

IBM

*Personal Computer
Professional Series*

Professional Editor

6936795



*Personal Computer
Professional Series*

Professional Editor

Program by: Walter J. Paul

First Edition (December 1982)

International Business Machines Corporation provides this manual "as is" without warranty of any kind, either express or implied, including, but not limited to the implied warranties of merchantability and fitness for a particular purpose. IBM may make improvements and or changes in the product(s) and or the program(s) described in this manual at any time and without notice.

Products are not stocked at the address below. Requests for copies of this product and for technical information about the system should be made to your authorized IBM Personal Computer Dealer.

This publication could contain technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of this publication.

A Reader's Comment Form is provided at the back of this publication. If this form has been removed, address comments to: IBM Corp., Personal Computer, P.O. Box 1328-C, Boca Raton, Florida 33432. IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligations whatever.

PREFACE

This book is a guide to the installation and use of the IBM Personal Computer Professional Editor. This book provides both operating instructions and reference information. The following topics are covered:

- Capabilities and requirements of the program
- How to install and start up the program
- Special uses of the keyboard and screen
- How to use the program to create and edit files
- How to use the Professional Editor commands and macros
- Error messages and recovery procedures

Note: Throughout this manual, we will refer to the IBM Personal Computer Professional Editor as simply *Professional Editor*.

Organization of This Book

The information in this manual is organized as follows:

- Chapter 1 contains an introduction to the Professional Editor. It lists the functions and requirements of the program and tells you how to install it.
- Chapter 2 is a tutorial. It teaches you how to start up the Professional Editor and use the program menus and other options to create and edit a file. It also teaches you some of the more often-used commands available.

You should read Chapter 2 and complete the exercises while seated at your computer. You do not need to perform all of the exercises at one sitting; just be sure to save your data if you intend to stop and then start up the exercises again at a later time. (Instructions are provided for saving your files.)

Chapters 3-10 are reference chapters.

- Chapter 3 describes how the Professional Editor uses the IBM Personal Computer keyboard and screen and the types of files that it can process.
- Chapter 4 provides information on using the Professional Editor. It includes a description of the initialization function keys and the use of menus.

- Chapter 5 provides more information on using the Professional Editor. It tells how to start and end an editing session, describes the different modes, and covers file editing.
- Chapter 6 provides information on line commands.
- Chapter 7 provides detailed descriptions of the available editor commands, listed in alphabetical order.
- Chapter 8 discusses the use of data macros.
- Chapter 9 covers text processing features available in the Professional Editor.
- Chapter 10 discusses what you should do to recover from an error while running the program.
- Appendix A lists the messages produced by the Professional Editor and their meanings.
- A detachable reference card provides a quick reference for commands and keyboard keys.

Related Publications

We assume that you are familiar with the operation of the IBM Personal Computer and the Disk Operating System (DOS). DOS is described in the following publications:

- IBM Personal Computer *Guide to Operations*
- IBM Personal Computer *Disk Operating System*

CONTENTS

CHAPTER 1. INTRODUCTION	1-1
What Is the Professional Editor?	1-3
What Can You Do With It?	1-4
Program Versions	1-7
What Do You Need?	1-8
Installing the Program	1-9
CHAPTER 2. HOW TO USE THE PROFESSIONAL EDITOR IN FIVE EASY LESSONS	2-1
About the Lessons	2-3
Lesson 1 – Starting the Professional Editor	2-4
Primary Options Menu	2-4
Lesson 2 – Creating a File and Moving the Cursor	2-5
Profile Options Menu	2-6
Entering Data	2-7
Moving the Cursor	2-8
Steps to End the Editing Session	2-9
Lesson 3 – Modifying Text	2-10
If You Stopped after Lesson 2	2-10
Replacing a Character	2-11
Deleting a Character	2-12
Inserting a Character	2-12
Inserting a Line	2-13
Deleting a Line	2-14
Recovering Deleted Data	2-14
Ending Lesson 3	2-15
Lesson 4 – Using SEARCH and CHANGE	2-16
If You Stopped after Lesson 3	2-16
Using the Command Line	2-17
Using SEARCH and CHANGE Function Keys and Commands	2-17
Ending Lesson 4	2-19

Lesson 5 – Using Line Commands	2-20
If You Stopped after Lesson 4	2-20
Copying Lines	2-21
Deleting Lines	2-22
Moving Lines	2-23
Ending the Lessons	2-24

REFERENCE SECTION

CHAPTER 3. GENERAL

INFORMATION	3-1
Display Screen Operation	3-5
Screen Format	3-5
Current Line	3-6
Data Display Area (Lines 1 to 24)	3-6
Command Input Area and Status	
Display (Line 24)	3-7
Function Keys and Messages	
(Line 25)	3-9
Cursor Display	3-10
Using Tab Characters	3-10
Automatic Indentation	3-12
Display of Right Margin	3-13
Audio Operation	3-14
Keyboard Operation	3-15
The Enter Key ()	3-16
Control and Numeric Keypad Usage ...	3-16
Normal Mode Operation	3-17
Control Mode Operation	3-18
Scroll Mode Operation	3-19
Numeric Mode Operation	3-20
Data Mode Operation	3-21
Cursor Operations	3-21
Character Operations	3-24
Line Operations	3-25
Scrolling Operations	3-26
Keyboard Mode Selection	3-28
Program Function Keys	3-30
Key Set 0 Assignments	3-31
Key Set 1 Assignments	3-33
Key Set 2 Assignments	3-35
Key Set 3 Assignments	3-38

File Operations	3-39
File Format	3-39
Maximum Line Length	3-39
Character Set Restrictions	3-40
Multi-Segment Files	3-41
File Integrity	3-42
Cautions on File Specifications	3-43
Changing Diskettes	3-43

CHAPTER 4. INITIALIZATION FUNCTION

KEY USAGE AND MENUS	4-1
Initialization Function Key Usage	4-3
KEYS Function Key (F1)	4-3
PRIME Function Key (F3)	4-3
OPTION Function Key (F4)	4-4
TABS Function Key (F5)	4-4
MACROS Function Key (F6)	4-4
FILES Function Key (F7)	4-4
RESETP Function Key (F8)	4-4
EDIT Function Key (F9)	4-4
RETURN Function Key (F10)	4-4
Entering Data on the Menus	4-5
Initialization Menus	4-6
Primary Options Menu	4-7
Selecting the Input File	4-8
Selecting the Output File	4-9
Selecting the Profile File	4-10
Selecting the List Files	4-11
Selecting the Display Mode	4-11
Selecting the Default Disk	4-12
Text Buffer Size	4-12
Profile Options Menu	4-13
Selecting Options	4-14
Tabs and Mask Options Menu	4-23
Tab Settings	4-24
Input Mask Data	4-24
Macro Definition Menu	4-25
File List Menu	4-26
Displaying the File List Menu	4-26

CHAPTER 5. USING THE PROFESSIONAL EDITOR	5-1
Beginning the Edit Session	5-3
Professional Editor Modes	5-5
Edit Mode	5-5
Input Mode	5-6
Scrolling	5-6
Vertical Scrolling	5-6
Horizontal Scrolling	5-7
Automatic Scrolling	5-8
Editing Lines	5-9
Changing Text	5-9
Inserting Characters	5-11
Deleting Characters	5-11
Recovering Deleted Characters	5-12
Splitting a Line into Multiple Lines	5-13
Inserting New Lines	5-13
Deleting Lines	5-14
Recovering Deleted Lines	5-14
Ending the Edit Session	5-15
With Save (END Function Key)	5-15
Without Save (QUIT Function Key) ...	5-16
From the Primary Options Menu	5-16
CHAPTER 6. LINE COMMANDS	6-1
Line Commands	6-3
Entering Line Commands	6-3
Cancelling an Incomplete (Pending) Line Command	6-4
Performing Other Editing Functions	6-4
Syntax of Line Commands	6-5
Count Operand	6-5
Operation Sub-Commands	6-5
Destination Sub-Commands	6-6
Copy Operations	6-7
Copying a Single Line	6-7
Copying a Group of Lines	6-7
Copy and Repeat Operations	6-10

Delete Operations	6-11
Deleting a Single Line	6-11
Deleting a Group of Lines	6-11
Deleting to End of Segment	6-12
Deleting from Beginning of Segment	6-12
Extract Operations	6-13
Opening the Extract File	6-13
Extracting a Single Line	6-14
Extracting a Group of Lines	6-14
Extracting to End of Segment	6-15
Extracting from Beginning of Segment	6-15
Closing the Extract File	6-16
Gather Operations	6-16
Gathering a Group of Lines	6-17
Gathering to End of Segment	6-18
Gathering from Beginning of Segment	6-18
Move Operations	6-19
Moving a Single Line	6-19
Moving a Group of Lines	6-19
Print Operations	6-21
Printing a Group of Lines	6-21
Printing to End of Segment	6-22
Printing from Beginning of Segment ...	6-22
Uppercase and Lowercase	
Translation Operations	6-23
Translating a Single Line	6-23
Translating a Group of Lines	6-23
Translating to End of Segment	6-24
Translating from Beginning of Segment	6-24
Displaying the Column Grid	6-25

CHAPTER 7. EDITOR COMMANDS	7-1
Using Editor Commands	7-3
Using the Command Stack	7-4
Command Syntax	7-5
BOTTOM Command	7-6
BOUNDS Command	7-7
CHANGE Command	7-8
COLUMN Command	7-14
DOWN Command	7-15
ECLOSE Command	7-16
END Command	7-17
EOPEN Command	7-18
FILES Command	7-19
GET Command	7-20
LEFT Command	7-21
LINENO Command	7-22
MACRO Command	7-23
MARGIN Command	7-25
MASK Command	7-26
MERGE Command	7-27
NEXT Command	7-28
PUT Command	7-30
QUIT Command	7-31
RIGHT Command	7-32
SEARCH Command	7-33
SET Commands	7-40
TABS Command	7-44
TOP Command	7-45
UP Command	7-46
CHAPTER 8. USING MACROS	8-1
Data Macros	8-3
Phrase Macros	8-4
File Macros	8-4
Defining Data Macros During	
Initialization	8-5
Defining a Phrase Macro	8-6
Defining a File Macro	8-7
Deleting a Macro Definition	8-7
Saving the Macro Definitions	8-7

Calling Macros During Editing	8-8
Defining and Deleting Macro Definitions	
During Editing	8-9
Examples of Data Macro Usage	8-10
Example 1 - BASIC Language	
Data Macros	8-10
Example 2 - GML Language	
Data Macros	8-11
CHAPTER 9. TEXT PROCESSING	
FEATURES	9-1
Using Margins	9-5
Defining Margins	9-6
Display of the Right Margin	9-6
Using Automatic Word Wrapping	9-7
Paragraph Indentation	9-8
Text Formatting Operations	9-9
Formatting a Block of Text	9-10
Justifying a Block of Text	9-12
Centering Lines	9-13
Removing Tab Characters	9-14
CHAPTER 10. RECOVERING FROM	
ERROR SITUATIONS	10-1
Using the Backup File Option	10-3
Recovering from Disk I/O Errors	10-5
Recovering from Disk Full Conditions	10-9
Switching Drives (Option 1)	10-10
Changing Diskettes (Option 2)	10-11
Ending the Session (Option 3)	10-11
Picking up the Pieces	10-11
Combining the Pieces with DOS	10-12
Errors During Print Operations	10-13
APPENDIX A. MESSAGES	A-1
INDEX	X-1

CHAPTER 1. INTRODUCTION

Contents

What Is the Professional Editor?	1-3
What Can You Do with It?	1-4
Program Versions	1-7
What Do You Need?	1-8
Installing the Program	1-9

What Is the Professional Editor?

The Professional Editor is a program that operates on the IBM Personal Computer under the IBM Personal Computer Disk Operating System (DOS). We assume that you are familiar with the operation of the IBM Personal Computer and DOS. For information on DOS, refer to the IBM Personal Computer *Guide to Operations* or IBM Personal Computer *Disk Operating System*. We also assume that you are familiar with an editor program and/or have done some file editing. Many of the terms used in this book are commonly used with most editor programs.

The Professional Editor uses the extensive functions available in the IBM Personal Computer keyboard and displays to let you create and modify text files which may be stored on diskettes. These DOS format files may contain any type of data in ASCII code with line lengths that do not exceed 140 characters, such as:

- Program source code
- Document text
- Program data

Warning: You must not use the Professional Editor with non-ASCII files, such as absolute program modules (COM or EXE files) or binary data, because the Professional Editor deletes any null characters (binary zeros) from the file.

What Can You Do with It?

The Professional Editor program is called a *full screen editor* because it displays 24 lines of data on the screen and allows you to directly modify the data in any portion of the screen. The Professional Editor supports the monochrome display option and the color graphics display option. It operates in 80-character black and white or color modes.

Some of the advantages of the Professional Editor are:

- Menus provide easy selection of files and program options.
- All data present on the screen may be directly modified from the keyboard.
- In addition to moving the cursor on the screen, the cursor control keys can be used to perform functions such as skipping words and positioning the cursor to the beginning or end of the line.
- Characters may be inserted and deleted within a line. Characters that are deleted are saved on a 140-byte stack (the last 140 characters that were deleted are saved in a list) and may be recovered if they were deleted accidentally.
- Program function keys simplify many editing operations. Line 25 of the screen shows the function performed by each key.

- Lines may be inserted and deleted. Lines that are deleted are saved on a 5-line stack (the last five lines that were deleted are saved in a list) and may be recovered if they were deleted accidentally.
- Extensive scrolling facilities let you scroll forward and backward over the file, left and right over the lines on the screen, and to the top and bottom of the text.
- Optional automatic scrolling changes the portion of the file displayed on the screen when the cursor is moved out of the current screen.
- You can define a specific character string for lines to be inserted.
- Extensive search and change facilities are available.
- Line commands let you operate on single lines or groups of consecutive lines. You can perform operations such as move, copy, delete, translate, and others.
- Text from another file may be merged into the file being edited at any selected point.
- Portions of text from the file being edited may be extracted (copied) into another file.
- Data macros allow you to imbed frequently-used phrases and data structures in the file being edited with a single keystroke.

