contents.

**MOD LIST ACCESS NAME:**

Name of the file or device that is to receive the listing of all the modifications made during the editing session. The device can be the user's terminal. If a file having this pathname exists, it is replaced automatically with the listing. If a pathname is specified that does not exist, the Text Editor automatically creates a sequential file (any required directories must have been created previously). If no access name is specified, no modifications listing is provided. The modification listing file always has a logical record length of 80, no matter how wide the file you are editing.

*Example:*

In the following example, the contents of the editing session are to be retained in an existing file having the file name .EDIT.SRC. The existing contents of .EDIT.SRC are to be replaced with the edit file and all modifications made during the edit session. The following prompt appears in response to the QE command.

```
QUIT EDIT
          ABORT?: NO
```

The default value, NO, is accepted to indicate that the contents of the editing session are to be saved. (A Y or YES entry would cause the results of the editing session to be discarded.) The following is displayed:

```
QUIT EDIT
  OUTPUT FILE ACCESS NAME: .EDIT.SRC
                   REPLACE?: YES
  MOD LIST ACCESS NAME: .MODLIST
```

These responses cause the results of the editing session to be retained in the file having the file name, .EDIT.SRC. The Y response to the REPLACE? prompt indicates that .EDIT.SRC is an existing file and its contents are to be replaced with the contents of the currently edited file. The response to the MOD LIST ACCESS NAME prompt indicates that the list of all modifications made to the file, .EDIT.SRC, are to be written to the file, .MODLIST.

**Special Cases:**

The output file has the following characteristics:

- If no input file was specified and an output file does not exist, the output file is created as a sequential file.
- If the output file is to replace the input file, the characteristics of the input file are used.
- If the input file exists and the output file does not (the output file is not to replace the input file), the characteristics of the input file are used.
- If the output file exists, its characteristics are used.
- If the output file is to be a relative record file, it must first be created using the Create File (CF) command. Its assigned pathname must then be specified as the Text Editor output file.

**Notes:**

If you use the QE command in a secure system, you must have delete access rights to any existing file you want to replace (the OUTPUT FILE ACCESS NAME). If you specify a new file that must be created, you are assigned all access rights. You must have write access rights to the file you name in response to the MOD LIST ACCESS NAME prompt. If you do not have the appropriate access rights, see your security manager.

The following form of the QE command can be entered to *abort* the editing session and terminate Text Editor processing:

```
[ ] QE A
```

No further field prompts are displayed.

The following form of the QE command can be entered to specify that the currently edited file is to *replace* the input file:

```
[ ] QE R
```

No further field prompts are displayed and Text Editor processing is terminated.

Key indexed files cannot be edited by the Text Editor.

Relative record files cannot be created automatically by the Text Editor.

An example of the modification listing is presented in Section 4 (for a VDT) and Section 5 (for a hard-copy terminal).

*Related Commands:*

RE	(Recover Edit)
XE	(Execute Text Editor)
XES	(Execute Text Editor with Scaling)

**3.14 RE (RECOVER EDIT)***Prompts:*

```

RECOVER EDIT
  OUTPUT FILE ACCESS NAME: [site:]filename@
                        REPLACE?: YES/NO          (NO)
  MOD LIST ACCESS NAME: [site:][pathname]@

```

*Purpose:*

The RE command allows you to recover a major portion of a text edit that was in progress when a system crash or power failure occurred. The amount of data that can be recovered depends on the nature of the system failure. The command is entered after the system has been initialized and must be entered from the terminal at which the edit was originally in progress.

*Prompt Details:***OUTPUT FILE ACCESS NAME:**

The file name in which the results of the current editing session are to be stored. Specify the name for a file that the Text Editor is to automatically create or an existing file name that is not the original input file name. If you specify an existing file name in a secure system, you must have delete access rights to that file or you receive an error message. It is recommended that you do not replace the original input file since the Text Editor work files can contain some inaccurate or missing data, depending on the nature of the system failure. If the file name uses a synonym assigned during the most recent log-on, that synonym may have been lost during the system failure.

**REPLACE?:**

Valid responses are Y, YES, N, or NO. This prompt allows you to specify whether the contents of the output file are to be replaced. If an existing file is to receive the output of the Text Editor, the response should be Y. An N or NO response indicates the output file is to be created. An error message is returned if an N or NO is entered and the output file already exists. This prevents the accidental replacement of the contents of an existing file.

**MOD LIST ACCESS NAME:**

The pathname of the file or device that is to receive the listing of all the modifications made during the editing session. The device can be the user's terminal. If a pathname is specified and a file does not exist with that name, the Text Editor will automatically create a sequential file (any required directories must have been previously created). If no access name is specified, no modifications listing is provided.

*Example:*

After the system has been initialized following a system crash or a power failure, enter the RE command from the terminal at which the edit was originally in progress. In the following example, the Text Editor is to automatically create a sequential file named .DIR1.FILE2 (the directory .DIR1 has been created previously by the user) and write the contents of the editing session to that file. The modifications list is to be printed to a file, .DIR1.FILE3, that the Text Editor is to automatically create. Thus, the user can determine which of the modifications made during the editing session were lost when the system crashed.

```
RECOVER EDIT
  OUTPUT FILE ACCESS NAME: .DIR1.FILE2
                        REPLACE?: NO
  MOD LIST ACCESS NAME: .DIR1.FILE3
```

*Assumptions:*

The system has been initialized.

The Recover Edit (RE) command is entered from the terminal at which the edit was in progress prior to the system crash.

The Text Editor has not been activated prior to your entering the RE command.

*Notes:*

If you use the RE command on a system with file security, you must have delete access rights to replace the file named in response to the OUTPUT FILE ACCESS NAME prompt and also write access to the file named in response to the MOD LIST ACCESS NAME prompt. If you do not have the appropriate access rights, see your security manager.

Since the Text Editor buffers data before writing it to the disk, the most recent changes made during the editing session will probably not be recoverable. This usually involves no more than the changes made on the last screen displayed.

Following a system crash or power failure, the RE (Recover Edit) command must be entered before starting a new editing session, if the previous editing session is to be saved. When you initiate the Text Editor following the system crash, the Text Editor deletes the existing work files from the edit session that was in progress when the crash occurred. Recovery of the previous editing session is then no longer possible.

An error message is returned if the RE command is entered while an editing session is in progress. Also, if the RE command is entered and there was no edit in progress prior to the system crash, an error message will be returned.

Depending on the nature of the system crash (for example, disk parity errors), the Text Editor work files can be affected and recovery may not be possible.

**3.15 RS (REPLACE STRING)***Prompts:*

```

REPLACE STRING
  NUMBER OF OCCURRENCES: integer          (1)
    START COLUMN: integer                 (*)
    END COLUMN: integer                   (*)
      STRING: [character(s)]              (*)
      CHANGE: [character(s)]

```

*Purpose:*

The RS command replaces a specified number of occurrences of a string with a replacement string. The command causes a search, beginning with the first character of the current line, of each line of the file for the specified string. Each time the string is found, it is replaced with the replacement string until the specified number of occurrences of the string are replaced. A null value can be specified for the STRING prompt. If a null value is specified, the RS command deletes all characters between the specified columns, regardless of the characters in those columns. The RS command performs the deletion for the specified number of lines (occurrences) and inserts the specified replacement string in each line, beginning at the specified start column.

*Prompt Details:***NUMBER OF OCCURRENCES:**

The integer that specifies the number of occurrences of the specified string to be replaced.

**START COLUMN:**

The integer that specifies the number of the column on each line where the search for the specified string will begin.

**END COLUMN:**

The integer that specifies the last column of each line to be searched for the specified string.

**STRING:**

The character string that is to be replaced. There is no default value; however, the previously used value may be displayed in this field. Leading and trailing blanks are not recognized as part of the string unless the string and all leading and trailing blanks are enclosed in a set of double quotation (“ ”) marks. When you have set, OPTION LOWERCASE = YES a string that includes lowercase characters must also be enclosed in a set of double quotation marks. If it is not, all lowercase characters are translated to uppercase.

**CHANGE:**

A string that is to replace the previously specified string when it is found. If a null response is entered, the specified string is deleted. If the string includes leading or trailing blanks or lowercase characters, it must be encased in a set of double quotation marks.

*Example:*

In the following example, columns 15 through 70 are searched for the string, *next week*. The search begins with the current line and continues until four occurrences of the string have been replaced with the string, *tomorrow*.

```
REPLACE STRING
  NUMBER OF OCCURRENCES: 4
    START COLUMN: 15
    END COLUMN: 70
    STRING: "next week"
    CHANGE: "tomorrow"
```

After the fourth replacement of the string has occurred, the file is positioned so that the last line in which a replacement occurred is displayed on the line on which the cursor is positioned (printed on a hard-copy device). If less than four occurrences were found and replaced, the end-of-file mark is displayed.

*Notes:*

When selecting a character string that is to be replaced, choose a unique set of characters to avoid modifying other data that contains the same character string. Leading and/or trailing blanks are useful in this respect.

If the character string contains two or more separate words, only those occurrences in which the entire string occurs on one line are recognized and replaced. You must scan the margin of your file to locate any occurrences in which parts of the string have been typed on two separate lines.

This command only searches forward in the file. To replace a string in the entire file, use the Show Line (SL) command to return to the beginning of the file, then perform the Replace String (RS) operation.

If the replacement string has fewer characters than the replaced string, characters to the right of the string column are shifted to the left, and the line is filled with trailing blanks. If the replacement string is longer than the replaced string, the characters to the right of the replacement string (and to the left of the right margin) are shifted right. This can cause some of the characters to be lost if the line is already filled with characters.

*Related Commands:*

DS	(Delete String)
FS	(Find String)

### 3.16 SL (SHOW LINE)

*Prompts:*

```
SHOW LINE  
LINE: [B/BEGINNING/E/END/line number] (1)
```

*Purpose:*

The SL command repositions the input file to display a designated line within the file.

*Prompt Details:*

**LINE:**  
The integer or expression that specifies the number of the line that is to be displayed. The beginning of the file can be specified by entering a B; the end-of-file mark can be specified by entering an E in response to this prompt.

*Example:*

The following example causes the current file to be positioned with line 33 displayed at the top of the VDT screen (or to be the next line printed on the hard-copy device).

```
SHOW LINE  
LINE: 33
```

*Note:*

If the line specified has been previously deleted, the next numbered line is displayed.

### 3.17 SVL (SAVE LINES)

*Prompts:*

```
SAVE LINES
          START LINE: [BEGINNING/END/line number]
          END LINE:   [BEGINNING/END/line number]
          SAVE FILE PATHNAME: [site:]pathname@
OPTION(ADD/REPLACE/EXTEND): {ADD/REPLACE/EXTEND}          (ADD)
```

*Purpose:*

The SVL command copies a designated set of contiguous lines in the current file to another file. The original lines remain intact in their original position in the current file.

*Prompt Details:*

**START LINE:**

The integer or expression that specifies the first line of the set of lines to be copied to another file. The character B or E can be entered to specify the beginning or ending line, respectively, of the file. If no response is entered, the line on which the cursor is positioned (whether numbered or unnumbered) is assumed to be the first line of the set to be copied.

**END LINE:**

The integer or expression that specifies the last line of the set of lines to be copied to another file. The character B or E can be entered to specify the beginning or ending line, respectively, of the file. This value should be the same as or greater than the START LINE value. If the value is less than the START LINE value, the START LINE value is used. If no response is entered, the line on which the cursor is positioned, whether numbered or unnumbered, is assumed to be the last line of the set to be copied.

**SAVE FILE PATHNAME:**

The pathname of the relative record or sequential file to which the set of lines is to be copied. The Text Editor automatically creates a sequential file with this file name if the pathname specified does not exist. (Any required directories must have been created previously.)