- You can print the current contents of the screen or a selected portion of the file being edited on the IBM Matrix Printer.
- Text formatting features allow selected blocks of text to be reformatted to fit between margins that you define. You can optionally justify text with respect to the left and right margins. You can also center lines.
- An automatic word wrapping feature allows text to be entered quickly. As you type, text that extends beyond the right margin is automatically placed at the beginning of the next line.
- Recovery procedures permit you to recover from out-of-disk-space conditions and from certain types of disk I/O errors.
- Files that are too large to fit into available memory may be processed in segments.
- Variable tabs are supported and may be redefined during the editing session.
- A command stack lets you recall and reuse previously entered commands with or without modification.
- A profile facility lets you save editor options for different types of data and recall those options during future editing sessions.
- Both monochrome and color displays are supported in 80-column mode. You can switch between display modes while editing.

Program Versions

The Professional Editor program comes in two versions:

- The standard version (EDIT), which runs on IBM Personal Computers with more than 64K bytes of memory. This version contains all of the features described in this manual.
- A minimum version (EDIT64), which runs on IBM Personal Computers with only 64K bytes of memory. This version contains all of the features of the Professional Editor described in this manual *except* the following:
 - The Data Macro Definition Menu. (You can still define macros by using the MACRO command.)
 - The File List Menu.

The Data Macro Definition Menu and File List Menu are discussed in Chapter 4.

What Do You Need?

You need the following hardware and software to operate the Professional Editor:

- An IBM Personal Computer with a minimum of 64K bytes of memory (for the minimum version) or 96K bytes of memory (for the standard version). For most applications, 128K bytes is recommended.
- At least one diskette drive. A second drive is highly recommended.
- An 80-column display. The IBM Monochrome Display is recommended because of its high resolution and readability. The Color Display adapter may also be used in both black and white and color modes. (Forty column mode *is not* supported.) When using the color adapter, you must use an RGB monitor in order to obtain sufficient resolution.
- To use the print screen and print text features, the IBM Matrix Printer must be installed.
- The IBM Personal Computer Disk Operating System (DOS).
- The Professional Editor diskette, which contains the file EDIT.EXE (or EDIT64.EXE) and some sample profile and data files.

Installing the Program

The editor program is distributed as two executable files: EDIT and EDIT64. Before you run the program, you must make a backup copy of the editor program you are going to use and put your original distribution diskette away in a safe place.

We will call the diskette with the backup copy your *working* diskette. This is the diskette you will use for your day-to-day operations.

- To install the standard version of the program, copy the EDIT.EXE file from the distribution diskette to your working diskette.
- To install the minimum version of the program, copy the EDIT64.EXE file from the distribution diskette to your working diskette.

You may rename the program module to any desired filename as long as the file extension is .EXE. We suggest that you use the name **EDIT**, to remind you which program you are running. Also, all of the examples in this book refer to EDIT, even if you are using EDIT64.

You should also copy the DOS command processor module COMMAND.COM to the working diskette, since the Professional Editor program may overlay the command processor in memory when a large file is being edited. If the command processor is overlaid in memory, you will have to insert the DOS diskette after ending the editing session.

The distribution diskette also contains some sample profile and data files, which you may copy onto your working diskette if you wish. These files will be discussed in Chapter 8.

CHAPTER 2. HOW TO USE THE PROFESSIONAL EDITOR IN FIVE EASY LESSONS

Contents

About the Lessons	2-3
Lesson 1 – Starting the Professional Editor	2-4
Primary Options Menu	2-4
Lesson 2 – Creating a File and Moving the Cursor	2-5
Profile Options Menu	2-6
Entering Data	2-7
Moving the Cursor	2-8
Steps to End the Editing Session	2-9
Lesson 3 – Modifying Text	2-10
If You Stopped after Lesson 2	2-10
Replacing a Character	2-11
Deleting a Character	2-12
Inserting a Character	2-12
Inserting a Line	2-13
Deleting a Line	2-14
Recovering Deleted Data	2-14
Ending Lesson 3	2-15

Lesson 4 – Using SEARCH and CHANGE	2-16
If You Stopped after Lesson 3	2-16
Using the Command Line	2-17
Using SEARCH and CHANGE	
Function Keys and Commands	2-17
Ending Lesson 4	2-19
 Lesson 5 – Using Line Commands	2-20
If You Stopped after Lesson 4	2-20
Copying Lines	2-21
Deleting Lines	2-22
Moving Lines	2-23
Ending the Lessons	2-24

About the Lessons

This chapter teaches you in five lessons how to use some of the more commonly-used functions of the Professional Editor. We do not try to teach you everything that you can do with the Professional Editor, but we will cover some basic operations. If you are experienced in the use of an editor program, you may choose to skip this chapter and use only the reference section of this book.

Each lesson in this chapter builds on the information you learned and work you did in the previous lesson. You do not need to complete all five lessons in one sitting. At the end of each lesson are instructions for ending the session or going on to the next lesson.

The lessons cover the following topics:

- Lesson 1 – Starting the Professional Editor.
- Lesson 2 – Entering data in a new file and moving the cursor on the screen.
- Lesson 3 – Modifying your file by changing, inserting, and deleting characters and lines of text.
- Lesson 4 – Performing *search* and *change* operations.
- Lesson 5 – Using *line commands* to copy, delete, move, and print lines of text in a file.

Just follow the instructions in this book, and you will soon be ready to use the Professional Editor for many of your editing tasks.

All set? Okay, let's begin.

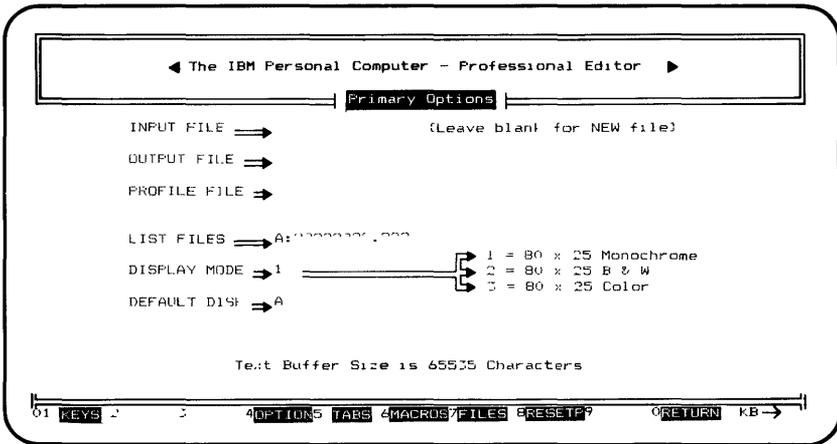
Lesson 1 – Starting the Professional Editor

You should be seated at your IBM Personal Computer with your working diskette inserted in drive A and the Dos A> prompt on the screen. (See “Installing the Program” in Chapter 1.)

At the DOS prompt A>, type **EDIT** and press the Enter key.

You see the IBM logo screen. After about three seconds, the logo screen disappears, and you see the Professional Editor Primary Options Menu.

Primary Options Menu



That's all you have to do to start the Professional Editor. Remember,

1. Start DOS.
2. Insert your Professional Editor working diskette in drive A.
3. At the **A>** prompt, type **EDIT** and press the Enter key.

Lesson 2 – Creating a File and Moving the Cursor

We are going to create a new file, so we do not have an input filename to enter. Press the Enter key to move the cursor to the **OUTPUT FILE** line, and type the name **EXAMPLE.TXT**.

Now press the Enter key.

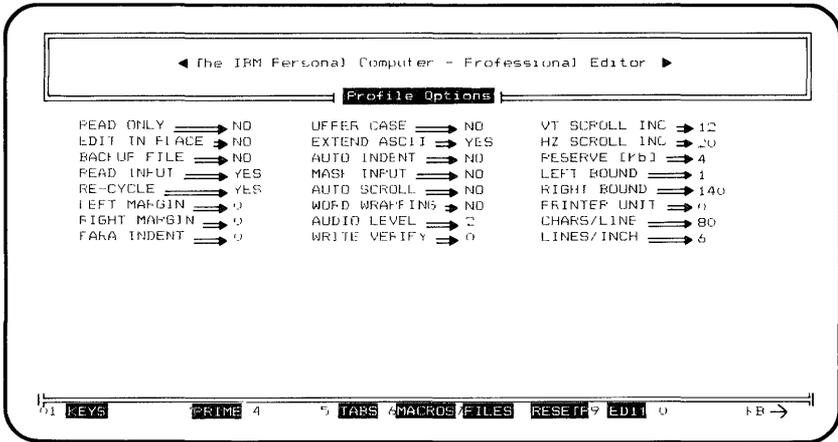
You see that the **PROFILE FILE** has been named **TXT.PRF**. The Professional Editor uses the extension that you assign to the output file as the name of the profile file, adding an extension of **.PRF**.

For the lessons in this chapter, we are going to use the default values for the display mode and default disk.

Now notice the bottom line of the screen. This line shows you the active function keys. Press the F4 (OPTION) key.

You see the following Professional Editor Profile Options Menu:

Profile Options Menu

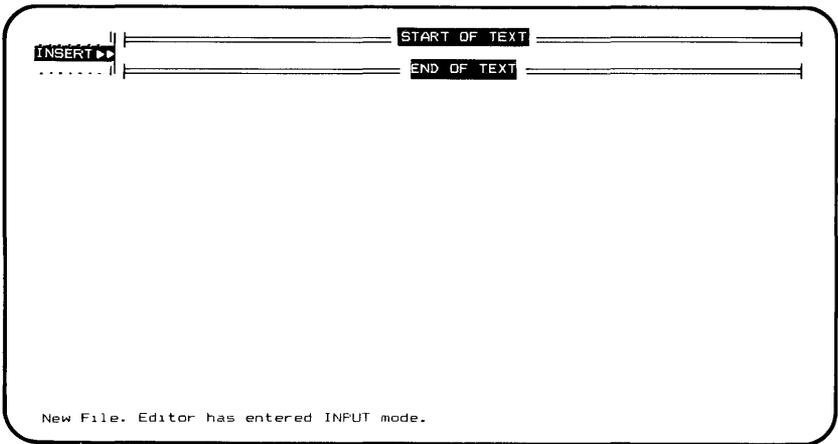


On this menu, change the **RIGHT MARGIN** option to **72** and the **WORD WRAPPING** option to **YES** by moving the cursor to those fields and typing over the existing values. (Use the keys with the arrows on the numeric keypad.)

This tells the Professional Editor that you want your lines of text to be only 72 characters long, and if the words you are typing extend beyond column 72, they should be moved automatically to the next line.

Entering Data

Now press the F9 (EDIT) key. Your screen should look like this:



We want you to type the following paragraph, just as you see it here. (Don't worry if you make a mistake, we will go back and correct any mistakes later. For now, we just want you to practice typing and using the keyboard.) Ready? Okay, type the following:

The Professional Editor is easy to use. It is a full-screen editor that lets you enter or modify data in any area on the screen. Notice how the Professional Editor "wraps" your words around when you select Word Wrapping on the Primary Menu.

Moving the Cursor

Notice how the line with the cursor is highlighted on your screen. As the cursor moves to a different line, the new line is highlighted. Let's see that again.

Press the Cursor Up key (Up Arrow/8). Notice how the next-to-the-last line is now highlighted. Now press the Cursor to BOL key (Home/7), and then press the Cursor Up key until the cursor is at the beginning of your file.

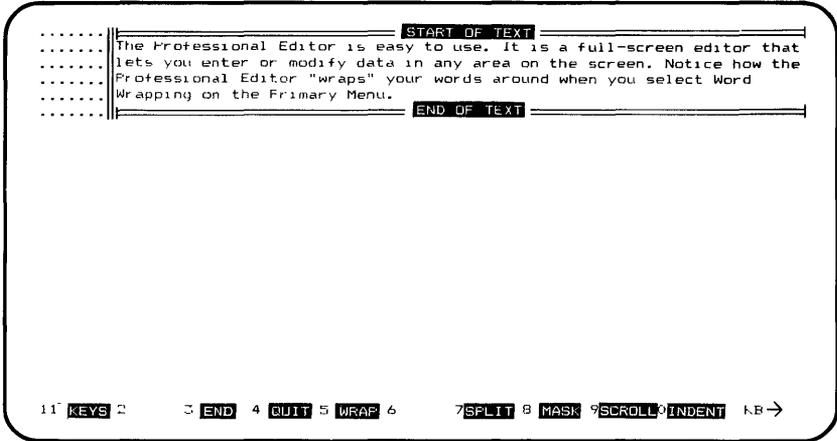
Notice how the highlighted line changed each time the cursor moved to a different line; but none of the data on the line changed.

You have finished Lesson 2. You may want to practice moving the cursor some more on your own before you go on to Lesson 3. If you want to end your session at this point, without going on to the next lesson, perform the following steps. Otherwise, turn now to Lesson 3.

Note: You may want to read the following section even if you don't want to end your session at this time. The steps for ending a session after editing a file are always the same.

Steps to End the Editing Session

Continue to press the F1 key until you see 11 in the bottom left corner of the screen. This is key set 1.



You see that F3 lets you end the session, so press the F3 key. You now have a file called EXAMPLE.TXT stored on your diskette. When you are ready to begin the next lesson, you will need to read this file in on the Primary Options Menu.

To return to DOS from the Primary Options Menu, press the RETURN function key (F10).

Lesson 3 – Modifying Text

If You Stopped after Lesson 2

If you are continuing with Lesson 3 directly after Lesson 2, go to “Replacing a Character” below. If you are starting Lesson 3 after saving the `EXAMPLE.TXT` file in Lesson 2, perform the following steps:

1. Start up DOS, and then at the `A>` prompt, enter `EDIT` to start up the Professional Editor.
2. Type the name `EXAMPLE.TXT` for the `INPUT FILE` on the Primary Options Menu.
3. Press the Enter key.
4. Notice that the `OUTPUT FILE` and `PROFILE FILE` are filled in by the program.
5. Press the Enter key again to tell the Professional Editor that you want to use those filenames.
6. Now press the F9 function key to edit the file.
7. You should see the same file on your screen that you worked on in Lesson 2.

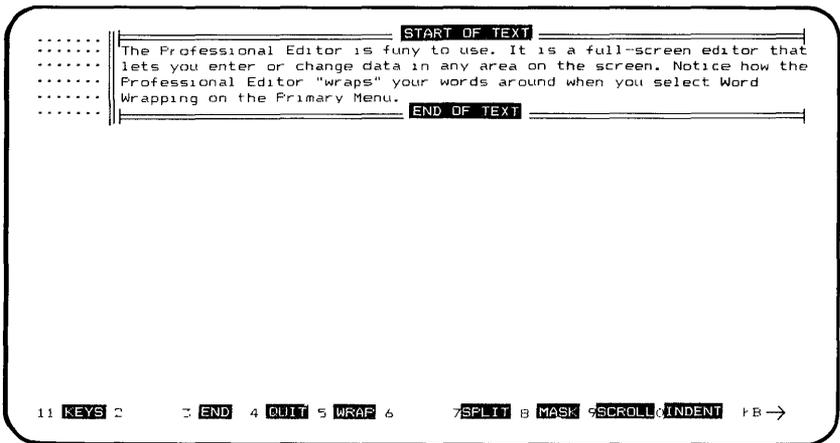
Replacing a Character

Now that you have some data on the screen, let's see how you can change it by using the Professional Editor. The first thing we're going to do is replace some characters in the text with some new characters.

Using what you learned in the last lesson about moving the cursor, place the cursor under the **m** in the word **modify**. We're going to change the word **modify** to the word **change**. Now, just type the word **change** on top of the word **modify**.

To replace one character with another character, all you do is type the new character in the same position as the old character. Let's try it again.

Move the cursor to the **e** in **easy**. Now type the word **fun**



Notice that you have a **y** on the end of **fun**. We didn't type as many new letters as we had in the old word. Do you know how to delete the **y**?

Deleting a Character

Your cursor should be under the **y** at the end of the word **fun**. If it isn't, move it there now.

Now press the Del key. The **y** disappears, and the rest of the line moves to the left one space. Each time you press the Del key, a character is deleted from the screen and from your file.

Inserting a Character

Let's see how to insert characters in a line. We're going to add the word **Options** to the name of the menu in the last line.

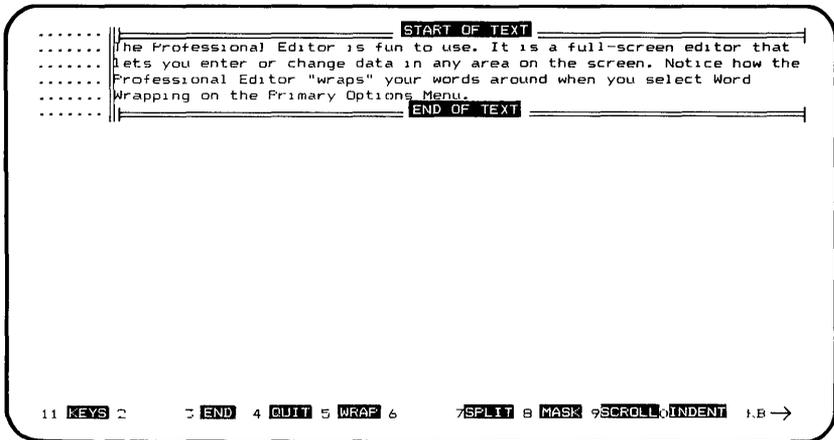
Move the cursor to the **M** in the word **Menu** in the last line.

Now press the Ins key. Pressing the Ins key puts you in Insert mode. (Notice the **I** in the bottom right corner of the screen.)

Now type the letter **O**. Notice how the **O** appears on the screen, and the word **Menu** shifts over to the right one space.

Type the rest of the letters to finish the word **Options**, and then press the Spacebar. Then press the Ins key again to take you out of Insert mode.

See how the word **Options** is placed correctly.



Inserting a Line

You've seen how to insert characters in an existing line of text. Now let's see how to insert a new line.

With the cursor still in the last line of data, press the INSERT function key (F5). (If you don't see key set 0 on your screen, continue to press the F1 function key until you see key set 0 in order to use the INSERT function key F5.) You see a new blank line, starting with the word **INSERT**.

Type the following:

In the Professional Editor, you can insert and delete single characters or entire lines.

Now press the Enter key to end the line. Press Enter once more to exit from input mode.

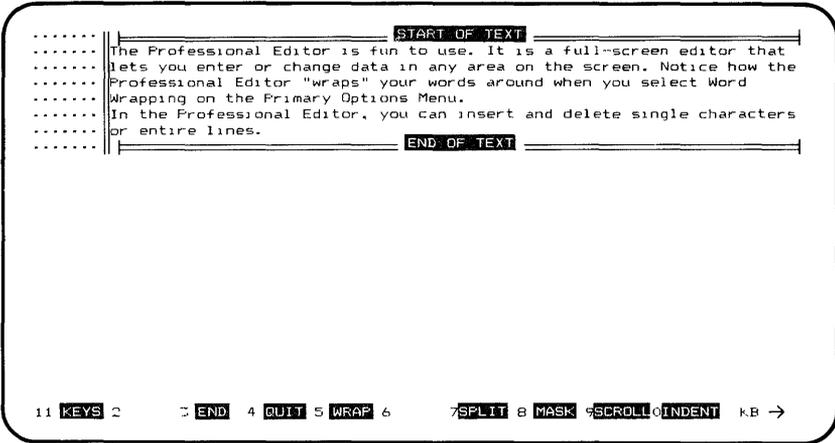
Deleting a Line

Let's try deleting a line. Move the cursor to anywhere within the first line.

Now press the DELETE function key (F6). The line disappears, and all lines that follow move upward on the screen.

Recovering Deleted Data

Suppose now that you decide you really do want that first line in your file. Leaving the cursor where it is, press the UNDELE function key (F16). (If you don't see key set 1 on your screen, continue to press the F1 function key until you see key set 1 in order to use the UNDELE function key F16.)



Pressing the UNDELE function key brings back a line that you have deleted by pressing the DELETE function key.

Note: The UNDELE function key works differently when you delete the *last* line of a file. Because you cannot enter a line command on the **END OF TEXT** line, you will need to position the cursor to another line before you press the UNDELE key. The undeleted line will appear above the current line. You can then move the line to its proper position. (See Lesson 5 for instructions on moving lines.)

Ending Lesson 3

If you made any typing mistakes when you first entered this file, you may want to go back and correct them now.

You have now finished Lesson 3. If you want to stop the lessons at this time, go back and read the instructions at the end of Lesson 2. Otherwise, go on to Lesson 4.

Lesson 4 – Using SEARCH and CHANGE

If You Stopped after Lesson 3

If you are continuing with Lesson 4 directly after Lesson 3, go to “Using the Command Line” below. If you are starting Lesson 4 after saving the EXAMPLE.TXT file in Lesson 3, perform the following steps:

1. Start up DOS, and then at the A> prompt, enter **EDIT** to start up the Professional Editor.
2. Type the name **EXAMPLE.TXT** for the INPUT FILE on the Primary Options Menu.
3. Press the Enter key.
4. Notice that the OUTPUT FILE and PROFILE FILE are filled in by the program.
5. Press the Enter key again to tell the Professional Editor that you want to use those filenames.
6. Now press the F9 function key to edit the file.
7. You should see the same file on your screen that you worked on in Lesson 3.

Using the Command Line

Press the SWAP function key (F2). This moves the cursor to line 24 on your screen, the command line. This is where you enter the editor commands listed in Chapter 7.

Pressing the SWAP key moves the cursor from the data area to the command line. Pressing it again moves the cursor back to its original position on the screen.

Using SEARCH and CHANGE Function Keys and Commands

With the cursor still on the command line, type the following, exactly as shown:

CHANGE /you/one/

Press the Enter key.

You just told the Professional Editor to change the letters **you** to the letters **one**.

Now, we want to start at the beginning of our file and change some of the occurrences of **you** to **one**. Press the SRCH F function key (F24) to locate the first occurrence of **you**. (If you don't see 21 in the bottom left corner on your screen, continue to press the F1 function key until you see 21 (key set 2) in order to use the SRCH F function key F24).


```
..... |-----| START OF TEXT |-----|
..... | The Professional Editor is fun to use. It is a full-screen editor that
..... | lets one enter or change data in any area on the screen. Notice how the
..... | Professional Editor "wraps" your words around when you select Word
..... | Wrapping on the Primary Options Menu.
..... | In the Professional Editor, one can insert and delete single characters
..... | or entire lines.
..... |-----| END OF TEXT |-----|

1 Changes were made.
```

Remember, once you have told the Professional Editor what to change, you can use the SRCH and CHNG function keys to selectively make changes.

Note: You can use a CHNG function key by itself (without a SRCH function key) if you know that you do want to change the next occurrence.

Ending Lesson 4

You have now finished Lesson 4. If you want to stop the lessons at this time, go back and read the instructions at the end of Lesson 2. Otherwise, go on to Lesson 5.

Lesson 5 – Using Line Commands

If You Stopped after Lesson 4

If you are continuing with Lesson 5 directly after Lesson 4, go to “Copying Lines” below. If you are starting Lesson 5 after saving the `EXAMPLE.TXT` file in Lesson 4, perform the following steps:

1. Start up DOS, and then at the `A>` prompt, enter `EDIT` to start up the Professional Editor.
2. Type the name `EXAMPLE.TXT` for the `INPUT FILE` on the Primary Options Menu.
3. Press the Enter key.
4. Notice that the `OUTPUT FILE` and `PROFILE FILE` are filled in by the program.
5. Press the Enter key again to tell the Professional Editor that you want to use those filenames.
6. Now press the `F9` function key to edit the file.
7. You should see the same file on your screen that you worked on in Lesson 4.

Copying Lines

Your cursor should still be in the data area on the screen. Move the cursor to the first line of your file and press the LINE function key (F3) in key set 0. This moves your cursor out to the lefthand column on the screen.

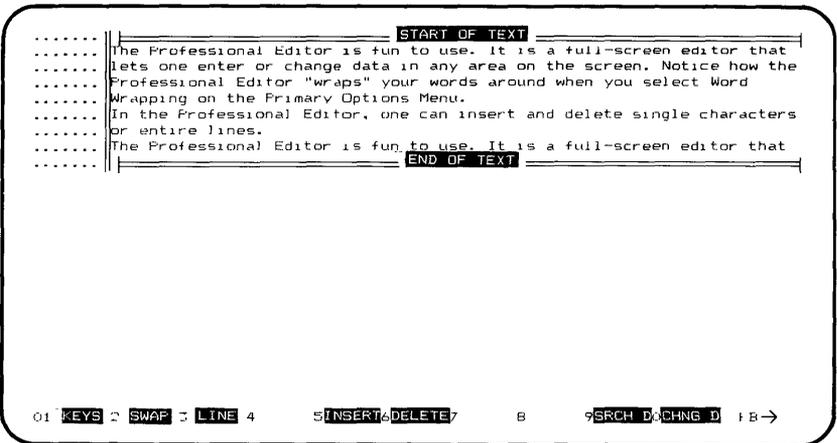
This is where you enter the line commands that are discussed in Chapter 6. What you enter in these lefthand columns is not treated as data in your file. Instead, it is treated as a command to tell the Professional Editor to do something to your file.

Type the letter C (or c). Then move the cursor to the last line of your file and type the letter A (or a).

Now, before you press the Enter key, let's look at what you did. You told the Professional Editor to *copy* the first line *after* the last line of the file.

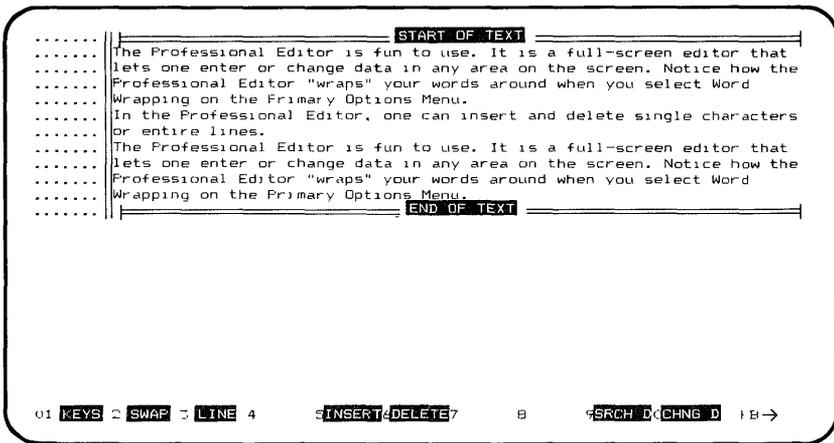
Now press Enter. Pressing Enter tells the Professional Editor to perform the command.

Look at your screen. The first line appears again at the end of the file.



Let's try that again, only this time let's copy three lines. Place the cursor at the left of the second line of the file, and type **CC**. Then move the cursor to the left of the fourth line and type **CC**. (The **CC**s tell the Professional Editor to copy all the lines from the first **CC** through the last **CC**.)