**OPTION**

Valid responses are A, ADD, R, REPLACE, or E, EXTEND. The A or ADD response prohibits replacement of an existing file and returns an error message, if a file already exists with the pathname specified in response to the SAVE FILE PATHNAME prompt. The R or REPLACE response allows replacement of the contents of an existing file with the lines specified by the SVL command. The E or EXTEND response extends an existing file if one already exists with the pathname specified, and if not, this response creates a new file.

**Example:**

In the following example, a file named .EDIT.SAVE is created automatically by the Text Editor (the directory .EDIT has been created previously using the Create Directory File (CFDIR) SCI command). Lines 33 through 56 of the file being edited are copied into this new file.

```

SAVE LINES
                START LINE: 33
                END LINE: 56
                SAVE FILE PATHNAME: .EDIT.SAVE
OPTION(ADD/REPLACE/EXTEND): ADD

```

Notice that the response to the OPTION prompt is ADD. The response to this prompt could have been REPLACE or EXTEND and the operation would still have executed properly, since the file does not exist. The file is to be created automatically by the Text Editor as an empty file to be “replaced” by the set of lines specified. However, to avoid the possibility of ever writing over an existing file by mistake, always respond ADD to an OPTION prompt unless you intend to replace or extend an existing file.

**Special Cases:**

To save (copy) an entire file, enter B as the START LINE response and E as the END LINE response. The entire file is saved (copied) into the file specified.

If either of the line numbers specified has been previously moved or deleted, an error message is displayed. Enter an XE or XES command to return to the display that existed when the command was entered. If the END LINE value specified is greater than the last existing line number in the file, E (ending line) is assumed.

Any modifications made to the lines prior to the save (copy) process are included when the lines are copied.

If the saved lines record length is shorter than that of the receiving file, the records are padded with blanks to match the receiving file. If the saved lines record length is longer than that of the receiving file, the records are truncated. Any data occurring beyond the record length of the receiving file will be lost.

**Notes:**

To use the SVL command on a system with file security, you must have write access to the file named in response to the SAVE FILE PATHNAME prompt. If you do not have the appropriate access rights, see your security manager.

**Related Commands:**

CL	(Copy Lines)
DL	(Delete Lines)
IF	(Insert File)
ML	(Move Lines)

### 3.18 XE (EXECUTE TEXT EDITOR)

*Prompts:*

```
EXECUTE TEXT EDITOR
FILE ACCESS NAME: [site:][filename]@      (*)
EXCLUSIVE EDIT?: YES/NO                    (YES)
LINE LENGTH: integer                       (80)
```

*Purpose:*

The XE command activates the Text Editor. The prompts establish initial characteristics of the file being edited.

*Prompt Details:*

**FILE ACCESS NAME:**

Specify a null response or the file name of an existing sequential or relative record file (that is, a file that has been created, which may or may not contain data). To enter a null response, press the Skip key. (On a hard-copy terminal, enter two quotation marks (" ") and press the Return key.)

If you specify a file in a secure system, you must have write access rights to it. If you do not, an error message is displayed on the screen.

If no file name is entered, the Text Editor creates a work file that is either discarded, written to an existing file, or written to a file created by the Text Editor when the editing session is terminated with the Quit Edit (QE) command. The file name of the most recently edited file may be displayed as an initial value in this field.

**EXCLUSIVE EDIT:**

Specify YES (or accept the default condition) to the exclusive edit prompt if you desire exclusive write access privileges to the file specified. If you specify NO, other users share access to the file specified. This prompt does not appear if you reenter an edit already in progress.

**LINE LENGTH**

Specify an even integer between 80 and 240 for this prompt. This number is the maximum line length of the file specified. This value defaults to 80 columns if no other value is specified. This prompt does not appear if you reenter an edit already in progress.

*Example:*

In the following example, the file named .EDIT.XE.SRC, which has a record length of 120 characters, is entered as the input file. The contents of this file are displayed for exclusive editing.

```
EXECUTE TEXT EDITOR
      FILE ACCESS NAME: .EDIT.XE.SRC
      EXCLUSIVE EDIT?: YES
      LINE LENGTH: 120
```

*Special Cases:*

An error message is displayed if an attempt is made to execute the Text Editor while an editing session is already in progress that is using a different input file.

When returning to an edit that is in progress, you can enter the command name followed by a period (XE.), which indicates that the current XE prompt response is accepted and the XE field prompt need not be presented.

The value specified in response to the EXCLUSIVE EDIT? and LINE LENGTH prompts is set for the current editing session and cannot be changed for the specified file. At a later editing session (following the termination of the current session) you can specify different values for these prompts.

**NOTE**

Specifying a line length shorter than the file to be edited results in records being truncated. This could result in the loss of data.

*Notes:*

To use the XE or XES command on a file in a secure system, you must have read access rights for the file listed in response to the FILE ACCESS NAME prompt. You must also have the appropriate access to execute the QE command when you finish your edit (delete access rights to replace the file you are editing). If you do not have the access rights you need, see your security manager.

## *Text Editor Commands*

All SCI commands can be executed during the editing session by entering command mode, which suspends the Text Editor. Text editing can then be resumed by entering the XE (or XES) command when the SCI operation is complete.

It is not necessary to terminate the Text Editor to change to the Execute Editor with Scaling (XES) command. Simply enter the XES command and the display that existed prior to entering the XES command will exhibit the scale on the bottom line of the VDT.

### *Related Commands:*

RE	(Recover Edit)
QE	(Quit Edit)
XES	(Execute Text Editor with Scaling)

**3.19 XES (EXECUTE TEXT EDITOR WITH SCALING)***Prompts:*

```

EXECUTE TEXT EDITOR WITH SCALING
FILE ACCESS NAME: [site:][filename@]      (*)
EXCLUSIVE EDIT?: YES/NO                  (YES)
LINE LENGTH: integer                      (80)

```

*Purpose:*

The XES command activates the Text Editor at a video display terminal that is in VDT mode and displays a calibrated scale indicating column numbers, on line 24 of the screen display. This allows easy horizontal positioning of the cursor to the correct column. On a hard-copy terminal, the scaling line is displayed prior to the display of the first line to be edited.

*Prompt Details:***FILE ACCESS NAME:**

Specify a null response or the file name of an existing sequential or relative record file (that is, a file that has been created, which may or may not contain data). To enter a null response, press the Skip key. (On a hard-copy terminal, enter two quotation marks (" ") and press the Return key.)

If you specify a file in a secure system, you must have write access rights to the file. If you do not, an error message is displayed on the screen.

If no file name is entered, the Text Editor creates a work file that is either discarded, written to an existing file, or written to a file created by the Text Editor when the editing session is terminated with the Quit Edit (QE) command. The file name of the most recently edited file can be presented as an initial value in this field.

### EXCLUSIVE EDIT

Specify YES (or accept the default condition) to the exclusive edit prompt if you desire exclusive write access privileges to the file specified. If you specify NO, other users share access to the file specified. This prompt does not appear if you reenter an edit already in progress.

### LINE LENGTH

Specify an even integer between 80 and 240 for this prompt. This number is the maximum line length of the file specified. The default value is 80 column records unless another value is specified. This prompt does not appear if you reenter an edit already in progress.

#### *Example:*

In the following example, the file named .EDIT.XES.SRC, with a record length of 240 lines, is entered as the input file. The contents of this file are presented for non-exclusive editing and line 24 displays the column scale.

```
EXECUTE TEXT EDITOR WITH SCALING
FILE ACCESS NAME: .EDIT.XES.SRC
EXCLUSIVE EDIT?: NO
LINE LENGTH: 240
```

#### *Special Cases:*

An error message is presented if an attempt is made to execute the Text Editor with Scaling while an editing session is already in progress with a different input file specified.

When returning to an edit that is in progress, you can enter the command name followed by a period (XES.), which indicates that the current XES prompt response is accepted and the XES field prompt need not be presented.

The value specified in response to the EXCLUSIVE EDIT? and LINE LENGTH prompts is set for the current editing session and cannot be changed for the specified file. At a later editing session (following the termination of the current session) you can specify different values for these prompts.

### NOTE

Specifying a line length shorter than the file to be edited results in records being truncated. This could result in the loss of data.

**Notes:**

All SCI commands can be executed during the editing session by entering command mode, which suspends the Text Editor. Text editing can then be resumed by entering the XES (or XE) command when the SCI operation is complete.

It is not necessary to terminate the Text Editor to change to the XE (without scaling) command. Simply enter the XE command and the display that existed prior to entering the XE command will appear without the scale on line 24.

To use the XE or XES command on a file in a secure system, you must have read access rights for the file listed in response to the FILE ACCESS NAME prompt. You must also have the appropriate access to execute the QE command when you finish your edit (delete access rights to replace the file you are editing). If you do not have the access rights you need, see your security manager.

**Related Commands:**

RE	(Recover Edit)
QE	(Quit Edit)
XE	(Execute Text Editor)



# Exercises Using a Video Display Terminal

---

## 4.1 INTRODUCTION

This section contains Text Editor exercises using a video display terminal (VDT) in VDT mode. Step-by-step instructions are provided to familiarize you with each of the actions required to use the Text Editor effectively. For this reason, you should complete this section while at the terminal.

Before beginning these exercises, check with the person in charge of your system to determine if a log-on procedure is required. If so, you will need a user identification (ID). Also, confirm that the file names .EXAMPLE and .LISTA are available for use. If not, obtain other file names to replace them (in all occurrences) for the following exercises.

## 4.2 ACTIVATING THE SYSTEM COMMAND INTERPRETER

To activate SCI at a terminal, use the following procedure.

1. Turn on the terminal.
2. Press the Attention key.
3. Press the exclamation point (!) key. (This is a shifted key.) If your terminal does not require a log-on procedure, the SCI main menu of available command groups is displayed, and the terminal is ready for command input.

The following is a typical log-on procedure required on some systems:

4. If your system requires identification from the user, DNOS displays the following:

```
DNOS x . x . x x
```

```
USER ID:  
PASSCODE:  
JOB NAME:
```

5. Enter your assigned user ID and press the Return key. DNOS responds by positioning the cursor beside the PASSCODE prompt.
6. Enter your assigned passcode and press the Return key. To preserve passcode security, the characters of the passcode are not displayed as they are entered.
7. Enter TEST following the JOB NAME prompt and press the Return key.

## NOTE

Depending on the options specified by your particular system, a news file can appear. Press the Command key to obtain the SCI main menu. If additional log-on prompts are displayed, refer to the *DNOS System Command Interpreter (SCI) Reference Manual* for further information regarding the log-on procedure.

8. SCI is activated at the terminal, and the SCI main menu of available command groups is displayed.

The terminal must be in VDT mode. If your terminal is in TTY (teletypewriter) mode, the prompts will be successively displayed on the bottom line of your screen. To change the mode of operation to VDT mode, use the Modify Terminal Status (MTS) SCI command. Then enter the Quit (Q) SCI command and again perform the steps to activate the SCI. For further information concerning terminal mode, refer to the *DNOS System Command Interpreter (SCI) Reference Manual*.

The following exercises assume that you have specified `.OPTION LOWERCASE = YES`. You can respond to all command prompts except Delete String (DS), Find String (FS), and Replace String (RS) in either uppercase or lowercase characters.

### 4.3 EXERCISE 1 — TEXT EDITING SESSION

A typical text editing session includes the following activities:

- Activating the Text Editor
- Establishing a format
- Entering data
- Terminating the Text Editor
- Printing the file

#### 4.3.1 Activating the Text Editor

Do the following to activate the Text Editor.