Now move the cursor to the last line of the file and type **A**. Press the Enter key. Your screen looks like this:



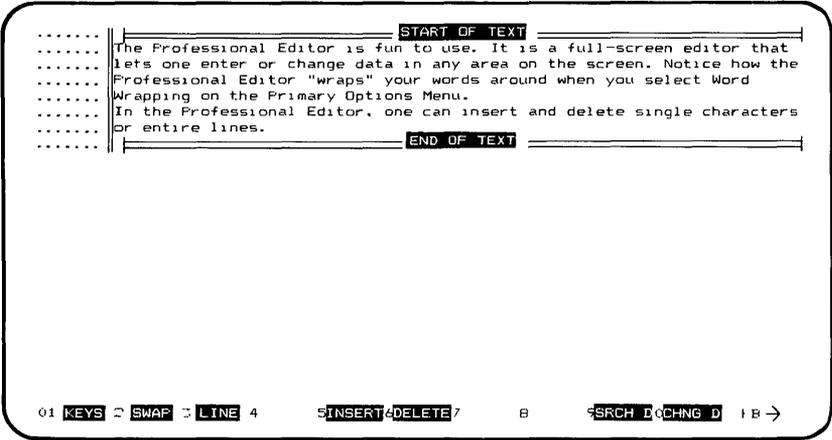
The screenshot shows a terminal window with a text editor. The text is wrapped. At the top, there is a line with "START OF TEXT" in a box. Below it are four lines of text: "The Professional Editor is fun to use. It is a full-screen editor that lets one enter or change data in any area on the screen. Notice how the Professional Editor 'wraps' your words around when you select Word Wrapping on the Primary Options Menu. In the Professional Editor, one can insert and delete single characters or entire lines. The Professional Editor is fun to use. It is a full-screen editor that lets one enter or change data in any area on the screen. Notice how the Professional Editor 'wraps' your words around when you select Word Wrapping on the Primary Options Menu." At the bottom, there is a line with "END OF TEXT" in a box. The status bar at the bottom shows "01 KEYS SWAP LINE 4 INSERT/DELETE 8 SRCH D CHNG D FB->".

Deleting Lines

Deleting lines is similar to copying lines, but you use the letter **D**, not **C**. Let's delete the four lines that we added to the end of our file.

Move the cursor to the last line of the file, and type **DD** in the line command area. Now move the cursor to the seventh line of the file, and type **DD**. (You can enter line commands from bottom-to-top or from top-to-bottom in the file, it doesn't make any difference.)

Press the Enter key. The four lines are gone.

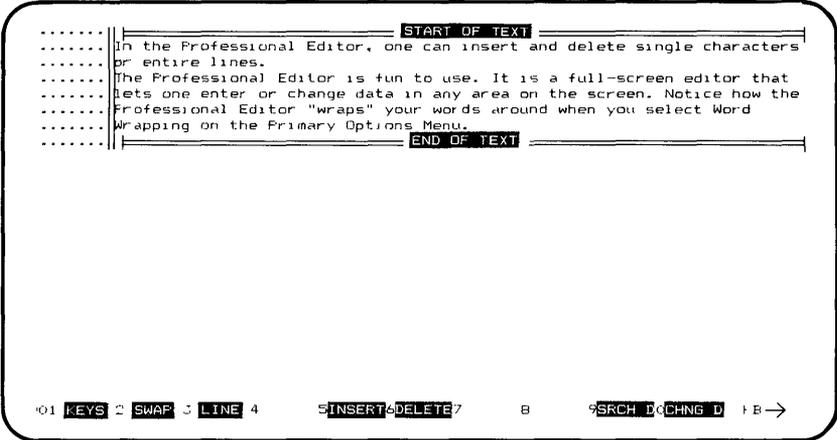


Moving Lines

Let's try one more line command. Move the cursor to the line command area of the next to the last line in the file. Type the letters **MM**. Then move the cursor to the last line in the file and type **MM**.

Now move the cursor to the first line, and type the letter **B**. This tells the Professional Editor to *move* the last two lines *before* the first line, so that they become the new first two lines of the file.

Press the Enter key, and your screen looks like this:



Ending the Lessons

We have made all of the changes that we are going to make to the EXAMPLE file at this time. If function key set 1 is not on line 25 of your screen, continue to press the F1 key until you see key set 1.

You see that F3 lets you end the session, so press the F3 key. The final version of the EXAMPLE.TXT file is now stored on your diskette. To return to DOS from the Primary Options Menu, press the F10 key.

We don't want you to think that you have learned all about the Professional Editor in these five short lessons. However, you have learned about many of the more frequently-used features of the program. You can start the Professional Editor, enter a new file, make changes to the file, save the file on diskette, and end your editing session.

With this background and more practice on your part, you should be well on your way to becoming productive with your full-screen editor. From now on, you will probably use the reference chapters of this book when you need to look up some information, but you can always refer to these lessons for additional examples.

REFERENCE SECTION

CHAPTER 3. GENERAL INFORMATION

Contents

Display Screen Operation	3-5
Screen Format	3-5
Current Line	3-6
Data Display Area (Lines 1 to 24)	3-6
Command Input Area and Status	
Display (Line 24)	3-7
Special Messages	3-8
Function Keys and Messages	
Line 25)	3-9
Cursor Display	3-10
Using Tab Characters	3-10
Defining Tab Stops	3-11
Automatic Indentation	3-12
Display of Right Margin	3-13
Audio Operation	3-14
Keyboard Operation	3-15
The Enter Key ()	3-16
Control and Numeric Keypad	
Usage	3-16
Normal Mode Operation	3-17
Control Mode Operation	3-18
Scroll Mode Operation	3-19
Numeric Mode Operation	3-20
Data Mode Operation	3-21

Cursor Operations	3-21
Cursor Up ()	3-22
Cursor Down ()	3-22
Cursor Left ()	3-23
Cursor Right ()	3-23
Cursor Word Left ( - )	3-23
Cursor Word Right ( - ) ...	3-24
Cursor to BOL ()	3-24
Cursor to EOL ( - )	3-24
Character Operations	3-24
Insert Character ()	3-25
Delete Character ()	3-25
Line Operations	3-25
Delete to EOL ()	3-25
Restore Field ( - )	3-25
Scrolling Operations	3-26
Scroll Up ()	3-26
Scroll Down ()	3-26
Scroll Left ()	3-26
Scroll Right ()	3-27
Scroll Top ()	3-27
Scroll Bottom ()	3-27
Save Scroll ()	3-27
Exchange Scroll ()	3-27
Keyboard Mode Selection	3-28
CAPS LOCK Mode	3-28
NUM LOCK Mode	3-29
SCROLL LOCK Mode	3-29
Program Function Keys	3-30
Key Set 0 Assignments	3-31
KEYS Function Key (F1, F11, F21, F31)	3-31
SWAP Function Key (F2)	3-31
LINE Function Key (F3)	3-31
RECALL Function Key (F4)	3-31
INSERT Function Key (F5)	3-32
DELETE Function Key (F6)	3-32
CANCEL Function Key (F7)	3-32
SRCH D Function Key (F9)	3-32
CHNG D Function Key (F10)	3-33

Key Set 1 Assignments	3-33
END Function Key (F13)	3-33
QUIT Function Key (F14)	3-33
WRAP Function Key (F15)	3-33
UNDELE Function Key (F16)	3-34
SPLIT Function Key (F17)	3-34
MASK Function Key (F18)	3-34
SCROLL Function Key (F19)	3-34
INDENT Function Key (F20)	3-35
Key Set 2 Assignments	3-35
SRCH U Function Key (F23)	3-35
SRCH F Function Key (F24)	3-35
SRCH D Function Key (F25)	3-36
SRCH L Function Key (F26)	3-36
CHNG U Function Key (F27)	3-36
CHNG F Function Key (F28)	3-37
CHNG D Function Key (F29)	3-37
CHNG L Function Key (F30)	3-37
Key Set 3 Assignments	3-38
PRINT Function Key (F32)	3-38
SPACE Function Key (F33)	3-38
EJECT Function Key (F34)	3-38
File Operations	3-39
File Format	3-39
Maximum Line Length	3-39
Character Set Restrictions	3-40
Multi-Segment Files	3-41
File Integrity	3-42
Cautions on File Specifications	3-43
Changing Diskettes	3-43

Display Screen Operation

The Professional Editor supports both the IBM Monochrome Display Adapter with the IBM Monochrome Display and the IBM Color Graphics Monitor Adapter attached to an RGB color monitor or black and white monitor. Either or both options may be installed and used during editing.

Only display modes with 25 lines and 80 columns are supported. The screen format is the same for all of these modes, except for the type of highlighting. When you use the monochrome display, intensification and reverse video are used for highlighting. When you use the color display, color and intensification are used for highlighting.

Screen Format

When you start the program, it presents a set of menus, from which you select the parameters and options for the editing session. Figure 1 shows what the display screen looks like during an editing session.

Line 24 contains the command input area and status display. Line 25 shows the function key labels. (On the actual screen, the labels appear in reverse video, making them easier to read.)

Lines 1-24 may also contain special messages (see “Special Messages” below), as well as other information you create by entering commands. While in command mode, line 24 contains special information (see below).

Command Input Area and Status Display (Line 24)

In command mode, line 24 contains the command input area and status display. The area follows the command prompt **COMMAND>** and extends to the end of the line.

Status information is displayed in the right half of the input area. You can type over the status information when you enter a command. The status display contains the following fields:

- **MD>** (automatic mode) – contains four indicators which, when present, show which of the following automatic functions are enabled:

W Automatic word wrapping

M Automatic input masking

S Automatic scrolling

I Automatic indentation

- **VS>** (vertical scroll) – indicates the number of lines to be scrolled when a scroll up or down operation is performed.
- **HS>** (horizontal scroll) – indicates the number of columns to be scrolled when a scroll left or right operation is performed.

- **BN>** (boundaries) - indicates the current left and right boundary columns which restrict the scope of **SEARCH** and **CHANGE** commands.
- **WI>** (window) - indicates the leftmost and rightmost columns in the current screen window.
- **SP>** (free space) - indicates the number of free bytes in the text buffer. If the free space becomes critically low, this number blinks to inform you that space must be freed before any more changes can be made. You can free space by writing out part of the file or deleting lines.

Special Messages

Three types of special messages can be displayed in the data area of the screen (lines 1-24), but they do not become part of the text file. These types of special messages are:

- **START OF TEXT** is displayed at the top of the screen to denote the beginning of the first line in the text buffer. You cannot delete this message by using the Delete Line function key (F6).
- **END OF TEXT** is displayed following the last line of text and denotes the end of the text buffer. You cannot delete this message by using the Delete Line function key (F6).
- **Miscellaneous messages** are displayed between lines of text in response to a command that you enter. You can delete these messages from the screen by using the Delete Line function key (F6). They are automatically deleted when scrolled off the screen.

Function Keys and Messages (Line 25)

Line 25 displays the function key labels, keyboard shift status, and messages. The first character in the line indicates the key set number currently selected. This number is intensified on your screen and represents the following sets of keys:

Key Set	Keys
0	Function keys 1-10
1	Function keys 11-20
2	Function keys 21-30
3	Function keys 31-40

Following the key set number, the key numbers and labels for all the function keys are displayed. The function key labels are displayed in reverse video. (Function keys that are not enabled are not displayed and are rejected by the program.)

The area to the extreme right shows the current keyboard mode:

Character	Active Keyboard Mode
KB->C	CAPS LOCK
KB->N	NUM LOCK
KB->S	SCROLL LOCK
KB->I	INSERT

When the Professional Editor displays an error message or informational message, it displays the message in line 25, where the function keys are normally displayed. The message remains on line 25 until you press a key. Then, the message is cleared, and the function keys are displayed again.

Cursor Display

The position of the cursor indicates where data entered from the keyboard is placed on the screen and whether it is interpreted as a command or as an update to the file.

- When the cursor is positioned in the data area of the screen, keyboard input updates the file being edited.
- When the cursor is positioned in the line command area (columns 1-8), data entered is processed as a line command.
- When the cursor is positioned in the command input area, data entered is processed as a command.
- When the Professional Editor is busy performing a task, the cursor is removed from the screen.

Using Tab Characters

Most micro-computer systems use tab characters in ASCII files to reduce the space required to store the files on diskette. The IBM Personal Computer hardware supports a fixed set of tab stops, with the stops being spaced every 8 columns beginning with column 1.

The Professional Editor permits you to specify tab stop settings to fit the requirements of the file being edited. Tab stops are completely variable and you can change them while editing. When file data is displayed on the screen, tab characters

within the file are expanded on the screen so that a true representation of the data is always displayed. The spaces between a tab character and the next tab stop are referred to as *tab spaces*. These appear as actual spaces on the screen. These spaces cannot be modified on the screen, and the cursor always skips over them. They belong to the preceding tab character and are adjusted dynamically as the line is modified.

In lines of data, tab characters are always expanded as described above. In most menu fields, in line command fields, and in command lines, tabs are not expanded and are displayed as the corresponding graphic character (circle). Tab characters in these fields are not generally valid, except in SEARCH and CHANGE commands.

Defining Tab Stops

The Professional Editor has two methods for defining and changing tab stop settings. First, you may define tab stop settings during initialization by pressing the TABS function key (F5) and then setting the tab stops you want from a menu. The Professional Editor initializes these settings to match the hardware fixed tab settings. You may change these tab settings as desired, and the new values will be stored as part of the profile file (described in Chapter 4). This allows different tab settings to be maintained for different types of data and always to be set when you edit each type of data.

The second method allows tab settings to be changed during the editing session by using the TABS command. Tab settings made with the TABS command are *not* stored in the profile and are lost when the session is terminated.

Automatic Indentation

Automatic indentation is a useful feature when the source statements for a structured program are being entered or altered. The automatic indentation feature operates in edit and input modes in a slightly different way. (See Chapter 5 for a complete description of edit and input modes.)

In input mode, where new lines are being added, automatic indentation sets the cursor to the same column as the first non-blank character in the preceding line. Of course, the cursor may be moved as desired before data is entered.