1. Press the Command key, enter the Execute Text Editor (XE) command by typing the characters XE, and press the Return key. The following display appears:

```
EXECUTE TEXT EDITOR
      FILE ACCESS NAME:
      EXCLUSIVE EDIT?:  YES
      LINE LENGTH:     80
```

2. If the response field for the FILE ACCESS NAME prompt is blank, press the Return key. However, if a value is displayed, press the Skip key to clear the field. Since you have not entered any characters in the prompt response field, you have given a null response. In this case, the response indicates that the Text Editor is to create a new file instead of editing an existing one. When you terminate the editing session, you are required to assign a file name to this file if you want to retain it as an output file.
3. After you have responded to the FILE ACCESS NAME prompt, the cursor moves down to the EXCLUSIVE EDIT? prompt and then the LINE LENGTH prompt. For purposes of this exercise, accept the initial values for these prompts by pressing the Return key two times. The file you create has the attribute of exclusive write and a line length of 80 characters. After a file has been created, these characteristics cannot be changed in this edit session.
4. After you have responded to these prompts, the following display appears at the top of the screen:

\*EOF

The \*EOF indicates that the only record in the file is the end-of-file (\*EOF) mark.

#### 4.3.2 Establishing a Format

When you print the text of a file, it is similar in appearance to the text on the video display screen. In this exercise, you are preparing a memorandum that is to be centered on the printed page. Therefore, your next step is to set the margins and tabs using the Modify Tabs (MT) and Modify Right Margin (MRM) commands. These commands can be used either during or before the editing session.

The left margin for the cursor and the tabs are set by the Modify Tabs (MT) SCI command. There are 80 columns (spaces) available for data characters on each line (record). To indent the memorandum one inch from each edge of the printout, the margins must be set at 10 and 74. This leaves a print area of six and a half inches, or 65 characters, on a standard pica (10 characters per inch) printer. Use the following procedure to set the tab stops and the right margin.

1. Press the Command key to enter command mode. The cursor appears at the lower left corner of your screen.
2. To execute the Modify Tabs (MT) command, type the characters MT, and press the Return key.
3. The TAB COLUMNS prompt appears displaying initial values. Clear the initial values by pressing the Erase Field key. (Alternately, you can simply type over existing values.) Set the tab stops by typing the numbers, separated by commas, as they appear below:

MODIFY TABS

TAB COLUMNS: 10,15,20,25,30,35,40,45,50,55,60,65,70

4. Now press the Return key. This establishes the first tab stop at column 10 and additional tabs at five-space intervals. Although you do not need all of these tab stops to type the exercise text, the regularly spaced tabs permit you to use the Forward Tab key to move the cursor quickly across a line. This also sets the left margin since the cursor moves to the lowest numbered tab setting when the Return key is pressed.
5. To execute the Modify Right Margin (MRM) command, press the Command key. Type the characters MRM, then press the Return key to activate the Modify Right Margin command. Enter the number 74 in response to the prompt as follows:

```
MODIFY RIGHT MARGIN
RIGHT MARGIN POSITION: 74
```

6. Press the Return key. The right margin is now set at 74 and the display shows the \*EOF mark.

### 4.3.3 Entering Data

Begin entering the text of the example memorandum shown in Figure 4-1 by following the next few steps; then complete the memorandum using the text shown in the figure.

1. The Text Editor has been activated in compose mode since you did not specify an existing input file. When this function is active, a new blank-filled line is generated each time you press the Return key. Consequently, new lines can be added easily as the file is built. (You can use the F7 key to alternate between compose and edit modes as appropriate.)
2. Press the Return key. The cursor is in column 10 (the first tab stop) of the newly generated line, and the \*EOF marker is in the next line. Press the Return key again to create a blank line at the top of the screen.
3. To center the word *MEMORANDUM*, use the Forward Tab key to position the cursor on column 30.
4. Place your keyboard in uppercase lock mode to cause all of the alphabetic keys to print in uppercase. This key does not affect the operation of the other keys. Type the word *MEMORANDUM*, leaving one space between each of the letters. Press the Return key three times to create two blank lines.
5. Enter the word *DATE:*. Press the Forward Tab key once. Release the Uppercase Lock key and type the date *June 15, 1981*. Press the Return key two times.
6. Type the word *TO:*, using either the Shift key or the Uppercase Lock key. Forward tab until the cursor is aligned with the word *June*. Continue typing the memorandum.
7. To correct an error, simply backspace, using the Previous Character key, and reenter the word. Experiment using the arrow keys to move the cursor. The arrow keys do not affect the displayed data as the cursor moves, whereas the space bar replaces the characters it passes with blanks.

M E M O R A N D U M

DATE: June 15, 1981  
TO: Tom Smith  
COPY: Jane Doe  
FROM: John Q. Public  
SUBJECT: Contract commitment

All work on the XYZ project will be completed by July 3, 1981. This corresponds to the original commitment in the contract, and this commitment will be met.

Regards,

John Q. Public  
Public Corporation

Figure 4-1. Input for Memorandum Using a VDT

#### 4.3.4 Terminating the Text Editor

After you enter the data, terminate the Text Editor by using the Quit Edit (QE) command, as follows:

1. Press the Command key to enter command mode, enter the letters QE, then press the Return key. The following prompt appears:

```
QUIT EDIT
          ABORT?: NO
```

2. The ABORT? prompt allows you to retain the data entered during the editing session (by entering N or NO) or to discard the data entered and abort the editing session (by entering Y or YES). Since the data is to be retained, press the Return key to accept the initial value supplied (NO). The following display appears:

```
QUIT EDIT
  OUTPUT FILE ACCESS NAME:
                    REPLACE?: NO
  MOD LIST ACCESS NAME:
```

3. The file you have created must be assigned a file name so that it can be stored and recalled. For this exercise, use the file name .EXAMPLE for the output file, as follows:

```
QUIT EDIT
  OUTPUT FILE ACCESS NAME: .EXAMPLE
                    REPLACE?: NO
  MOD LIST ACCESS NAME:
```

4. Since a new file is being created (instead of an existing file being replaced), accept the initial value (NO) supplied after the REPLACE? prompt by pressing the Return key. This response prevents the accidental destruction of an existing file having the same name.
5. Enter a null response to the MOD LIST ACCESS NAME prompt by pressing the Return key. (The use of the modification listing is described in a subsequent paragraph.) The Text Editor output file is then assigned the file name you have given it and is stored. The Text Editor is no longer active and the SCI main menu is displayed.

#### 4.3.5 Printing the File

To print the file you have created, first determine the access name of a printer and confirm that the printer is online and ready to print. Then, proceed as follows:

1. Press the Command key, type PF, then press the Return key to activate the Print File SCI command. The following is displayed:

```
PRINT FILE
      FILE PATHNAME(S):
          ANSI FORMAT?: NO
LISTING DEVICE OR CLASS:
DELETE AFTER PRINTING?: NO
  NUMBER OF LINES/PAGE: 62
                        FORM: STANDARD
  NUMBER OF COPIES: 1
      BANNER SHEET?: NO
```

2. Enter .EXAMPLE in response to the FILE PATHNAME(S) field prompt.
3. Accept the initial value in response to the ANSI FORMAT? prompt by pressing the Return key.
4. In response to the LISTING DEVICE prompt, enter the name of the printer, for example, LP01 (the last two digits are numerals). Then press the Return key.
5. Accept the initial values and default values for each of the remaining prompts by pressing the Enter key. A detailed description of the Print File (PF) field prompts is contained in the *DNOS System Command Interpreter (SCI) Reference Manual*.
6. Remove the printed memorandum from the printer.

#### 4.4 EXERCISE 2 — REVISING AN EXISTING FILE

The following exercise provides the general procedure for editing an existing file. The input file used in this text editing session is the one you created previously. Proceed as follows:

1. Activate the Text Editor by pressing the Command key, typing XES for the Execute Text Editor with Scaling (XES) command and then pressing the Return key. The following is displayed:

```
EXECUTE TEXT EDITOR WITH SCALING
FILE ACCESS NAME: .EXAMPLE
EXCLUSIVE EDIT?: YES
LINE LENGTH: 80
```

The file name .EXAMPLE appears following the FILE ACCESS NAME prompt because the Text Editor uses the output file name from your most recent text editing session as the initial value for the input file name (if no initial value is displayed, type .EXAMPLE).

2. Since this is the file you are going to revise, press the Enter key. The first 23 lines from the input file, .EXAMPLE, are displayed as shown in Figure 4-2. Each line is sequentially numbered, the Text Editor is in edit mode, and a scale indicating column numbers is at the bottom of the display screen.

```
1
2
3
4
5     DATE:      June 15, 1981
6
7     TO:        Tom Smith
8
9     COPY:      Jane Doe
10
11    FROM:      John Q. Public
12
13    SUBJECT:   Contract commitment
14
15    All work on the XYZ project will be completed by July 3,
16    1981. This corresponds to the original commitment in
17    the contract, and this commitment will be met.
18
19    Regards,
20
21
22    John Q. Public
23    Public Corporation
.....010.....020.....030.....040.....050.....060....:
```

Figure 4-2. Display of the Input File on a VDT

- Use the edit control keys and commands given in the next steps to revise the memo as shown. The changes to be made are underlined in Figure 4-3.

M E M O R A N D U M

DATE: June 15, 1981  
TO: Tom Smith  
COPY: Jane Doe  
Travis Williamson  
FROM: John Q. Public, Project Manager  
SUBJECT: Contract deadline

All work on the XYZ project **MUST** be completed by June 27, 1981. This corresponds to the amended deadline in the contract and this deadline MUST be met.

I repeat:

All work on the XYZ project MUST be completed by June 27, 1981. This corresponds to the amended deadline in the contract, and this deadline MUST be met.

Regards,

John Q. Public, Project Manager

Figure 4-3. Revision to Memorandum Using a VDT

- Use the Replace String (RS) command to make the following revision. Since you selected .OPTION LOWERCASE = YES for your system, you must enclose the string to be replaced and the change within double quotation marks (" "). Add the title, *Project Manager*, following John Q. Public's name at lines 11 and 22 by first positioning the cursor on any line preceding or including the line numbered 11. Press the Command key, type RS, then press the Return key, activating the Replace String command. The Text Editor responds by displaying:

```
REPLACE STRING
  NUMBER OF OCCURRENCES: 1
    START COLUMN: 1
      END COLUMN: 80
        STRING:
```

5. Since you only want to replace the first two occurrences of the word *Public* in the file, specify that the search is to include every column position and specify the number of occurrences to be replaced as 2. In response to the NUMBER OF OCCURRENCES prompt, replace the initial value by typing a 2 and pressing the Return key.
6. Type 10 following the START COLUMN prompt and press the Return key.
7. Type 74 following the END COLUMN prompt and press the Return key.
8. Since the word *Public* is a unique string of characters in this document, this one word can be used as the string to be replaced. Type the word "Public" following the STRING prompt.
9. Type the words "Public, Project Manager" in response to the CHANGE prompt. The display appears as follows:

```

REPLACE STRING
  NUMBER OF OCCURRENCES: 2
      START COLUMN: 10
      END COLUMN: 74
      STRING: 'Public'
      CHANGE: 'Public, Project Manager'

```

10. Press the Return key. The display returns with the cursor positioned on the line on which the last change was made.
11. Next, insert a line after the name *Jane Doe* by positioning the cursor on line 10, using the Previous Line key. Press the Initialize Input key to activate the insert line function. An unnumbered line is inserted immediately above line 10. Press the Forward Tab key combination to position the cursor in the column under the J in *Jane Doe*, and type the name *Travis Williamson*.
12. Replace the word *commitment* with the word *deadline* throughout the document by using the Replace String (RS) command. (If you were revising a larger file, you would use the Show Line (SL) command to return to the beginning of the file before executing the Replace String command.) Leave the cursor where it is located, press the Command key, type RS, then press the Return key. Since every occurrence of the word is to be replaced, enter a number large enough to accomplish this. Type 50 in response to the NUMBER OF OCCURRENCES prompt. Respond to the prompts as follows:

```

REPLACE STRING
  NUMBER OF OCCURRENCES: 50
      START COLUMN: 10
      END COLUMN: 74
      STRING: 'commitment'
      CHANGE: 'deadline'

```

The display returns with the cursor positioned at the end-of-file (\*EOF) mark.

13. Return to the beginning of the file by pressing the F2 key (Roll Down function) or by using the Show Line (SL) command. To use the SL command, press the Command key and accept the initial value (1) supplied for the LINE prompt by pressing the Return key. The display returns with the cursor positioned in line 1. This command is very useful when you want to locate a particular line number in a large file.
14. To change the word *will* to *MUST*, use the Next Line key to position the cursor on line 15. Use the Forward Tab key combination to move the cursor to the column where the change is to be made, then simply type *MUST* over the word *will*. In the same manner, change the date on the same line to *June 27*.
15. Use the Next Line key to move the cursor to line 16. Use the Previous Character key to position the cursor over the first letter of the word *original*. Type the word *amended* and delete the extra character using the Delete Character key.
16. Use the Next Line key to move the cursor to line 17 and change the word *will* to *MUST*.
17. The next change to be made is the repetition of the paragraph. First, note the line numbers of the lines to be copied. Then, press the Command key and use the Copy Lines (CL) command to copy lines 15 through 17. Place them following line 17. Enter your responses to the CL prompts as follows:

```
COPY LINES
      START LINE: 15
      END LINE: 17
INSERT AFTER LINE: 17
```

18. To insert the line, *I repeat:*, use the Previous Line key to position the cursor on the first line of the newly copied paragraph. Press the Initialize Input key twice to insert two new blank lines. Position the cursor on the line preceding the second paragraph and type the words, *I repeat:*.
19. For the final revision, use the Delete Lines (DL) command. Press the Command key to enter command mode, type DL, then press the Return key. The following display appears:

```
DELETE LINES
      START LINE:
      END LINE:
```

20. Since you are deleting only line 23, type 23 in response to the START LINE prompt and type 23 in response to the END LINE prompt. Press the Return key.
21. When the revisions to the file are complete, use the Quit Edit (QE) command to terminate the editing session. First, press the Command key. Then enter QE and press the Return key. The following appears:

```
QUIT EDIT
      ABORT?: NO
```

22. Since you want to retain the revised version in the output file, respond to the ABORT? prompt by pressing the Return key to accept the initial value (NO). The following is displayed:

```

QUIT EDIT
  OUTPUT FILE ACCESS NAME:  .EXAMPLE
                        REPLACE?:  NO
  MOD LIST ACCESS NAME:

```

Notice that the value entered earlier, in response to the FILE ACCESS NAME prompt for the XES command, appears after FILE ACCESS NAME. Press the Return key to accept it as the output file name.

23. Type Y or YES in response to the REPLACE? prompt. This indicates that the output file is to replace the input file.
24. In response to the MOD LIST ACCESS NAME prompt, type .LISTA (the listing produced is explained in the next paragraph). Now, press the Return key. Once this operation is complete, control again returns to the SCI and the SCI main menu is displayed.

#### 4.4.1 Using the Modification Listing File

The Quit Edit command prompt, MOD LIST ACCESS NAME, allows you to specify a file to which a list of file modifications will be written. This listing only contains the lines that were changed in the recent editing session and thus provides an efficient method of confirming the changes. The file containing the list of all modifications made during the text editing session for the revision exercise is assigned the access name .LISTA and appears as shown in Figure 4-4. Use the SCI command Show File (SF) to examine the contents of .LISTA. The flags in the left margin have the following meanings:

Flag	Meaning
O	Old value
N	New value
I	Inserted line
D	Deleted line

Note that in the modification listing shown in Figure 4-4 the input file name and the output file name are both listed. Modified lines are indicated by flagging the old value with an O and the new value with an N. Line numbers are indicated in the second column. The line number given for an inserted line, indicated with an I, is the line number of the line after which the insertion was made. The deleted line is indicated with a D, followed by the line number of the line that was deleted.

TEXT EDITOR MODIFICATION LISTING

INPUT ACCESS NAME: .EXAMPLE  
OUTPUT ACCESS NAME: .EXAMPLE  
MOD LIST ACCESS NAME: .LISTA

```
I 9*          Travis Williamson
O 11      FROM:   John Q. Public
N 11      FROM:   John Q. Public, Project Manager
O 13      SUBJECT: Contract commitment
N 13      SUBJECT: Contract deadline
O 15      All work on the XYZ project will be completed by July 3,
N 15      All work on the XYZ project MUST be completed by June 27,
O 16      1981. This corresponds to the original commitment in
N 16      1981. This corresponds to the amended deadline in
O 17      the contract, and this commitment will be met.
N 17      the contract, and this deadline MUST be met.
I 17 *
I 17 *      I repeat:
I 17 *      All work on the XYZ project MUST be completed by June 27,
I 17 *      1981. This corresponds to the amended deadline in
I 17 *      the contract, and this deadline MUST be met.
O 22      John Q. Public
N 22      John Q. Public, Project Manager
D 23      Public Corporation
```

\* This is the line number after which the insertion was made.

Figure 4-4. Modification Listing File for the VDT Exercise

When you finish the exercises, delete the files that you created, .EXAMPLE and .LISTA, using the Delete File (DF) SCI command.

# Exercises Using a Hard-Copy Terminal

## 5.1 INTRODUCTION

This section contains Text Editor exercises using the 820 keyboard send/receive (KSR) terminal. Some procedures may vary for other teleprinter devices (TPDs). Step-by-step instructions are provided to familiarize you with each of the actions required to use the Text Editor effectively. For this reason, you should complete this section while at the terminal. Since this exercise uses a specific keyboard, key names that are printed on the keycaps are used in the instructions. Generic key names are not used in Section 5 except in reference to VDTs.

Before beginning these exercises, check with the person in charge of your system to determine if a log-on procedure is required. If so, you will need a user ID. Also, confirm that the file names .EXAMPLE and .LISTA are acceptable. If not, obtain other file names to replace them in all occurrences for the following exercises. Remember that you must press the RETURN key at the end of each line in which you enter data.

## 5.2 ACTIVATING THE SYSTEM COMMAND INTERPRETER

To activate SCI at a terminal, use the following procedure:

1. Turn on the terminal by pressing the ON/OFF switch on the back of the terminal to the ON position.
2. Place the operating mode switches in the LINE position and the VIEW position.
3. Hold down the CTRL key and press S, then press the exclamation point (!). (This is a shifted key).

The following additional steps for a log-on procedure are required on some systems:

4. If your system requires identification from the user (log-on), DNOS prints the following, one line at a time:

```
DNOS X.X.X
```

```
USER ID:  
PASSCODE:  
JOB NAME:
```

5. Type your assigned user ID and press the RETURN key. The printhead prints the PASSCODE prompt.
6. Type your assigned passcode and press the RETURN key. To preserve passcode security, the characters of the passcode are not displayed when you enter them.

#### NOTE

If additional log-on prompts are displayed, refer to the *System Command Interpreter (SCI) Reference Manual* for further information regarding log-on procedure.

7. Type TEST following the JOB NAME prompt and press the RETURN key.
8. SCI is activated and the terminal is ready for input. Refer to the *DNOS System Command Interpreter (SCI) Reference Manual* for further details.

### 5.3 EXERCISE 1 — TEXT EDITING SESSION

A typical text editing session includes the following activities:

- Activating the Text Editor
- Establishing a format
- Entering data
- Terminating the Text Editor
- Printing the file

When you respond to an SCI prompt, initial values are printed for some prompts. It is not necessary to backspace or clear the field. Type in your response from the original position of the printhead and press the Return key.

#### 5.3.1 Activating the Text Editor

1. Press CTRL X to enter SCI command mode.
2. Type XE and press RETURN to activate the Text Editor. The following field prompt is printed:

```
EXECUTE TEXT EDITOR
      FILE ACCESS NAME:
      EXCLUSIVE EDIT?:  YES
      LINE LENGTH:    80
```

3. Type two double quotation marks (" ") following the FILE ACCESS NAME prompt, then press RETURN. This response indicates that the Text Editor is to create a new file instead of editing an existing one. When you terminate the Text Editor, you are required to assign a file name to this file if you want to retain it. Press RETURN twice to accept the initial values for the next two prompts.
4. The following is printed:

```
* EOF
```

The \*EOF mark indicates that the only record in the file is the end-of-file record.

### 5.3.2 Establishing a Format

In this exercise, you are preparing a memorandum that is to be centered on the printed page. Therefore, your next step is to set the margins and tabs using the Modify Tabs (MT) and Modify Right Margin (MRM) commands. These commands can be used either during or before the editing session.

You set the left margin for the cursor and the tabs with the Modify Tabs (MT) command. Eighty columns (spaces) are available for data characters on each line (record). To indent the memorandum approximately one inch from each edge of the printout, the margins must be set at 10 and 74. This leaves a print area of six and a half inches, or 65 characters, on a standard pica (10 characters per inch) printer. Use the following procedure to set the tab stops and the right margin.

1. Press CTRL X to enter command mode. The SCI command mode prompt ([ ]) is printed.
2. To execute the Modify Tabs (MT) command, type the characters MT and press RETURN.
3. The TAB COLUMNS prompt appears displaying the initial values. Set the tab stops by typing the numbers, with no spaces and separated by commas, as they appear below:

```
MODIFY TABS
TAB COLUMNS: 10,15,20,25,30,35,40,45,50,55,60,65,70
```

4. Now press RETURN. This establishes the first tab stop at column 10 and additional tabs at five-space intervals. Although you do not need all of these tab stops to type the exercise text, the regularly spaced tabs permit you to use the edit control functions and commands that depend on tab settings. This also sets the left margin since the print-head moves to the lowest numbered tab setting when the RETURN key is pressed.
5. To execute the Modify Right Margin (MRM) command, press the CTRL X key. Type the characters MRM and press return to activate the Modify Right Margin command. Enter the number 74 in response to the prompt as follows:

```
MODIFY RIGHT MARGIN
RIGHT MARGIN POSITION: 74
```

6. Press the RETURN key. The right margin is now set at 74 and the display shows the \*EOF mark.

### 5.3.3 Entering Data

All edit functions are implemented through control keys and are represented by the word CTRL followed by the character.

1. To begin entering data, press RETURN. In compose mode, pressing RETURN generates a new, blank-filled line, which allows lines to be added easily as the new file is built. (You can use CTRL V to alternate between compose mode and edit mode, as appropriate.)

Notice the newly generated line. The printhead is positioned at the first tab stop.

2. Press the RETURN key again. Now, type the memorandum shown in Figure 5-1.

#### M E M O R A N D U M

DATE: December 5, 1980

TO: Tom Smith

COPY: John Doe

FROM: John Q. Public

SUBJECT: General Business Contract

All product documentation will be completed by March 23, 1981. This corresponds to the original deadline established in the contract.

Regards,

John Q. Public  
Public Corporation

Figure 5-1. Input for Memorandum Using the 820 KSR

### 5.3.4 Terminating the Text Editor

After you enter all of the data, terminate the Text Editor using the Quit Edit (QE) command as follows:

1. Press CTRL X to enter command mode, type QE, then press RETURN. The following lines are printed:

```
QUIT EDIT
      ABORT?: NO
```

2. Your response to the ABORT? field prompt allows you to choose to retain the data entered during the editing session (N or NO response) or to discard it (Y or YES response). Since you want to retain the data, press RETURN to accept the initial value (NO) that is supplied. The following is printed:

```
QUIT EDIT
      OUTPUT FILE ACCESS NAME:
```

3. Specify .EXAMPLE as the file name of the output file. The next field prompt is printed as follows:

```
QUIT EDIT
      OUTPUT FILE ACCESS NAME: .EXAMPLE
                        REPLACE?: NO
```

4. Since you are creating a new file, accept the initial value of NO for the REPLACE? prompt by pressing RETURN. This response prevents the accidental destruction of an existing file having the same file name. The next field prompt is printed as follows:

```
QUIT EDIT
      OUTPUT FILE ACCESS NAME: .EXAMPLE
                        REPLACE?: NO
      MOD LIST ACCESS NAME:
```

5. Respond to the MOD LIST ACCESS NAME field prompt by pressing RETURN (a null response) to specify that no modification listing is requested. The listing created by this prompt is described in a subsequent paragraph.