In edit mode, the cursor is set to the first non-blank character in the line being edited. This avoids having to manually move the cursor when structured data is being edited. Automatic indentation occurs *only* when you press the Enter key to end editing of the previous line. It does not occur if you use the Cursor Up or Cursor Down key to select the next line.

In either mode, a non-blank character is a character which is not an ASCII space or tab character and is not within the space created by the expansion of a tab.

Activating Indentation

Automatic indentation may be selected as an option on the Primary Options Menu, where it becomes part of the profile file. It may be turned on and off during editing by pressing the INDENT function key (F20).

Display of Right Margin

When a right margin column has been defined, the column is highlighted on the screen using reverse video on the IBM Monochrome Display and reverse color on a color display.

Audio Operation

The Professional Editor generates audio tones using the built-in speaker of the IBM Personal Computer to alert you to various conditions during editing. Different conditions are indicated by the use of combinations of two tone frequencies and two tone durations:

- Positive acknowledgements inform you that a requested action has been completed correctly when there would otherwise be no visual result of the action (such as the screen being updated). A positive acknowledgement consists of a high pitched tone of short duration.
- Negative acknowledgements inform you that a requested action is invalid or cannot be performed. It is used instead of an error message to indicate an error (such as pressing a function key that is presently undefined). A negative acknowledgement consists of a low pitched tone of short duration.
- A message tone is issued when an informational message is displayed on the screen. It consists of a high pitched tone of medium duration.
- An error message tone is issued when an error message is displayed on the screen. It consists of a low pitched tone of medium duration.

You can select the level of audio prompting you wish to use on the Profile Options Menu (see Chapter 4), where it becomes part of the profile, or during the edit session with the SET AUDIO command (see Chapter 7).

Keyboard Operation

The IBM Personal Computer keyboard contains the following three logical groups of keys:

- The main keypad contains a standard set of ASCII character keys with control keys at the left and right edges.
- The program function keys are located at the left of the main keypad and are labeled F1, F2, ... F10.
- The control and numeric keypad is located at the right of the main keypad. It is used for control functions.

The Professional Editor can process all ASCII numeric, alphabetic, and special characters on the keyboard. ASCII control characters (Ctrl-A, Ctrl-B, ...) may be used when they do not conflict with the restrictions previously outlined. These characters are displayed as special graphic characters using the IBM Personal Computer extended ASCII character set. (See the *Disk Operating System (DOS)* manual for a complete list.)

You can input the special graphic characters (decimal codes 128-254) by holding down the Alt key while typing the 3-digit decimal code on the keypad. The Backspace key is used as a “rubout” key, deleting the current character and backspacing the cursor.

The Caps Lock, Num Lock, and Scroll Lock keys toggle the respecting keyboard modes on and off.

Holding a key down firmly causes the key to be entered in rapid succession. This may be used with almost any keyboard key to achieve “typamatic” operation. It is particularly useful when used with the cursor control keys for fast cursor positioning.

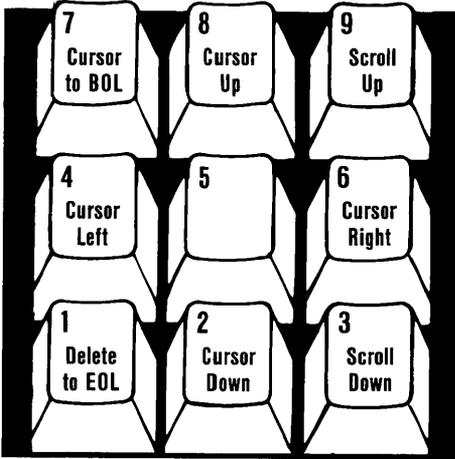
The Enter Key ()

The Enter key is the large key located at the right end of the main keypad and is labeled with the Carriage Return/Line Feed symbol. You use this key to inform the Professional Editor that you have completed editing a line of data or a data field of a menu. When you press Enter, the Professional Editor processes the data and proceeds to the next line or field. The cursor is always returned to the left edge of the screen or the left-most character of the data field.

Control and Numeric Keypad Usage

The control and numeric keypad lets you perform frequently-used functions such as cursor positioning, scrolling, and more. Where possible, the function of a key corresponds to the key label. The five modes of operation for the control and numeric keypad are: normal, control, scroll, numeric, and data.

Normal Mode Operation

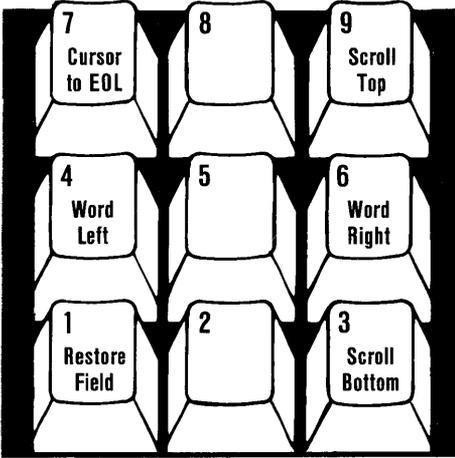


In normal mode,

- Num Lock is toggled OFF
- Scroll Lock is toggled OFF
- The Ctrl and Alt keys are not held down

You use this mode for frequently-used editing functions.

Control Mode Operation

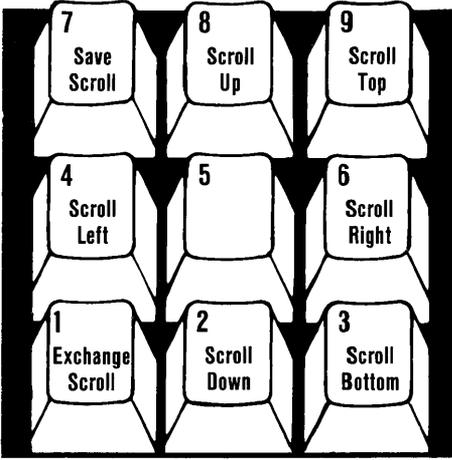


In control mode,

- Num Lock is toggled OFF
- Scroll Lock is toggled OFF
- The Ctrl key is held down
- The Alt key is not held down

You use this mode for special editing operations.

Scroll Mode Operation

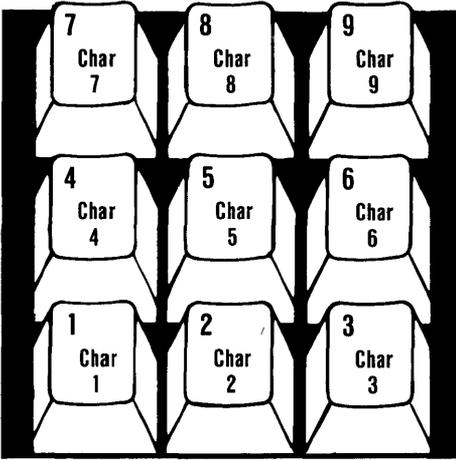


In scroll mode,

- Num Lock is toggled OFF
- Scroll Lock is toggled ON
- The Ctrl and Alt keys are not held down

You use this mode for scrolling operations.

Numeric Mode Operation

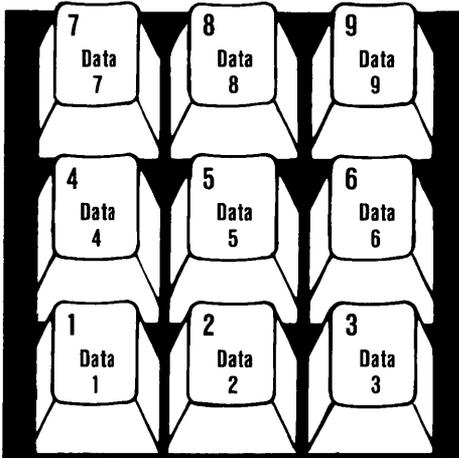


In numeric mode,

- Num Lock is toggled ON
- Scroll Lock is toggled ON or OFF
- The Ctrl and Alt keys are not held down

You use this mode for entering large quantities of numeric data.

Data Mode Operation



In data mode,

- Num Lock is toggled OFF
- Scroll Lock is toggled OFF
- The Ctrl key is not held down
- The Alt key is held down

You use this mode to enter special extended ASCII characters. The decimal number typed on the keypad is interpreted as an ASCII character code.

Cursor Operations

You use the various cursor operations in normal mode to move the cursor over the data on the screen. These operations do not alter the data on the screen in any way.

Cursor Up ()

The Cursor Up key moves the cursor upward on the screen one line to the previous line. If the cursor is moved off the top of the screen, and AUTO SCROLL mode is not selected, the cursor is placed on the last line of the screen. If AUTO SCROLL is selected, a scroll up operation is performed automatically. The same horizontal position of the cursor is maintained, unless the cursor would be positioned in the space created by a tab character. In this case, the cursor is repositioned over the tab character. If the cursor is thus repositioned, the Professional Editor remembers the cursor's original position if another Cursor Up or Cursor Down is the *next* key pressed.

Cursor Down ()

The Cursor Down key moves the cursor downward on the screen one line to the next line. If the cursor is moved off the bottom of the screen, and AUTO SCROLL is not selected, the cursor is placed on the first line of the screen. If AUTO SCROLL is selected, a scroll down operation is performed automatically. The same horizontal position of the cursor is maintained, unless the cursor would be positioned in the space created by a tab character. In this case, the cursor is positioned over the tab character. If the cursor is thus repositioned, the Professional Editor remembers the cursor's original position if another Cursor Up or Cursor Down is the *next* key pressed.

Cursor Left ()

The Cursor Left key moves the cursor one position left from its current position toward the beginning of the line. If the cursor is already positioned at the beginning of the line, the cursor is positioned at the start of the line command area. If AUTO SCROLL is selected, a scroll left operation is performed automatically if the cursor is at the left edge of the screen but not at the beginning of the line. If the cursor moves into a tab space, the cursor is positioned leftward to the tab character.

Cursor Right ()

The Cursor Right key moves the cursor one position right from its current position toward the end of the line. If the cursor is already positioned at the rightmost character on the screen, and AUTO SCROLL is not selected, the cursor remains there. If AUTO SCROLL is selected, a scroll right operation is performed automatically. If the cursor moves into a tab space, the cursor is positioned rightward to the next tab stop.

Cursor Word Left (-)

The Cursor Word Left key moves the cursor left to the beginning of the current word. If the cursor is positioned between words, it is moved to the beginning of the previous word. If AUTO SCROLL is selected, automatic scrolling occurs if the cursor moves off the left edge of the screen.

Note: A *word* is a consecutive group of alphabetic characters (A-Z and a-z) and/or numeric characters (0-9). Special characters and spaces are treated as delimiters which separate words.

Cursor Word Right (-)

The Cursor Word Right key moves the cursor right to the beginning of the next word. If the end of the line is encountered within the current window, the cursor is placed following the last word of the line to allow the line to be extended. If AUTO SCROLL is selected, automatic scrolling occurs if the cursor moves off the right edge of the screen.

Cursor to BOL ()

The Cursor to BOL key causes the cursor to be positioned to the beginning of the current line or left edge of the screen. Automatic scrolling does not occur with this function when the AUTO SCROLL is off.

Cursor to EOL (-)

The Cursor to EOL key causes the cursor to be positioned to the end of the current line or right edge of the screen. Automatic scrolling does not occur with this function. If AUTO SCROLL is off and if you are editing a menu field, the cursor skips to the next field to the right.

Character Operations

You use the character operations to insert and delete individual characters within a line of text. You can use these operations with command lines and menu fields as well as data lines.

Insert Character ()

Pressing the Ins key places the program in *insert character* mode. In this mode, characters that you enter are placed before the character at the cursor, shifting the remainder of the line to the right. Proper tab spacing is maintained at all times. When the Ins key is pressed again, insert character mode is terminated. While insert character mode is active, the character I appears in the keyboard status display.

Delete Character ()

The Del key deletes the character over the cursor, and shifts the remainder of the line to the left. Proper tab spacing is maintained at all times.

Line Operations

You use the line operations to insert and delete lines of text.

Delete to EOL ()

The Delete to EOL key deletes all characters from the cursor position to the end of the current line.

Restore Field (-)

The Restore Field key may be used to restore the previous contents of a menu data field in the TABS and on the Profile Options Menu, or to restore the line being edited from the text buffer.

Scrolling Operations

You use the scrolling operations in normal and scroll mode to position data within the file being edited so that the area of interest is displayed on the screen.

Scroll Up ()

The Scroll Up key moves the screen window backward toward the beginning of the file in normal mode. The number of lines moved depends on the current scroll increment. The default scroll increment is 12 lines (half screen). In scroll mode, it moves the screen window to the top of the file.

Scroll Down ()

The Scroll Down key moves the screen window forward toward the end of the file in normal mode. The number of lines moved depends on the current increment count. The default scroll increment is 12 lines (half screen). In scroll mode, it moves the screen window to the bottom of the file.

Scroll Left ()

The Scroll Left key moves the screen window to the left toward the first column of the line. The number of columns moved depends on the current increment. The default scroll increment is 20 columns (quarter screen).

Scroll Right ()

The Scroll Right key moves the screen window to the right toward the last column of the line. The number of columns moved depends on the current increment. The default scroll increment is 20 columns (quarter screen).

Scroll Top ()

The Scroll Top key positions the file to the beginning of text (or current segment).

Scroll Bottom ()

The Scroll Bottom key positions the file to the end of text (or current segment).

Save Scroll ()

The Save Scroll key saves the current position within the file being edited for use later with the Exchange Scroll key.

Exchange Scroll ()

The Exchange Scroll key saves the current position within the file and scrolls to the position previously saved when you pressed the Save Scroll key.