The Text Editor output file is now stored. The Text Editor is no longer active, and the SCI prompt ([ ]) is printed.

### 5.3.5 Printing the File

Examine the file you have just created, using the Show File (SF) SCI command to print the file. Press CTRL X, type SF, then press RETURN. The following is displayed:

```
SHOW FILE
      FILE PATHNAME:
```

Type .EXAMPLE, the name of the file you created in Exercise 1, and press RETURN. The file is printed on the terminal printer.

#### 5.4 EXERCISE 2 — REVISING AN EXISTING FILE

The following exercise provides the general procedure for using the Text Editor to revise an existing file. The input file used is the one you created previously.

Since the 820 terminal prints only one line at a time for you to edit, the revision process becomes easier when you work from a printout of the complete file. By marking the intended changes on the printout, you can then decide which commands to use to make the revisions. Use the Show File (SF) command described previously to print the file. The printout should be similar to the one shown in Figure 5-2. The line numbers will not be printed.

```
1  
2  
3 M E M O R A N D U M  
4  
5 DATE:    December 5, 1980  
6  
7 TO:      Tom Smith  
8  
9 COPY:    John Doe  
10  
11 FROM:    John Q. Public  
12  
13 SUBJECT: General Business Contract  
14  
15 All product documentation will be completed by March 21, 1981.  
16 This corresponds to the original deadline established in the  
17 contract.  
18  
19 Regards,  
20  
21  
22 John Q. Public  
23 Public Corporation
```

Figure 5-2. Display of the Input File on the 820 KSR

To revise the memorandum, use the following procedure:

1. Mark the intended revisions on the printout as follows:
  - a. Change the date on line 5 to *December 15, 1980*.
  - b. Add the name *Travis Williamson* after line 9.
  - c. Change the date on line 15 to *May 1, 1981* and change the word *original* to *amended* on line 16.
  - d. Insert the following sentence, starting on line 17: *Any desired modifications to the product must be cleared through this office.*
  - e. Delete the name *Public Corporation* on line 23.
2. To activate the Text Editor, enter command mode by pressing CTRL X, type XE, then press the RETURN key. The following is printed:

```
EXECUTE TEXT EDITOR
      FILE ACCESS NAME: .EXAMPLE
      EXCLUSIVE EDIT?: YES
      LINE LENGTH: 80
```

The file name, .EXAMPLE, is printed following the FILE ACCESS NAME field prompt because the Text Editor uses the output file name from the most recent editing session as the initial value for the input file name. (If no value is supplied, type .EXAMPLE.) Press RETURN twice to accept the initial values for the next two prompts.

3. Press RETURN; the first record from the input file, .EXAMPLE, is printed. Note that the line number is printed and the Text Editor is in edit mode.
4. To change the date, use CTRL J to position the printhead at line 5. The next line of the file is printed. Press CTRL J until you have displayed line 5. Enter the date, *December 15, 1980*, over the existing one by using the forward tab function, CTRL I, to move the printhead to the position at which the date is to be retyped. Type the rest of the line and press the RETURN key.
5. To add the name after line 9, move the printhead to line 10 using the RETURN key and press the insert line function key (CTRL O). The new line will precede line 10. Now, forward tab (CTRL I) until you are in position to insert the name *Travis Williamson*, and press RETURN to enter the changes you have made.
6. To change the date on line 15, use the Show Line (SL) command. Press CTRL X to enter command mode, type SL, then press RETURN. The following is printed:

```
SHOW LINE
      LINE: 1
```

7. In response to the LINE prompt, type 15 and press RETURN. When line 15 is printed, use CTRL T to back tab to the date. Correct it by typing *May 1, 1981*, followed by CTRL K.
8. Press RETURN to print line 16. Change the word *original* to the word *amended*.
9. Press RETURN to print line 17. Then finish typing the message to read: *Any desired modifications to the product must be cleared through this office*. Use CTRL O after the word *be* to insert a new line and complete the sentence.
10. To delete the name *Public Corporation*, located on line 23, use the Delete Lines (DL) command. To use this command, first press CTRL X to enter command mode. When the SCI prompt is printed, type DL and press RETURN. The following is printed one line at a time as you enter the responses:

```
DELETE LINES
      START LINE:
      END LINE:
```

To delete only line 23, respond to the prompts as follows:

```
DELETE LINES
      START LINE: 23
      END LINE: 23
```

After a few seconds, the following is printed, since line 23 is the last line in the file.

```
*EOF
```

The Delete Lines command is particularly useful when deleting large sets of lines from the file.

11. When the modifications to the file are complete, use the QE command to terminate processing. To do so, press CTRL X. The SCI prompt is printed. Type QE and press RETURN. The following is printed:

```
QUIT EDIT
      ABORT?: NO
```

12. Since you are going to retain this revision of the file, press RETURN to accept the initial value, NO, at the ABORT? prompt. As you respond to each field prompt, the next one is printed. The Quit Edit field prompts are as follows;

```
QUIT EDIT
      OUTPUT FILE ACCESS NAME: .EXAMPLE
      REPLACE?: NO
      MOD LIST ACCESS NAME:
```

13. Notice that the value entered earlier in response to the INPUT FILE ACCESS NAME prompt from the XE command appears in response to the OUTPUT FILE ACCESS NAME prompt. Press RETURN to accept it as the output file.

14. Type Y in response to the REPLACE? prompt. This indicates that the output file is to replace the input file. Now, press the RETURN key.
15. Type .LISTA in response to the MOD LIST ACCESS NAME prompt and press RETURN. (The listing produced by this field prompt response is discussed in a subsequent paragraph.)

Once this operation is complete, the SCI prompt is printed again.

The result of this text editing session (Figure 5-3) is stored in the file with the file name .EXAMPLE. You can print this file on the terminal by using the Show File (SF) command.

M E M O R A N D U M

DATE: December 15, 1980

TO: Tom Smith

COPY: John Doe  
Travis Williamson

FROM: John Q. Public

SUBJECT: General Business Contract

All product documentation will be completed by May 1, 1981. This corresponds to the amended deadline established in the contract. Any desired modifications to the product must be cleared through this office.

Regards,

John Q. Public

Figure 5-3. Text Editor Output File on the 820 KSR

### 5.4.1 Using the Modification Listing File

The Quit Edit command field prompt, MOD LIST ACCESS NAME, allows you to specify a file to which a listing of the file modifications (made during the editing session) are written. The listing contains only the lines that were changed in the recent editing session and thus provides an efficient method of confirming the changes. The file would appear as shown in Figure 5-4. The flags in the left margin have the following meanings:

Flag	Meaning
O	Old value
N	New value
I	Inserted line
D	Deleted line

#### TEXT EDITOR MODIFICATION LISTING

```
INPUT ACCESS NAME: .EXAMPLE
OUTPUT ACCESS NAME: .EXAMPLE
MOD LIST ACCESS NAME: .LISTA
```

```
O 5      DATE:      December 5, 1980
N 5      DATE:      December 15, 1980
I 9      Travis Williamson
O 15     All project documentation will be completed by March 21, 1981.
N 15     All project documentation will be completed by May 1, 1981.
O 16     This corresponds to the original deadline established in the
N 16     This corresponds to the amended deadline established in the
O 17     contract.
N 17     contract. Any desired modifications to the product must be
I 17     cleared through this office.
D 23     Public Corporation
```

**Figure 5-4. Modification Listing File for the 820 Exercise**

Note that in the modification listing shown in Figure 5-4 the input file name and the output file name are both listed. Modified lines are indicated by flagging the old value with an O and the new value with an N. Line numbers are indicated in the second column. The line number given for inserted lines, indicated with an I, is the line number of the line after which the insertion was made. The deleted line is indicated with a D, followed by the line number of the line that was deleted.

When you finish the exercises in this section, delete the files that you created, .EXAMPLE and .LISTA, by using the Delete File (DF) SCI command.

# Appendix A

## Keycap Cross-Reference

---

Generic keycap names that apply to all terminals are used for keys on keyboards throughout this manual. This appendix contains specific keyboard information to help you identify individual keys on any supported terminal. For instance, every terminal has an Attention key, but not all Attention keys look alike or have the same position on the keyboard. You can use the terminal information in this appendix to find the Attention key on any terminal.

The terminals supported are the 931 VDT, 911 VDT, 915 VDT, 940 EVT, the Business System terminal, and hard-copy terminals (including teleprinter devices). The 820 KSR has been used as a typical hard-copy terminal. The 915 VDT keyboard information is the same as that for the 911 VDT except where noted in the tables.

Appendix A contains three tables and keyboard drawings of the supported terminals.

Table A-1 lists the generic keycap names alphabetically and provides illustrations of the corresponding keycaps on each of the currently supported keyboards. When you need to press two keys to obtain a function, both keys are shown in the table. For example, on the 940 EVT the Attention key function is activated by pressing and holding down the Shift key while pressing the key labeled PREV FORM NEXT. Table A-1 shows the generic keycap name as Attention, and a corresponding illustration shows a key labeled SHIFT above a key named PREV FORM NEXT.

Function keys, such as F1, F2, and so on, are considered to be already generic and do not need further definition. However, a function key becomes generic when it does not appear on a certain keyboard but has an alternate key sequence. For that reason, the function keys are included in the table.

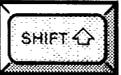
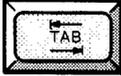
Multiple key sequences and simultaneous keystrokes can also be described in generic keycap names that are applicable to all terminals. For example, you use a multiple key sequence and simultaneous keystrokes with the log-on function. You log on by *pressing the Attention key, then holding down the Shift key while you press the exclamation (!) key*. The same information in a table appears as *Attention!(Shift)!*.

Table A-2 shows some frequently used multiple key sequences.

Table A-3 lists the generic names for 911 keycap designations used in previous manuals. You can use this table to translate existing documentation into generic keycap documentation.

Figures A-1 through A-5 show diagrams of the 911 VDT, 915 VDT, 940 EVT, 931 VDT, and Business System terminal, respectively. Figure A-6 shows a diagram of the 820 KSR.

Table A-1. Generic Keycap Names

Generic Name	911 VDT	940 <sup>1</sup> EVT	931 VDT	Business System Terminal	820 <sup>1</sup> KSR
Alternate Mode	None				None
Attention <sup>2</sup>		 			 
Back Tab	None	 	 	None	 
Command <sup>2</sup>					 
Control					
Delete Character					None
Enter					 
Erase Field					 

Notes:

<sup>1</sup>The 820 KSR terminal has been used as a typical hard-copy terminal with the TPD Device Service Routine (DSR). Keys on other TPD devices may be missing or have different functions.

<sup>2</sup>On a 915 VDT the Command Key has the label F9 and the Attention Key has the label F10.

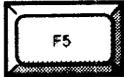
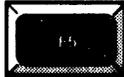
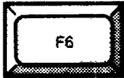
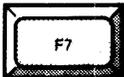
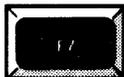
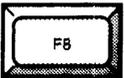
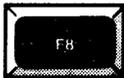
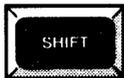
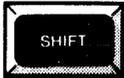
Table A-1. Generic Keycap Names (Continued)

Generic Name	911 VDT	940 EVT	931 VDT	Business System Terminal	820 <sup>1</sup> KSR
Erase Input					 
Exit			 	 	
Forward Tab	 			 	 
F1					 
F2					 
F3					 
F4					 

Notes:

<sup>1</sup>The 820 KSR terminal has been used as a typical hard-copy terminal with the TPD Device Service Routine (DSR). Keys on other TPD devices may be missing or have different functions.

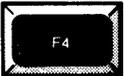
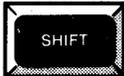
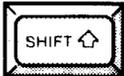
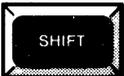
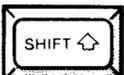
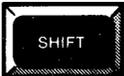
Table A-1. Generic Keycap Names (Continued)

Generic Name	911 VDT	940 EVT	931 VDT	Business System Terminal	820' KSR
F5					 
F6					 
F7					 
F8					 
F9	 			 	 
F10	 			 	 

Notes:

\*The 820 KSR terminal has been used as a typical hard-copy terminal with the TPD Device Service Routine (DSR). Keys on other TPD devices may be missing or have different functions.

Table A-1. Generic Keycap Names (Continued)

Generic Name	911 VDT	940 EVT	931 VDT	Business System Terminal	820' KSR
F11	 			 	 
F12	 			 	 
F13	 	 	 	 	 
F14	 	 	 	 	 
Home					 
Initialize Input		 			 

Notes:

The 820 KSR terminal has been used as a typical hard-copy terminal with the TPD Device Service Routine (DSR). Keys on other TPD devices may be missing or have different functions.

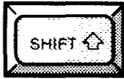
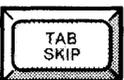
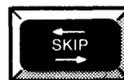
Table A-1. Generic Keypac Names (Continued)

Generic Name	911 VDT	940 EVT	931 VDT	Business System Terminal	820' KSR
Insert Character					None
Next Character	 or  				None
Next Field	 		 	 	None
Next Line					  or 
Previous Character	 or 				None
Previous Field		 			None

Notes:

\*The 820 KSR terminal has been used as a typical hard-copy terminal with the TPD Device Service Routine (DSR). Keys on other TPD devices may be missing or have different functions.

Table A-1. Generic Keypad Names (Continued)

Generic Name	911 VDT	940 EVT	931 VDT	Business System Terminal	820 <sup>1</sup> KSR
Previous Line					 
Print					None
Repeat		See Note 3	See Note 3	See Note 3	None
Return					
Shift					
Skip					None
Uppercase Lock					

Notes:

<sup>1</sup>The 820 KSR terminal has been used as a typical hard-copy terminal with the TPD Device Service Routine (DSR). Keys on other TPD devices may be missing or have different functions.

<sup>2</sup>The keyboard is typamatic, and no repeat key is needed.

228 47 34 (7/14)

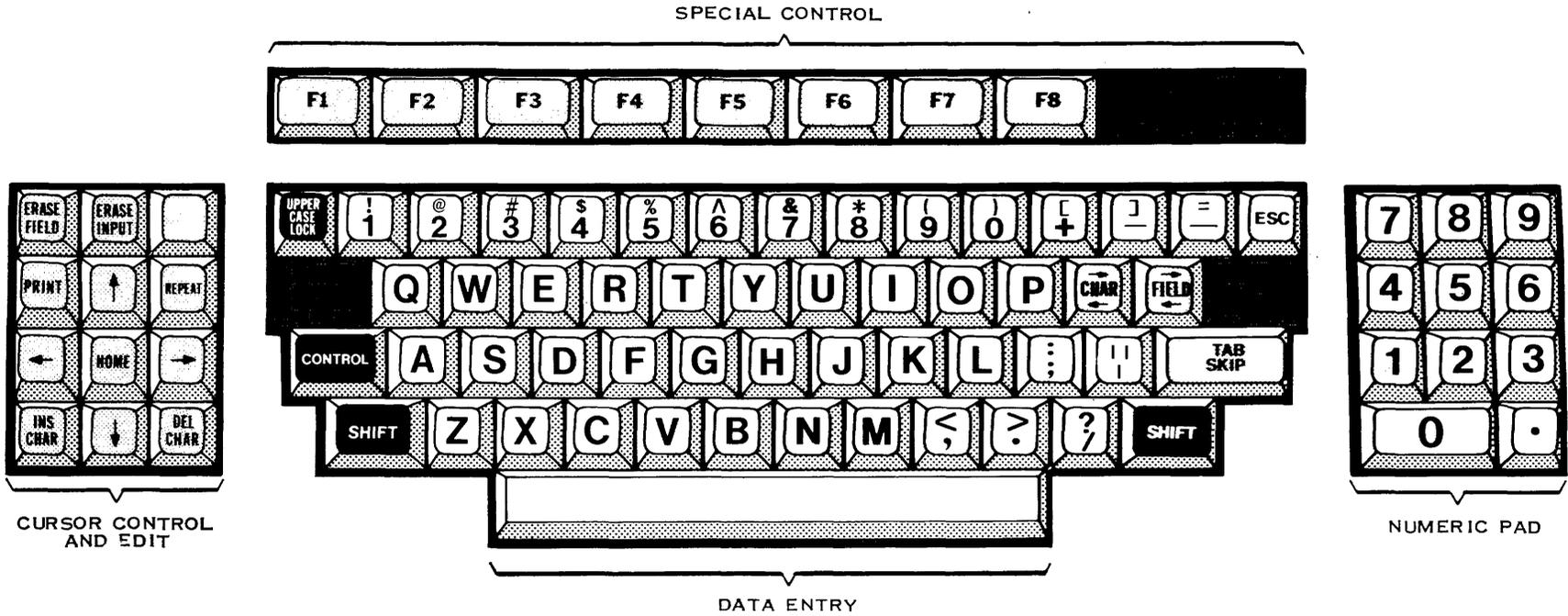
**Table A-2. Frequently Used Key Sequences**

Function	Key Sequence
Log-on	Attention/(Shift)!
Hard-break	Attention/(Control)x
Hold	Attention
Resume	Any key

**Table A-3. 911 Keycap Name Equivalents**

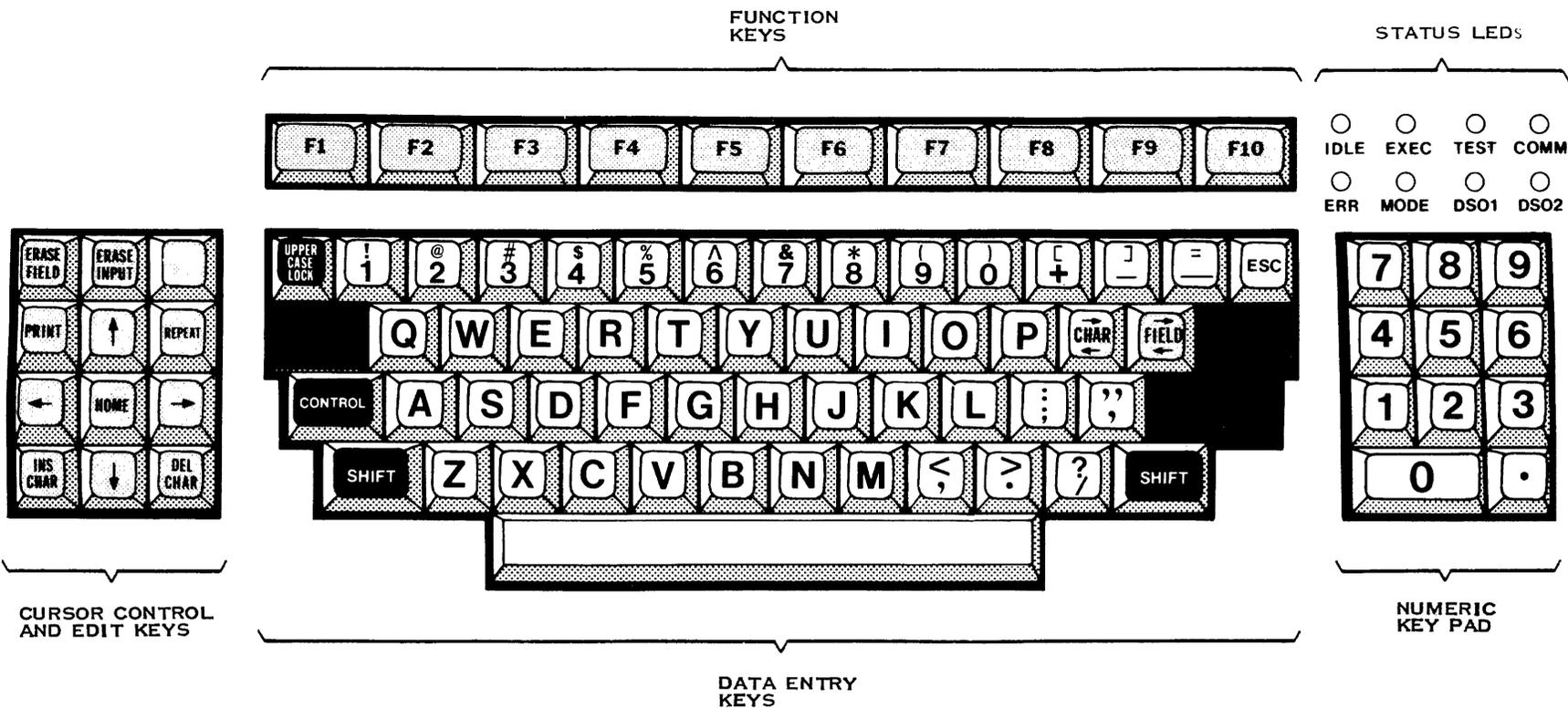
911 Phrase	Generic Name
Blank gray	Initialize Input
Blank orange	Attention
Down arrow	Next Line
Escape	Exit
Left arrow	Previous Character
Right arrow	Next Character
Up arrow	Previous Line

2284734 (8/14)



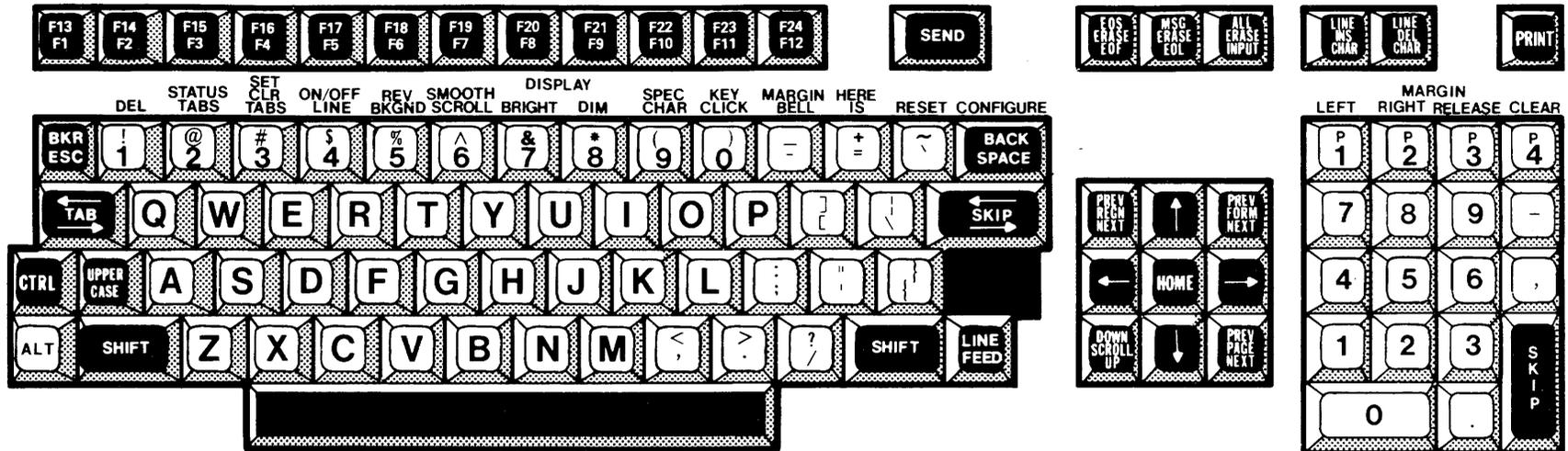
2284734 (9/14)