You use the Save Scroll key and Exchange Scroll key together. For example, you might want to perform the following steps to move the cursor:

1. Position the cursor to the first line of data.
2. Press the Scroll Lock key (to turn SCROLL LOCK mode on).
3. Press the Save Scroll key. This saves the current position.
4. Press Scroll Lock again (to turn SCROLL LOCK mode off and return to NORMAL mode).
5. Then position the cursor to another line.
6. Press the Scroll Lock key (to turn SCROLL LOCK mode on).
7. Press the Exchange Scroll key. This saves the exchange position and returns the cursor to the position previously saved with the Save Scroll key.
8. Press the Exchange Scroll key again to return to the line in step 5.

Keyboard Mode Selection

CAPS LOCK Mode

The Caps Lock key toggles CAPS LOCK mode on and off. When CAPS LOCK mode is on, all alphabetic characters are translated to uppercase when you enter them from the keyboard. Special and numeric characters are not affected. When CAPS LOCK mode is on, the character **C** appears in the keyboard status display.

NUM LOCK Mode

The Num Lock key toggles NUM LOCK mode on and off. When NUM LOCK mode is on, the keypad keys enter numeric characters, instead of control functions, when pressed. When NUM LOCK mode is on, the character **N** appears in the keyboard status display.

SCROLL LOCK Mode

The Scroll Lock key toggles SCROLL LOCK mode on and off. When SCROLL LOCK mode is on, the keypad keys perform scrolling operations in place of their standard functions. When SCROLL LOCK mode is on, the character **S** appears in the keyboard status display.

You can use one, two, or all three keyboard modes at one time. For example, the following could appear in the keyboard status display:

KB->CNS

One other character can appear after the keyboard status display. When you press the Ins key to insert characters, the letter **I** appears. You cannot use the Ins or Del key in NUMLOCK mode.

Program Function Keys

The function keys are located in the keypad at the left edge of the keyboard. You see only ten function keys on your keyboard, but the use of *key sets* permits these ten keys to be used in four different combinations. These key sets consist of the following groups of ten keys each:

Key Set	Function Keys
0	Function keys F1-F10
1	Function keys F11-F20
2	Function keys F21-F30
3	Function keys F31-F40

Line 25 displays the number of the currently selected key set at the left edge of the screen followed by the numbers and labels for each of the ten keys. The label for a given key indicates the function to be performed when that key is pressed. If no label is present, the corresponding key is either not defined or its use is currently prohibited.

Press F1 to select the desired key set. Each time you press the F1 key, you select the next key set, and the labels are displayed (in the succession Set 0, 1, 2, 3, 0, 1 ...). Once selected, the key set remains active until you select another key set by pressing F1.

The following paragraphs describe the key functions assigned during the editing session. The key assignments used during initialization are described in the section, "Initialization Function Key Usage."

Key Set 0 Assignments

F1 KEYS	F2 SWAP	F3 LINE	F4 RECALL	F5 INSERT	F6 DELETE	F7 CANCEL	F8	F9 SRCH D	F10 CHNG D
--------------------------	--------------------------	--------------------------	----------------------------	----------------------------	----------------------------	----------------------------	-----------	----------------------------	-----------------------------

Figure 2. Function Key Set 0 Assignments

KEYS Function Key (F1, F11, F21, F31)

You press the KEYS function key to select the desired key set. Each time you press this key, the next key set is selected and displayed in line 25. The KEYS function key performs the same function in all four key sets. After key set 3, key set 0 is selected when you press the F1 key again.

SWAP Function Key (F2)

The SWAP function key switches the cursor from the current line in the data area of the screen to the command input area. When you press it again, the cursor is returned to the current line from the command input area.

LINE Function Key (F3)

The LINE function key moves the cursor from its current position in the data area or command input area to the line command area for the current line. When pressed again, the cursor is returned to the data area for editing data.

RECALL Function Key (F4)

The RECALL function key is used to retrieve a previously entered command from the command stack. Its use is described more fully in “Using the Command Stack,” in Chapter 7.

INSERT Function Key (F5)

The INSERT function key places the Professional Editor in *input mode* and allows new lines to be inserted after the current line. When you press F5, a blank line is created after the current line and you may enter new lines as required. Input mode is terminated when a null (empty) line is entered. (See Chapter 5 for a complete description of input mode.)

DELETE Function Key (F6)

The DELETE function key deletes the current line from the screen and file. Data on the screen is scrolled upward to fill the deleted line, and the next line of data (if any) moves upward onto the screen.

If a line is accidentally deleted, you may restore it by using the UNDELE function key.

CANCEL Function Key (F7)

The CANCEL function key cancels a pending line command. Its use is described in “Cancelling an Incomplete (Pending) Line Command,” in Chapter 6.

SRCH D Function Key (F9)

The SRCH D function key searches downward from the current position toward the end of the file (or segment). The search argument must have been previously defined by a SEARCH or CHANGE command. See the description of the SEARCH and CHANGE commands in Chapter 7 for additional information.

CHNG D Function Key (F10)

The CHNG D function key searches downward from the current position toward the end of the file (or segment). If the search argument is found, it is replaced by the change argument. Both the search and change arguments must have been previously defined by a CHANGE command. See the description of the CHANGE command in Chapter 7 for additional information.

Key Set 1 Assignments

F1 KEYS	F2	F3 END	F4 QUIT	F5 WRAP	F6 UNDELE	F7 SPLIT	F8 MASK	F9 SCROLL	F10 INDENT
------------	----	-----------	------------	------------	--------------	-------------	------------	--------------	---------------

Figure 3. Function Key Set 1 Assignments

END Function Key (F13)

The END function key saves the file being edited and terminates the session. For additional information, see “Ending the Edit Session,” in Chapter 5.

QUIT Function Key (F14)

The QUIT function key terminates the session without saving the file being edited. For additional information, see “Ending the Edit Session,” in Chapter 5.

WRAP Function Key (F15)

The WRAP function key turns automatic word wrapping on and off. This feature is described in “Using Automatic Word Wrapping,” in Chapter 9.

UNDELE Function Key (F16)

The UNDELE function key recovers lines (undeletes) which have been accidentally deleted with the DELETE function key. For additional information, see “Recovering Deleted Lines,” in Chapter 5.

SPLIT Function Key (F17)

The SPLIT function key splits a line into two lines at a given point. See “Splitting a Line into Multiple Lines,” in Chapter 5 for a description of this function.

MASK Function Key (F18)

The MASK function key turns on or off the use of the input mask when new lines are being created in input mode. Pressing this key switches the mask function from on to off or from off to on. For additional information, see “Input Mask Data,” in Chapter 4.

SCROLL Function Key (F19)

The SCROLL function key turns automatic scrolling on or off. Automatic scrolling occurs when the cursor is moved off the screen at the top, bottom, left, or right edge. For additional information, see “Automatic Scrolling,” in Chapter 5.

INDENT Function Key (F20)

The INDENT function key turns on or off automatic line indentation. When in edit mode, automatic indentation places the cursor on the first non-blank character of the next line when you press the Enter key on the previous line. In input mode, the cursor is placed in the same column as the first non-blank character in the previous line. This feature is useful for creating and maintaining structured code. For additional information, see “Auto Indent,” in Chapter 4.

Key Set 2 Assignments

F1 KEYS	F2	F3 SRCH U	F4 SRCH F	F5 SRCH D	F6 SRCH L	F7 CHNG U	F8 CHNG F	F9 CHNG D	F10 CHNG L
------------	----	--------------	--------------	--------------	--------------	--------------	--------------	--------------	---------------

Figure 4. Function Key Set 2 Assignments

SRCH U Function Key (F23)

The SRCH U function key searches *upward* from the current position toward the beginning of the file (or segment). The search argument must have been previously defined by a SEARCH command. See the description of the SEARCH command in Chapter 7 for additional information.

SRCH F Function Key (F24)

The SRCH F function key searches downward from the top of the text buffer toward the end of the file (or segment) for the *first* occurrence of the search argument. The search argument must have been previously defined by a SEARCH command. See the description of the SEARCH command in Chapter 7 for additional information.

SRCH D Function Key (F25)

The SRCH D function key searches *downward* from the current position toward the end of the file (or segment). The search argument must have been previously defined by a SEARCH command. See the description of the SEARCH command in Chapter 7 for additional information.

SRCH L Function Key (F26)

The SRCH L function key searches upward from the bottom of the text buffer toward the beginning of the file (or segment) for the *last* occurrence of the search argument. The search argument must have been previously defined by a SEARCH command. See the description of the SEARCH command in Chapter 7 for additional information.

CHNG U Function Key (F27)

The CHNG U function key searches *upward* from the current position toward the beginning of the file (or segment). If the search argument is found, it is replaced by the change argument. Both the search and change arguments must have been previously defined by a CHANGE command. See the description of the CHANGE command in Chapter 7 for additional information.

CHNG F Function Key (F28)

The CHNG F function key searches downward from the top of the text buffer toward the end of the file (or segment) for the *first* occurrence of the search argument. If the search argument is found, it is replaced by the change argument. Both the search and change arguments must have been previously defined by a CHANGE command. See the description of the CHANGE command in Chapter 7 for additional information.

CHNG D Function Key (F29)

The CHNG D function key searches *downward* from the current position toward the end of the file (or segment). If the search argument is found, it is replaced by the change argument. Both the search and change arguments must have been previously defined by a CHANGE command. See the description of the CHANGE command in Chapter 7 for additional information.

CHNG L Function Key (F30)

The CHNG L function key searches upward from the bottom of the text buffer toward the beginning of the file (or segment) for the *last* occurrence of the search argument. If the search argument is found, it is replaced by the change argument. Both the search and change arguments must have been previously defined by a CHANGE command. See the description of the CHANGE command in Chapter 7 for additional information.

Key Set 3 Assignments

F1 KEYS	F2 PRINT	F3 SPACE	F4 EJECT	F5	F6	F7	F8	F9	F10
--------------------------	---------------------------	---------------------------	---------------------------	-----------	-----------	-----------	-----------	-----------	------------

Figure 5. Function Key Set 3 Assignments

PRINT Function Key (F32)

The PRINT function key prints all text lines currently on the screen on the line printer. *Only* data lines are printed; special messages and status data are omitted.

SPACE Function Key (F33)

The SPACE function key causes the line printer to space forward one line.

EJECT Function Key (F34)

The EJECT function key causes the line printer to space forward to the top of the next page.

File Operations

All files to be processed by the Professional Editor must reside on diskettes. DOS file routines are used exclusively to read and write files resident on diskettes.

File Format

Files to be processed by the Professional Editor must be in standard DOS format. DOS requires that the ASCII Carriage Return Line Feed sequence (CR-LF) be used to denote the end of a line. DOS files contain lines of variable length which end with the ASCII CR-LF end of line sequence. The Professional Editor pads the last sector of a file with End of File (HEX 1A) characters.

Maximum Line Length

Files input to the Professional Editor may contain lines of up to 140 bytes in length or, if tabs are used, 140 columns of data. By using horizontal scrolling, all 140 bytes or columns are accessible on the screen. Lines which exceed the 140 byte or column restriction are split into multiple lines.

Character Set Restrictions

Files to be processed by the Professional Editor may contain ASCII and extended ASCII characters, subject to the restrictions listed in Figure 6.

Extended ASCII characters may be entered from the keyboard using ALT mode. Refer to your *Disk Operating System (DOS)* manual for a complete list of ASCII characters.

Character	ASCII	HEX	Restriction
Null	NUL	00	Used internally. Will be deleted from file. May not be entered from keyboard.
Backspace	BS	08	May appear in file. If entered from the keyboard, it will be interpreted as RUBOUT.
Tab	HT	09	Will always be interpreted as horizontal tab.
Line Feed	LF	0A	May appear in file ONLY in the end of line sequence CR-LF. May not be entered from the keyboard.
Carriage Return	CR	0D	May appear in file ONLY in the end of line sequence CR-LF. May not be entered from the keyboard.
End File	SUB	1A	Used by DOS to denote end of file. May not be used for any other purpose. May not be entered from the keyboard.
Delete	DEL	7F	May appear in file but may not be entered from keyboard.
	---	FF	Used internally. Will be deleted from file. May not be entered from keyboard.

Figure 6. Restricted and Extended ASCII Characters

Multi-Segment Files

The Professional Editor allocates a working text buffer from available memory left over after the program is loaded. The text buffer may be as large as 64K (65536) bytes. Files which are too large to be contained in the text buffer are processed sequentially in segments. When the editing session begins, file data is read into the text buffer until the end of the file is reached or the buffer reserve is exceeded. The *buffer reserve* is the amount of buffer space reserved for changes to the file during editing. If the entire file is contained in the buffer, editing proceeds normally and all lines of the file are available for editing. If the entire file does not fit, then the following procedure is used to edit the file in segments:

- You edit the first segment of the file making all required changes.
- You then use the NEXT command to write the first segment to the output file and read the next segment from the input file into the text buffer.
- If more control is needed over the amount of data read into or written from the buffer, you may use the PUT command to write a selected portion of the data in the buffer to the output file. You may then use the GET command to read a selected number of lines into the text buffer.
- Editing continues in this manner, segment by segment, until completed.

- You enter the END command when editing is complete to transfer the remainder of the file to the output file and terminate the editing session.
- To return to the beginning of the file, you must issue the END command and then re-edit the same file.

File Integrity

To preserve the integrity of the file being processed, the Professional Editor creates a temporary file for the output file. When the file is closed, the original file is deleted, and the output file is saved with the filename of the original file.

If you end the edit session with the QUIT command, if the Professional Editor ends abnormally, or if the system crashes, the output file is lost, but the original input file is left intact.

Because a second file is opened for output, you must ensure that the diskette has enough free space to contain another copy of the original file plus any additions to be made during the editing session. If the free sectors on the diskette are exhausted during the editing session, you will be requested to insert another diskette in the drive or select another drive to contain the remainder of the file. This procedure is described in “Recovering from Disk Full Conditions,” in Chapter 10.

If diskette space is a problem, the Edit-in-Place option may be used. This option allows the file to be updated directly without creating a temporary output file. Since the original file may be updated during the editing session, it may not be possible to QUIT the session if a disastrous error is made.

Cautions on File Specifications

The Professional Editor uses DOS routines to check file specifications entered in menu fields and commands. Two errors which may be present in a file specification are not detected:

- If the filename portion of the file specification contains more than eight characters, the excess characters are ignored.
- If the extension portion of the file specification contains more than three characters, the excess characters are ignored.

When you enter a file specification, be careful that the name and extension portions do not exceed eight and three characters, respectively. Otherwise, the wrong file could be over-written or erased.

Changing Diskettes

The *Disk Operating System (DOS)* manual does not provide information on when it is safe or unsafe to change diskettes in a drive. A few words on this subject are in order to prevent the accidental destruction of the diskette directory or files.

To prevent accidental destruction of data, we have made every attempt to insure that files are not left open unless there is I/O activity on the file. Since this is not possible when using the Edit in Place option, we strongly recommend that you never switch out the diskette containing the file being edited when using this option.

The following notes explain when files are opened and closed:

Notes:

1. The input and output files are not opened until you begin the editing session by pressing the EDIT (F9) function key.
2. When the Edit in Place option is in effect, the file selected for editing is opened when the EDIT (F9) function key is pressed and remains open until you end the session.
3. If Read Input = YES, as much of the file as will fit into the text buffer is read, and the input file is closed. If Read Input = NO, the input file is immediately closed. The output file is opened and immediately closed.
4. For GET, PUT, NEXT and END commands, the appropriate files are opened for the duration of the I/O operations only, then closed.

5. For extract operations, the extract file is opened and closed by the EOPEN command. It is then opened and closed for each extract operation. The ECLOSE or END command opens the extract file, writes the end of file mark, and closes it.
6. For merge operations, the merge file is opened when the MERGE command is issued and closed when all data has been read into the text buffer.
7. If a required file is not in the correct drive when a reopen for I/O is attempted, you are prompted with an error message allowing you to retry the operation or process the error.
8. If you run out of space on the diskette which contains the output file, the Professional Editor assists you in changing diskettes by providing an options menu.

CHAPTER 4. INITIALIZATION FUNCTION KEY USAGE AND MENUS

Contents

Initialization Function Key Usage	4-3
KEYS Function Key (F1)	4-3
PRIME Function Key (F3)	4-3
OPTION Function Key (F4)	4-4
TABS Function Key (F5)	4-4
MACROS Function Key (F6)	4-4
FILES Function Key (F7)	4-4
RESETP Function Key (F8)	4-4
EDIT Function Key (F9)	4-4
RETURN Function Key (F10)	4-4
Entering Data on the Menus	4-5
Initialization Menus	4-6
Primary Options Menu	4-7
Selecting the Input File	4-8
Selecting the Output File	4-9
Selecting the Profile File	4-10
Selecting the List Files	4-11
Selecting the Display Mode	4-11
Selecting the Default Disk	4-12
Text Buffer Size	4-12

Profile Options Menu	4-13
Selecting Options	4-14
Read Only (NO)	4-14
Edit in Place (NO)	4-14
Backup File (NO)	4-15
Read Input (YES)	4-16
Re-Cycle (YES)	4-16
Left Margin (0)	4-16
Right Margin (0)	4-17
Paragraph Indentation (0)	4-17
Uppercase (NO)	4-17
Extended ASCII (YES)	4-18
Auto Indent (NO)	4-18
Mask Input (NO)	4-18
Auto Scroll (NO)	4-19
Word Wrapping (NO)	4-19
Audio Level (2)	4-20
Write Verify (0)	4-20
Vertical Scroll Increment (12)	4-21
Horizontal Scroll Increment (20)	4-21
Buffer Reserve (4)	4-21
Bounds (Left=1 Right=140)	4-21
Printer Unit (0)	4-22
Characters per Line (80)	4-22
Lines per Inch (6)	4-22
 Tabs and Mask Options Menu	 4-23
Tab Settings	4-24
Input Mask Data	4-24
 Macro Definition Menu	 4-25
 File List Menu	 4-26
Displaying the File List Menu	4-26

This chapter describes how to specify the files to be edited and the options to be used. It also describes the function keys used during initialization and the use of menus.

You start editor initialization when you first start the program or when you stop editing by using the END or QUIT function key or command.

Initialization Function Key Usage

Figure 7 shows the assignment of the function keys during initialization. Key sets 1, 2, and 3 are not used in initialization.

F1 KEYS	F2	F3 PRIME	F4 OPTION	F5 TABS	F6 MACROS	F7 FILES	F8 RESETP	F9 EDIT	F10 RETURN
--------------------------	-----------	---------------------------	----------------------------	--------------------------	----------------------------	---------------------------	----------------------------	--------------------------	-----------------------------

Figure 7. Function Key Set 0 in Initialization

KEYS Function Key (F1)

Press the KEYS function key here as elsewhere to select the active key set.

PRIME Function Key (F3)

The PRIME function key is enabled while the Profile Options, Macro Definition, File List, or Tabs and Mask Options Menu is being displayed and may be pressed to return to the Primary Options Menu.

OPTION Function Key (F4)

Press the OPTION function key to display the Profile Options Menu.

TABS Function Key (F5)

Press the TABS function key to display the Tabs and Mask Options Menu.

MACROS Function Key (F6)

Press the MACROS function key to display the Macro Definitions Menu. Refer to Chapter 8, "Using Macros" for a complete description.

FILES Function Key (F7)

Press the FILES function key to display the List Files Menu.

RESETP Function Key (F8)

Press the RESETP function key to reset all profile options (including data macros, tabs, and input mask) to the initialization default values.

EDIT Function Key (F9)

Press the EDIT function key to begin editing after you have selected all the options.

RETURN Function Key (F10)

Press the RETURN function key to end the program and return control to DOS.

Entering Data on the Menu

You enter data into the fields of the menus by using the normal keyboard keys. The cursor control keys (Up, Down, Left, and Right) are used to position the cursor to the next option field on the menu or within a given field. When data is being entered into a field, the Cursor Left, Cursor Right, Ins, Del, and Delete to EOL keys may be used to move the cursor, to insert and delete characters, and to delete the entire field or to the end of the field. The “rubout” (Backspace) key may also be used. If you wish to skip over the current field to the next field to the right, press the Cursor to EOL key. You may use the Restore Field key to restore the previous value of the menu options.

Once data for the field has been entered, press the Enter key to allow the input to be processed. If an error is detected, an error message is displayed in the message area, and the cursor is positioned at the beginning of the field to allow correction. When the data for the field has been accepted, the cursor is positioned to the next field on the menu. To skip over a field, press the Enter key or use the cursor positioning keys to position to the desired field.

Initialization Menus

Initialization uses the following menus to obtain the filenames and options to be used in the edit session:

- The Primary Options Menu is displayed first and is used to select the names for the input, output, and profile files to be used. It also allows some of the editor options to be selected.
- The Profile Options Menu is used to select editor options which are stored in the profile file.
- The Tabs and Mask Options Menu is used to select tab settings and to define the input mask to be used.
- The Data Macro Definition Menu is used to define editor data macros to be used during the edit session. (Not available in the minimum version.)
- The File List Menu is used to display a list of diskette files as specified by you. (Not available in the minimum version.)

Primary Options Menu

The following Primary Options Menu is displayed as soon as the Professional Editor is loaded (after the IBM logo screen), or after you end or quit a previous session:

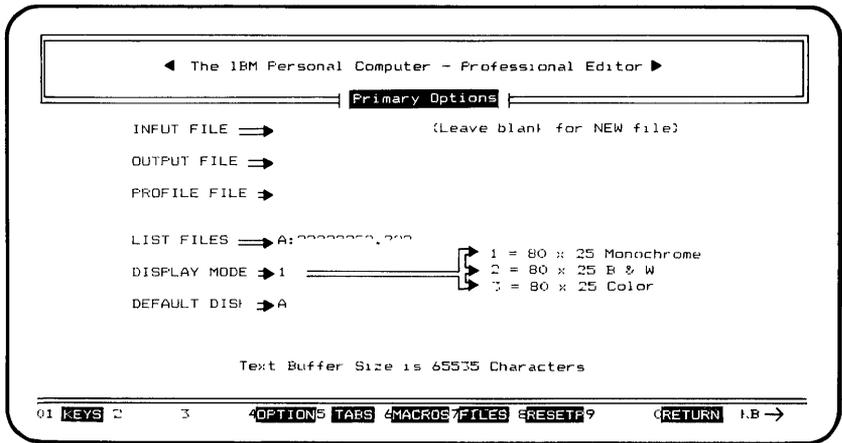


Figure 8. Primary Options Menu

Selecting the Input File

If a new file is to be created, leave the INPUT FILE field blank and press the Enter key. The Professional Editor then displays NEW FILE in this field.

If an existing file is to be edited, enter the filename in the INPUT FILE field using DOS file naming conventions. If no device is specified, the program assumes that the file resides on the default drive. Only disk devices (A or B) may be specified.

If a syntax error is detected in the filename, an error message is displayed. The error must be corrected before you may proceed to the next field.

You can modify the input filename by positioning the cursor to the beginning of the field. Then enter a new filename over the old one (or over NEW FILE, if it is being displayed), press the Erase EOL key (to clear any garbage remaining in the field), and press the Enter key.

When the input filename has been entered and verified, it is placed in the OUTPUT FILE field as a default output file. An appropriate profile name is then displayed in the PROFILE FILE field.

Note that the input filename may also be specified in the command which invokes the Professional Editor. For example,

EDIT EXAMPLE.TXT

Selecting the Output File

If the edited version of the input file is to replace the original version, press the Enter key to use the default name created when the input filename was entered. If a new file is being created, you must specify a different output filename.

If the edited file is to be stored under a different filename, enter the filename in the data field using DOS file naming conventions. If a file with this name already exists on the device specified, that file is replaced when the edited file is saved. If no device is specified, the Professional Editor assumes that the file resides on the default drive. Only disk devices may be specified.

Note: We recommend that you don't use **.TMP** for your file extension, because if you run out of space on your diskette, the Professional Editor creates a temporary file with that extension and your file may be overlaid.

If a syntax error is detected in the filename, an error message is displayed. The error must be corrected before you may proceed to the next field.

You can modify the output filename by positioning the cursor to the beginning of the field. Then enter a new filename over the old one and press the Erase EOL key followed by the Enter key.

Note: If you are planning to use the Read Only option, you *must not* specify an output filename.

Selecting the Profile File

The profile file is a small file which contains tab settings, options, etc. from a previous edit session. Generally, editor options are related to the type of data being edited. Consequently, it is often useful to have a different profile for each type of data being handled. In DOS, the file extension is used to indicate the type of data contained in the file (ASM for assembler source files, BAS for BASIC files, etc.).

To relate profile files to the types of data, the Professional Editor uses the file extension of the file being edited as the filename of the profile file with the extension of PRF. Thus, if the file TESTPROG.ASM is selected for editing, the profile file ASM.PRF is used. The profile file is read and written to the same device used by the input file. If a new file is being created, the same device used by the output file is used. If no file is found, one is created with the current options and tabs when you press the EDIT function key (F9) to begin editing.

After you select new options and press the EDIT function key, the updated options and tabs are written back into the profile file.

Selecting the List Files

You use this option to select a given set of files to be listed on the File List Menu. For example, on the Primary Options Menu, enter the following:

- *.*** Lists all files on the default diskette. (This is the default.)

- B:*.*** Lists all files on the diskette in drive B.

- A:*.COM** Lists all files with the extension **COM** on the diskette in drive A.

- A:COMMAND.COM** Lists only the file **COMMAND.COM** on the diskette in drive A.

The File List Menu is discussed later in this chapter.

Note: This feature is not available in the minimum version.

Selecting the Display Mode

This field shows the current display mode. If you want to use a different display mode, enter the appropriate value. (The required hardware must be installed or further operation of the system will not be possible.)

To select a different display and/or display mode, enter one of the following modes:

Mode Type of Display

- 1 Monochrome display (25 lines by 80 columns)
- 2 Color display in black and white mode (25 lines by 80 columns)
- 3 Color display in color mode (25 lines by 80 columns)

When you select a new mode, the menu is redisplayed on the screen, and that screen becomes active. The current contents of the old screen remain on the screen.

Selecting the Default Disk

This field shows the current DOS default drive. This drive is assumed when you enter a file specification without a designated drive. You can change the default by entering any valid installed diskette drive (A or B).

Text Buffer Size

This field shows the size of the text buffer and may not be altered.

Profile Options Menu

You use the Profile Options Menu to set and modify editor options that are stored in the profile file. This menu is displayed when you press the OPTION function key (F4) during the display of the Primary Options Menu.

To return to the Primary Options Menu, press the PRIME function key (F3).

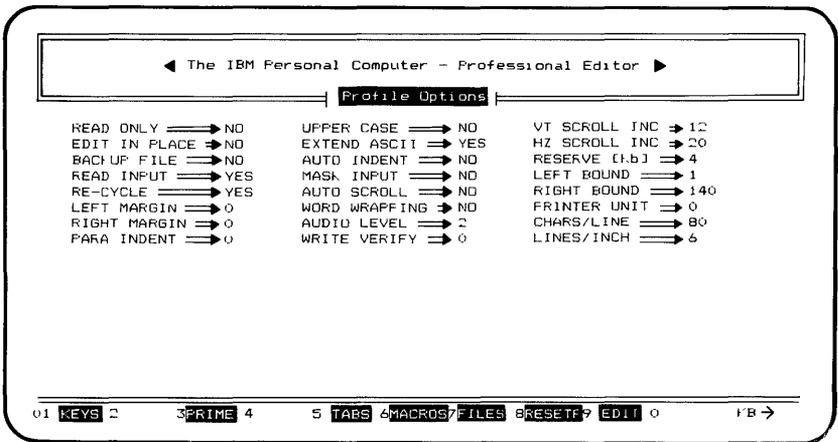


Figure 9. Profile Options Menu

Selecting Options

You can change the options by positioning the cursor to the data field and entering **YES** or **NO** (**Y** or **N** will do) or a new value. When an existing profile is used, the options on the menu are set to the values read from that profile. If no profile exists, default values are used.