Figure A-1. 911 VDT Standard Keyboard Layout



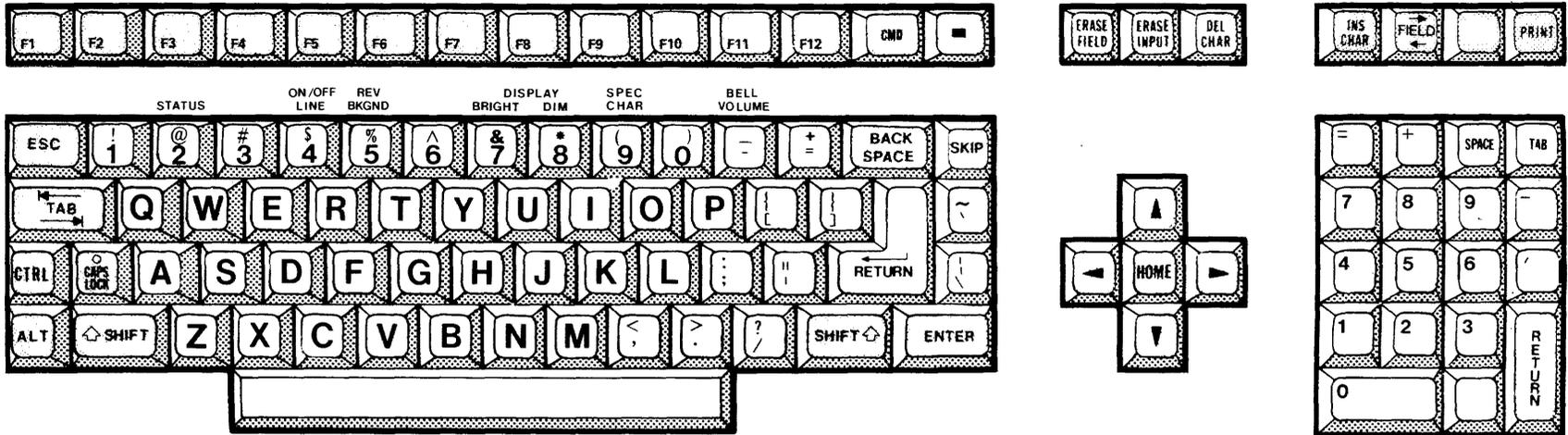
2284734 (10/14)

Figure A-2. 915 VDT Standard Keyboard Layout



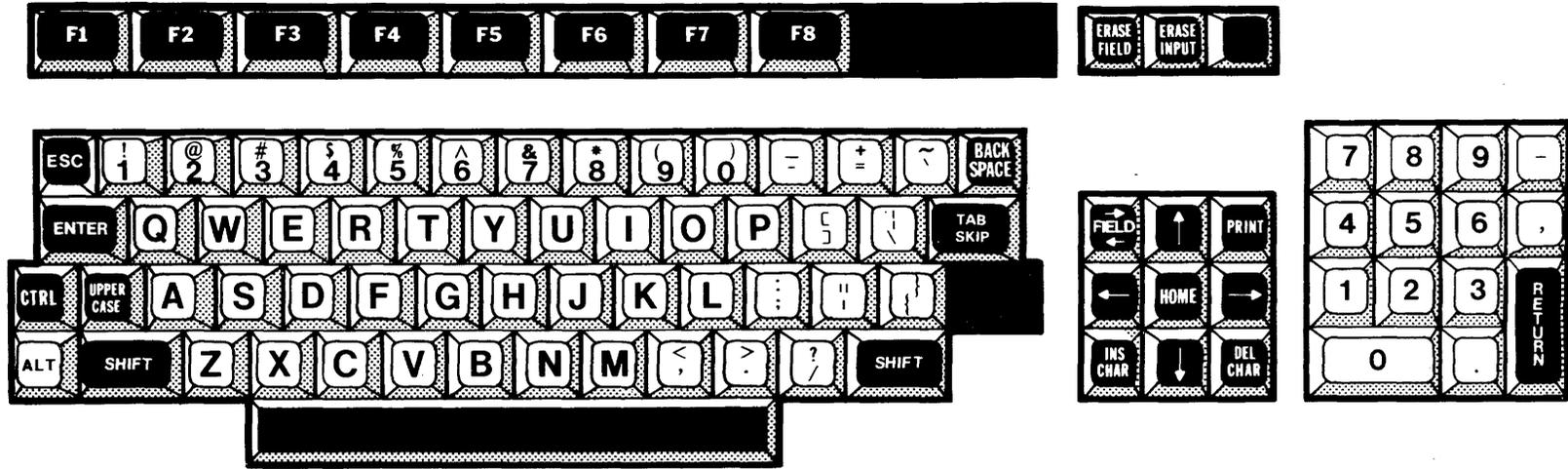
2284734 (11/14)

Figure A-3. 940 EVT Standard Keyboard Layout



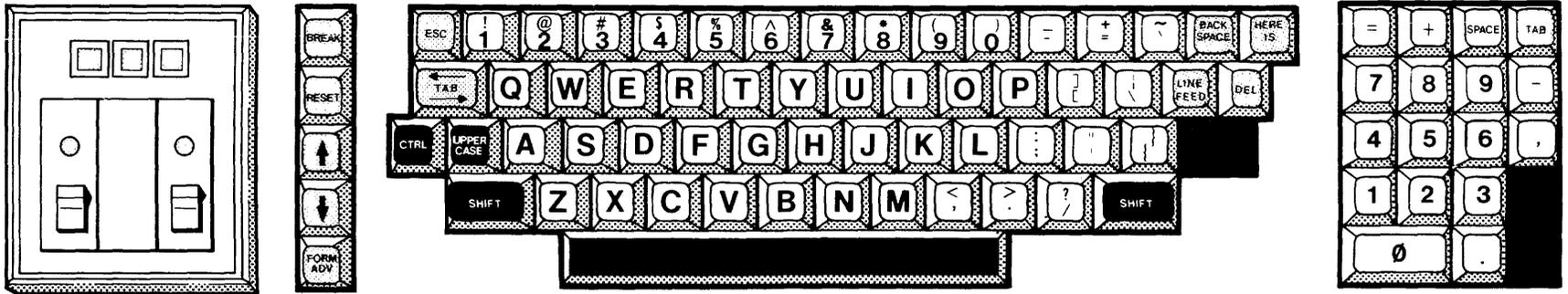
2284734 (12/14)

Figure A-4. 931 VDT Standard Keyboard Layout



2284734 (13/14)

Figure A-5. Business System Terminal Standard Keyboard Layout



2284734 (14/14)

Figure A-6. 820 KSR Standard Keyboard Layout

# Alphabetical Index

## Introduction

---

### HOW TO USE INDEX

The index, table of contents, list of illustrations, and list of tables are used in conjunction to obtain the location of the desired subject. Once the subject or topic has been located in the index, use the appropriate paragraph number, figure number, or table number to obtain the corresponding page number from the table of contents, list of illustrations, or list of tables.

### INDEX ENTRIES

The following index lists key words and concepts from the subject material of the manual together with the area(s) in the manual that supply major coverage of the listed concept. The numbers along the right side of the listing reference the following manual areas:

- Sections — Reference to Sections of the manual appear as “Sections x” with the symbol x representing any numeric quantity.
- Appendixes — Reference to Appendixes of the manual appear as “Appendix y” with the symbol y representing any capital letter.
- Paragraphs — Reference to paragraphs of the manual appear as a series of alphanumeric or numeric characters punctuated with decimal points. Only the first character of the string may be a letter; all subsequent characters are numbers. The first character refers to the section or appendix of the manual in which the paragraph may be found.
- Tables — References to tables in the manual are represented by the capital letter T followed immediately by another alphanumeric character (representing the section or appendix of the manual containing the table). The second character is followed by a dash (-) and a number.

Tx-yy

- Figures — References to figures in the manual are represented by the capital letter F followed immediately by another alphanumeric character (representing the section or appendix of the manual containing the figure). The second character is followed by a dash (-) and a number.

Fx-yy

- Other entries in the Index — References to other entries in the index preceded by the word “See” followed by the referenced entry.

- Access:**  
 File ..... 3.1.1  
 Rights ..... 1.7
- Activating:**  
 SCI:  
 Hard-Copy Terminal ..... 5.2  
 VDT ..... 4.2  
 Text Editor ..... 1.3  
 Exercises:  
 Hard-Copy Terminal ..... 5.3.1  
 VDT ..... 4.3.1
- Back Tab:**  
 Function:  
 CTRL T Key ..... 2.3.17  
 Previous Field Key ..... 2.2.15  
 Key ..... 2.2.15
- BACKSPACE/CTRL H Key, Move**  
 Printhead Left Function ..... 2.3.7
- Backspace, Nonerases Function**  
 CTRL 3 Key ..... 2.3.24
- Backspace One Character Function,**  
 Previous Character Key ..... 2.2.14
- Business System Keyboard Layout** .... FA-5
- CL Command** ..... 3.3
- Clear to tab Function:**  
 CTRL E Key ..... 2.3.5  
 F5 Key ..... 2.2.6
- Command** .... See Also Name or Mnemonic  
 of Individual Command  
 Description Format ..... 3.1.3, T3-4  
 Descriptions ..... 3.2  
 Key ..... 2.2.1  
 Mode ..... 2.2.1  
 Function CTRL X Key ..... 2.3.21
- Commands:**  
 Entering ..... 1.6  
 Text Editor ..... 3.2
- Control Character Codes Function:**  
 CTRL Key ..... 2.3.26  
 Control Key ..... 2.2.30
- Copy Lines (CL) Command** ..... 3.3
- Creation, File** ..... 3.1.1
- CTRL:**  
 A Key, Roll Up Function ..... 2.3.1  
 B Key, Roll Down Function ..... 2.3.2  
 C Key, Word Wrap Function ..... 2.3.3  
 D Key, Duplicate to Tab Function .... 2.3.4  
 E Key, Clear to Tab Function ..... 2.3.5  
 F Key, Line Numbers Function ..... 2.3.6  
 I Key, Forward Tab Function ..... 2.3.8  
 J Key, Move Printhead  
 Down Function ..... 2.3.9  
 K Key, Erase Rest of  
 Line Function ..... 2.3.10  
 Key, Control Character  
 Codes Function ..... 2.3.26  
 L Key, Move Printhead Home  
 Function ..... 2.3.12
- N Key, Delete Line Function** ..... 2.3.14
- O Key, Insert Line Function** ..... 2.3.15
- S Key, Left Margin Function** ..... 2.3.16
- T Key, Back Tab Function** ..... 2.3.17
- U Key, Move Printhead**  
 Up Function ..... 2.3.18
- V Key, Edit Mode/Compose**  
 Mode Function ..... 2.3.19
- W Key, Restore Line Function** ..... 2.3.20
- X Key, Command Mode Function** .. 2.3.21
- Y Key, Left Margin Function** ..... 2.3.22
- Z Key, Right Margin Function** ..... 2.3.23
- 3 Key, Backspace,**  
 Nonerases Function ..... 2.3.24
- Cursor Placement** ..... 3.1.1.2
- Data:**  
 Entering Exercise:  
 Hard-Copy Terminal ..... 5.3.3  
 VDT ..... 4.3.3
- Entry:**  
 Hard-Copy Terminal ..... 1.5  
 VDT ..... 1.4
- Delete:**  
 Character Function,  
 Delete Character Key ..... 2.2.23  
 Line Function CTRL N Key ..... 2.3.14  
 Line Function, Erase Input Key ..... 2.2.19  
 Lines (DL) Command ..... 3.4  
 String (DS) Command ..... 3.5
- Device:**  
 Name Prompt Responses ..... T3-2  
 Names, Legal ..... T3-2
- Display/Suppress Line Numbers**  
 Function, F6 Key ..... 2.2.7
- DL Command** ..... 3.4
- DS Command** ..... 3.5
- Duplicate to Tab Function:**  
 CTRL D Key ..... 2.3.4  
 F4 Key ..... 2.2.5
- Edit:**  
 Control Functions:  
 Hard-Copy Terminal ..... 2.3  
 VDT ..... 2.2, T2-1  
 Mode/Compose Mode Function:  
 CTRL V Key ..... 2.3.19  
 F7 Key ..... 2.2.8
- Editor, Function Keys, Text** ..... 2.3
- Enter Key, Left Margin Function** ..... 2.2.17
- Entering Commands** ..... 1.6
- Erase:**  
 Characters on Line Function,  
 Erase Field Key ..... 2.2.18  
 Entered Characters Function  
 RUB OUT/DEL Key ..... 2.3.11  
 Field Key, Erase Characters on Line  
 Function ..... 2.2.18  
 Input Key, Delete Line Function ... 2.2.19

- Rest of Line Function:
  - CTRL K Key ..... 2.3.10
  - Skip Key ..... 2.2.12
- Error Recovery ..... 1.6
- Execute Text Editor:
  - With Scaling (XES) Command ... 1.3, 3.19
  - (XE) Command ..... 1.3, 3.18
- Exercises:
  - Hard-Copy Terminal ..... 5.1
  - Text Editor VDT ..... 4.1
- Field Prompt:
  - Notation ..... 3.1.2, T3-3
  - Responses, Text Editor ..... 3.1.1
- File:
  - Access ..... 3.1.1
  - Creation ..... 3.1.1
  - Names ..... 3.1.1
  - Security ..... 1.7
  - Structure ..... 3.1.1
- Find String (FS) Command ..... 3.6
- Format, Command Description .. 3.1.3, T3-4
- Formatting Exercise:
  - Hard-Copy Terminal ..... 5.3.2
  - VDT ..... 4.3.2
- Forward Space One Character
  - Function, Next Character Key ..... 2.2.13
- Forward Tab:
  - Function:
    - CTRL I Key ..... 2.3.8
    - Forward Tab Key ..... 2.2.11
    - Key ..... 2.2.11
- FS Command ..... 3.6
- Function Keys, Text Editor ..... 2.2, 2.3, T2-1
- Functions:
  - Hard-Copy Terminal, Edit Control ..... 2.3
  - VDT, Edit Control ..... 2.2, T2-1
  - F1 Key, Roll Up Function, ..... 2.2.2
  - F2 Key, Roll Down Function ..... 2.2.3
  - F3 Key, Word Wrap Function ..... 2.2.4
  - F4 Key, Duplicate to Tab Function ..... 2.2.5
  - F5 Key, Clear to Tab Function ..... 2.2.6
  - F6 Key, Display/Suppress Line  
Numbers Function ..... 2.2.7
  - F7 Key, Edit Mode/Compose Mode  
Function ..... 2.2.8
  - F8 Key, Restore Line Function, ..... 2.2.9
- Generic Keycap
  - Names ..... Appendix A, 1.2, TA-1
- Hard-Copy Terminal:
  - Activating:
    - SCI ..... 5.2
    - Text Editor Exercise ..... 5.3.1
    - Data Entering Exercise ..... 5.3.3
    - Data-Entry ..... 1.5
    - Edit Control Functions ..... 2.3
    - Exercises ..... 5.1
  - Formatting Exercise ..... 5.3.2
  - Modification Listing File ..... 5.4.1, F5-4
  - Print File Exercise ..... 5.3.5
  - Revise File Exercise ..... 5.4
  - Terminating Text Editor Exercise ... 5.3.4
- Home Key, Move Cursor
  - Home Function ..... 2.2.28
- IF Command ..... 3.7
- Initialize Input Key, Insert Line  
Function ..... 2.2.20
- Insert:
  - Character:
    - Function, Insert Character Key ... 2.2.22
    - Key, Insert Character Function ... 2.2.22
  - File (IF) Command ..... 3.7
  - Line Function:
    - CTRL O Key ..... 2.3.15
    - Initialize Input Key ..... 2.2.20
- Key Sequences ..... TA-2
- Keyboard Layout:
  - Business System Terminal ..... FA-5
  - 820 KSR ..... FA-6
  - 911 VDT ..... FA-1
  - 915 VDT ..... FA-2
  - 931 VDT ..... FA-4
  - 940 VDT ..... FA-3
- Keycap:
  - Name Equivalents, 911 ..... TA-3
  - Names, Generic ... Appendix A, 1.2, TA-1
  - Keys Text Editor, Function .. 2.2, 2.3, T2-1
- Left Margin Function:
  - CTRL S Key ..... 2.3.16
  - CTRL Y Key ..... 2.3.22
  - Enter Key ..... 2.2.17
- Legal Device Names ..... T3-2
- Line Number:
  - Prompt Responses ..... 3.1.1
  - Values ..... 3.1.1
- Line Numbers ..... 3.1.1
  - Function CTRL F Key ..... 2.3.6
- Listing File for Hard-Copy  
Terminal, Modification ..... 5.4.1, F5-4
- MHR Command ..... 2.2, 2.3, 3.8
- ML Command ..... 3.9
- Mode, Command ..... 2.2.1
- Modification Listing File:
  - Hard-Copy Terminal ..... 5.4.1, F5-4
  - VDT ..... 4.4.1, F4-4
- Modify:
  - Horizontal Roll (MHR)  
Command ..... 2.2, 2.3, 3.8
  - Right Margin (MRM)  
Command ..... 2.2, 2.3, 3.11
  - Roll (MR) Command ..... 2.2, 2.3, 3.10
  - Tabs (MT) Command ..... 2.2, 2.3, 3.12

- Move:
  - Cursor:
    - Down Function, Next Line Key . . . . . 2.2.25
    - Home Function, Home Key . . . . . 2.2.28
    - Left Function, Previous Character Key . . . . . 2.2.27
    - Right Function, Next Character Key . . . . . 2.2.26
    - Up Function, Previous Line Key . . . . . 2.2.24
  - Lines (ML) Command . . . . . 3.9
  - Printhead:
    - Down Function CTRL J Key . . . . . 2.3.9
    - Home Function CTRL L Key . . . . . 2.3.12
    - Left Function BACKSPACE/CTRL H Key . . . . . 2.3.7
    - Up Function CTRL U Key . . . . . 2.3.18
  - MR Command . . . . . 2.2, 2.3, 3.10
  - MRM Command . . . . . 2.2, 2.3, 3.11
  - MT Command . . . . . 2.2, 2.3, 3.12
- Names, File . . . . . 3.1.1
- New Line Function:
  - RETURN/CTRL M Key . . . . . 2.3.13
  - Return Key . . . . . 2.2.10
- Next:
  - Character Key:
    - Forward Space One Character Function . . . . . 2.2.13
    - Move Cursor Right Function . . . . . 2.2.26
  - Field Key, Right Margin Function . . . . . 2.2.16
  - Line Key, Move Cursor Down Function . . . . . 2.2.25
- Notation:
  - Field Prompt . . . . . T3-3
  - Prompt . . . . . 3.1.2
  - Symbols . . . . . 3.1.2
- Placement, Cursor . . . . . 3.1.1
- Previous:
  - Character Key:
    - Backspace One Character Function . . . . . 2.2.14
    - Move Cursor Left Function . . . . . 2.2.27
  - Field Key, Back Tab Function . . . . . 2.2.15
  - Line Key, Move Cursor Up Function . . . . . 2.2.24
- Print File Exercise:
  - Hard-Copy Terminal . . . . . 5.3.5
  - VDT . . . . . 4.3.5
- Prompt:
  - Field Notation . . . . . 3.1.2, T3-3
  - Responses . . . . . T3-1
    - Device Name . . . . . T3-2
    - Line Number . . . . . 3.1.1
- QE Command . . . . . 1.8, 3.13
- Quit Edit (QE) Command . . . . . 3.13
- RE Command . . . . . 3.14
- Recover Edit (RE) Command . . . . . 3.14
- Replace String (RS) Command . . . . . 3.15, 4.4
- Response Type Indicators . . . . . T3-1
- Responses:
  - Device Name Prompt . . . . . T3-2
  - Line Number Prompt . . . . . 3.1.1
  - Prompt . . . . . T3-1
- Restore Line Function:
  - CTRL W Key . . . . . 2.3.20
  - F8 Key . . . . . 2.2.9
- RETURN/CTRL M Key, New Line Function . . . . . 2.3.13
- Return Key, New Line Function . . . . . 2.2.10
- Revise File Exercise:
  - Hard-Copy Terminal . . . . . 5.4
  - VDT . . . . . 4.4
- Right Margin Function:
  - CTRL Z Key . . . . . 2.3.23
  - Next Field Key . . . . . 2.2.16
- Rights, Access . . . . . 1.7
- Roll:
  - Down Function:
    - CTRL B Key . . . . . 2.3.2
    - F2 Key . . . . . 2.2.3
  - Up Function:
    - CTRL A Key . . . . . 2.3.1
    - F1 Key . . . . . 2.2.2
- RS Command . . . . . 3.15, 4.4
- RUB OUT/DEL Key, Erase Entered Characters Function . . . . . 2.3.11
- Save Lines (SVL) Command . . . . . 3.17
- Security, File . . . . . 1.7
- Show Line (SL) Command . . . . . 3.16
- Skip Key, Erase Rest of Line Function . . . . . 2.2.12
- SL Command . . . . . 3.16
- Suspending Text Editor . . . . . 1.3, 1.6
- SVL Command . . . . . 3.17
- Synonyms . . . . . 3.1.1
- Terminals Supported by Text Editor . . . . . 1.1
- Terminating Text Editor . . . . . 1.8
- Terminating Text Editor Exercise:
  - Hard-Copy Terminal . . . . . 5.3.4
  - VDT . . . . . 4.3.4
- Text Editor:
  - Activating . . . . . 1.3
  - Commands . . . . . 3.2
  - Field Prompt Responses . . . . . 3.1.1
  - Function Keys . . . . . 2.2, 2.3, T2-1
  - Suspending . . . . . 1.3, 1.6
  - Terminals Supported by . . . . . 1.1
  - Terminating . . . . . 1.8
  - VDT Exercises . . . . . 4.1
- Uppercase Function:
  - UPPERCASE Key . . . . . 2.3.25
  - Uppercase Lock Key . . . . . 2.2.29

Values, Line Number ..... 3.1.1  
VDT:  
  Activating:  
    SCI ..... 4.2  
    Text Editor Exercise ..... 4.3.1  
  Data:  
    Entering Exercise ..... 4.3.3  
    Entry ..... 1.4  
  Edit Control Functions ..... 2.2, T2-1  
  Exercises, Text Editor ..... 4.1  
  Formatting Exercise ..... 4.3.2  
  Modification Listing File ..... 4.4.1  
  Print File Exercise ..... 4.3.5  
  Revise File Exercise ..... 4.4  
  Terminating Text Editor Exercise ... 4.3.4

Word Wrap Function:  
  CTRL C Key ..... 2.3.3  
  F3 Key ..... 2.2.4  
  
XE Command ..... 1.3, 3.18  
XES Command ..... 1.3, 3.19  
  
820 KSR Keyboard Layout ..... FA-6  
911 VDT:  
  Keyboard Layout ..... FA-1  
  Keycap Name Equivalents ..... TA-3  
915 VDT Keyboard Layout ..... FA-2  
931 VDT Keyboard Layout ..... FA-4  
940 VDT Keyboard Layout ..... FA-3





FOLD



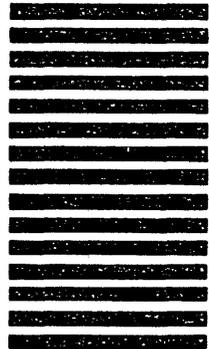
NO POSTAGE  
NECESSARY  
IF MAILED  
IN THE  
UNITED STATES

**BUSINESS REPLY MAIL**  
FIRST CLASS PERMIT NO. 7284 DALLAS, TX

POSTAGE WILL BE PAID BY ADDRESSEE

**TEXAS INSTRUMENTS INCORPORATED**  
DIGITAL SYSTEMS GROUP

ATTN: TECHNICAL PUBLICATIONS  
P.O. Box 2909 M/S 2146  
Austin, Texas 78769



FOLD