The options available (and their defaults) are as follows:

Read Only (NO)

If you select this option, you can view the input file, but you cannot change it. Use this option when you want to browse a file without making any modifications. You cannot specify an output file when using this option.

Edit in Place (NO)

The Edit in Place option causes the input file to be updated in place. That is, the output file overlays the input file in the same diskette sectors. This option effectively doubles the maximum editable file size since new sectors need not be allocated for the output file. The penalty for this convenience is that you may not always be able to **QUIT** a session if you make a disastrous editing error.

The Professional Editor keeps track of the file positions for both reading and writing and does not let you write into a sector which has not yet been read into the text buffer. You get a warning message when you issue a PUT command which would do this, and the range of the PUT is automatically reduced. If you add a lot of text, you may get into a situation where you can't PUT one sector without causing an "inplace overlap," and you can't GET one sector because the buffer reserve is exceeded. The only way to resolve this problem is to issue the END command.

Since the first PUT issued with this option writes over the first part of the input file, you receive a warning at this time that the QUIT command is no longer available, and you are given a chance to change your mind.

The Edit in Place option is mutually exclusive with the Backup File option, and the input and output filenames and device must be the same.

Backup File (NO)

If you select this option and specify the same filename for both the input and output files, then the Professional Editor saves a copy of the original input file on the same diskette by renaming the extension of the file to BAK when the session is terminated by an END command. An existing backup file on the same diskette is deleted when the session begins so that its space can be reused. This option is ignored if the input and output filenames and devices are not identical.

Read Input (YES)

When you select this option, the Professional Editor automatically fills the text buffer with text from the input file when editing begins. If you do not select this option, no text is loaded, and the text buffer is left empty. You must then use the GET command to load text from the input file.

Re-Cycle (YES)

This option permits the Professional Editor to be “warm started” after you end the current editing session with an END or QUIT command.

If you select the RE-CYCLE option, the Professional Editor saves the initialization code and data in a file named “\$\$EDIT\$.CKP” on the default diskette. The file is created the first time you begin a session by pressing the EDIT function key. When you stop the session with an END or QUIT command, the initialization code and data are read from the checkpoint file and the Primary Options Menu is displayed, without returning to DOS. At this point, you may begin editing a new file. When you exit to DOS from the primary menu, the checkpoint file is deleted.

If you do not select the re-cycle option, the Professional Editor returns to DOS when you end a session with an END or QUIT command.

Left Margin (0)

The left margin defines the leftmost column to be used in text formatting operations. It may have a value of 0 through 132. A detailed description of this feature can be found in Chapter 9, “Using Margins.”

Right Margin (0)

The right margin defines the rightmost column to be used in text formatting and line input operations. It may have a value of 0 (indicating no right margin), or 5 through 137. The value of the left margin must always be at least five columns less than the right margin. A detailed description of this feature can be found in Chapter 9, “Using Margins.”

Paragraph Indentation (0)

The paragraph indentation defines the number of spaces to be inserted at the beginning of the first line of a paragraph in text formatting operations. It may have a value of 0 through 80 spaces. A detailed description of this feature can be found in Chapter 9, “Paragraph Indentation.”

Uppercase (NO)

When you select this option, the Professional Editor translates all lowercase ASCII characters entered from the keyboard to uppercase. The function is identical to pressing the Caps Lock key on the keyboard. Special (nonalphabetic) characters are not affected.

Note: Translation occurs only for characters entered from the keyboard. If the file contains lowercase characters, they are *not* translated to uppercase unless re-entered from the keyboard. You can use the translate line commands in Chapter 6 to translate existing data to uppercase or lowercase.

Extended ASCII (YES)

When you select this option, the Professional Editor allows all of the unrestricted extended ASCII codes (decimal 128-256) to be entered into the file from the keyboard. If you enter **NO**, these codes will be rejected. You can use this option to ensure that extended ASCII characters are not included in a file that you wish to transmit to a host.

Auto Indent (NO)

If you select this option, line indentation occurs automatically when you press the Enter key during editing. If you do not select this option, the cursor is placed at the left edge of the screen when you press the Enter key. You can turn this option on or off during editing with the INDENT function key (F20). See “Automatic Indentation,” in Chapter 3 for more information.

Mask Input (NO)

If you select this option, the Professional Editor places the input mask into each new line entered in input mode.

Note: If the length of the input mask is greater than the length defined by the left and right margins, then only that portion of the line within the margins will be inserted.

If you do not select this option, the Professional Editor inserts a blank line for each new line. You define the input mask on the Tabs and Mask Options Menu. (See “Input Mask Data,” later in this chapter for more information.)

You can turn this option on and off during the editing session by using the MASK function key (F18).

Auto Scroll (NO)

If you select this option, automatic scrolling occurs when the cursor is moved off any edge of the screen. If you do not select this option, automatic scrolling does not occur, and the cursor wraps around the screen.

You can turn this option on and off during the editing session by using the SCROLL function key (F19).

Word Wrapping (NO)

If you select this option and a right margin is defined, automatic word wrapping occurs as new lines are being inserted in input mode. When a word extends beyond the right margin, it is automatically moved to the beginning of the next line. If you do not select this option, words will not automatically wrap.

You can turn this option on and off during the editing session by using the WRAP function key (F15). See “Using Automatic Word Wrapping,” in Chapter 9 for a detailed description of this feature.

Audio Level (2)

The audio level option allows you to select the level of audio prompting for the edit session. You can set this option to:

Level What Is Produced

- 0 Audio prompting occurs only for error messages.
- 1 Audio prompting occurs for all messages.
- 2 Audio prompting includes all acknowledgements and all messages.

Write Verify (0)

Write verify is a feature of DOS which was added in Version 1.10. When you select this option, each sector written to a diskette file is read back and compared to ensure validity. This causes a significant reduction in the speed of diskette write operations but ensures that the data just written can be read later.

The Professional Editor allows you to use this feature when running with DOS Version 1.10. The feature is ignored by DOS Version 1.00. You may select this feature by setting the value of the write verify option as follows:

- 0 The Professional Editor ignores the write verify feature by neither turning it on or off.
- 1 The Professional Editor turns off the write verify option when the edit session begins.
- 2 The Professional Editor turns on the write verify option when the edit session begins.

Vertical Scroll Increment (12)

The vertical scroll increment is the number of lines advanced when you press the Scroll Up or Scroll Down key. You can set this value to any convenient number of lines (from 1-24). Or you can enter **H** (for half page, 12 lines) or **P** (for full page, 24 lines). You can change this option during the editing session by using the SET command.

Horizontal Scroll Increment (20)

The horizontal scroll increment is the number of columns advanced when you press the Scroll Left or Scroll Right key. You can set this value to any convenient number of columns (from 1-72). Or you can enter **H** (for half page, 40 columns) or **P** (for full page, 72 columns). You can change this option during the editing session by using the SET command.

Buffer Reserve (4)

The buffer reserve is the amount of space in the text buffer that is reserved for additions to the file being edited. When reading data from the file, reading terminates when the reserve space is exceeded. You specify this value in multiples of **K** bytes ($K = 1024$). You can change this option during the editing session by using the SET command.

Bounds (Left=1 Right=140)

The left and right bounds limit the scope of SEARCH and CHANGE operations to a given set of columns. You can change the boundaries during editing by using the BOUNDS command.

Printer Unit (0)

If your system is equipped with more than one printer, you may select the printer to be used: 0, 1, or 2. The default value for the IBM Matrix Printer is 0. You can change this option during the editing session by using the SET command.

Characters per Line (80)

This parameter determines the size of the type font to be used during printing. A value of 80 (or 8) provides 80 characters per line using the standard character font. A value of 132 (or 1) provides 132 characters per line using the condensed character font. You can change this option during the editing session by using the SET command.

Lines per Inch (6)

This parameter determines the line spacing of printed output and may be either 6 (6 lines/inch) or 8 (8 lines/inch). You can change this option during the editing session by using the SET command.

Tabs and Mask Options Menu

On the Tabs and Mask Options Menu, you define the desired tab settings and the optional input mask. It is displayed when you press the TABS function key (F5) during the display of the Primary Options Menu.

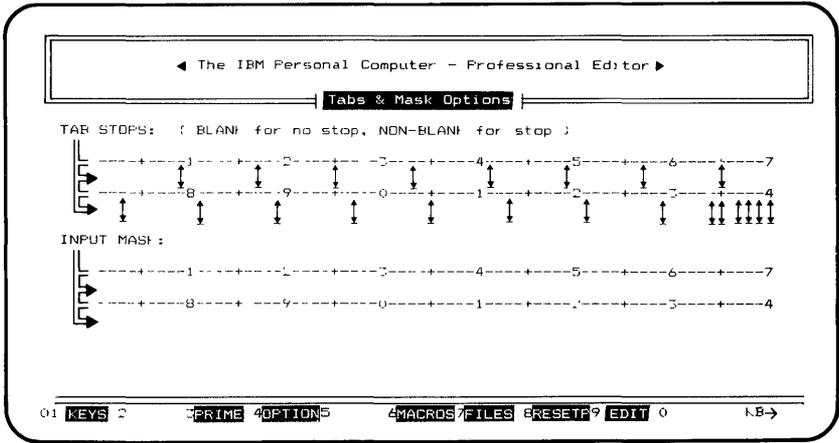


Figure 10. Tabs and Mask Options Menu

To return to the Primary Options Menu, press the PRIME function key (F3).

Tab Settings

The TAB STOPS area of the menu shows the current tab settings. When the Professional Editor is initialized, the tabs are set to the standard settings (a tab stop set every 8 columns beginning with column 1).

To change the tab settings, move the cursor to the appropriate display line for columns 1-70 or 71-140. Use a blank to indicate no tab stop and any non-blank character to indicate a tab stop in each column.

To restore the standard tab stops, move the cursor to the first display line and press the Restore Field key followed by the Enter key.

You can change this option during the editing session by using the TABS command.

We do not recommend using the Delete to EOL key for deleting tab stops because tab characters will be inserted in columns 71-140.

Input Mask Data

Data to be used as the input mask may be entered in the two data fields following INPUT MASK. The two lines represent columns 1-70 and 71-140, respectively. Any characters entered into these lines will be used as the mask data. To clear the mask, position the cursor to the beginning of each field on the menu and press the Delete EOL key.

You use the input mask to insert frequently-used lines. For example, you could use the following input mask in columns 72-80:

```
/* */
```

This mask causes all new lines to have `/* */` in columns 72-80. This is a commonly-used practice in some programming languages to add a comment at the end of each line.

Or, you could use the following input mask to specify a data classification:

INTERNAL USE ONLY

You can change this option during the editing session by using the **MASK** command.

Macro Definition Menu

On the Macro Definition Menu, you define data macros to be used during the edit session. This menu is displayed when you press the **MACROS** function key during the display of the Primary Options Menu. Refer to Chapter 8, “Using Macros” for a complete description of this menu.

To return to the Primary Options Menu, press the **PRIME** function Key (F3).

Note: This feature is not available in the minimum version.

File List Menu

The File List Menu provides a list of a given set of files on a diskette. This list is similar to the output produced by the DOS DIR command.

You specify the files to be listed in the LIST FILES option of the Primary Options Menu.

Note: This option is not available in the minimum version.

Displaying the File List Menu

After you select the files to be included in the list by setting the LIST FILES option on the Primary Options Menu, press the FILES function key (F7) to display the following File List Menu:

FILENAME	START OF DIRECTORY	SIZE	DATE	TIME	FILENAME	SIZE	DATE	TIME
COMMAND .COM		4959	05/07/82	12:00p	COLORBAR.BAS	1536	05/07/82	12:00p
FORMAT .COM		3816	05/07/82	12:00p	CALENDAR.BAS	3840	05/07/82	12:00p
CH1.DSK .COM		1720	05/07/82	12:00p	MUSIC .BAS	8704	05/07/82	12:00p
SYS .COM		605	05/07/82	12:00p	DONKEY .BAS	3584	05/07/82	12:00p
DISK COPY.COM		2008	05/07/82	12:00p	CIRCLE .BAS	1664	05/07/82	12:00p
DISK COMP.COM		1640	05/07/82	12:00p	PIECHART.BAS	2304	05/07/82	12:00p
COMP .COM		1649	05/07/82	12:00p	SPACE .BAS	1920	05/07/82	12:00p
EXE2BIN .EXE		1280	05/07/82	12:00p	HALL .BAS	2048	05/07/82	12:00p
MODE .COM		2509	05/07/82	12:00p	COMM .BAS	4352	05/07/82	12:00p
EDLIN .COM		2392	05/07/82	12:00p				
DEBUG .COM		5999	05/07/82	12:00p	END OF DIRECTORY			
LINK .EXE		41856	05/07/82	12:00p				
BASIC .COM		11392	05/07/82	12:00p				
BASICA .COM		16768	05/07/82	12:00p				
ART .BAS		1920	05/07/82	12:00p				
SAMPLES .BAS		2432	05/07/82	12:00p				
MORTGAGE.BAS		6272	05/07/82	12:00p				

01 KEYS 2 3PRIME 4OPTION 5 TABS 6MACROS 7 8 9 EDIT 0 FB

Figure 11. File List Menu

The information for each file includes:

- The filename and extension
- The number of bytes used by the file
- The date on which the file was created or last updated
- The time at which the file was created or last updated

Files are listed in the order of their occurrence in the directory. Hidden files are not included.

Note: If the directory contains more than 35 entries, the END OF DIRECTORY indicator does not appear. To see the next page of the file list, press the PgDn key. To scroll upward, press the PgUp key.

To exit from the file list, press any valid function key.

CHAPTER 5. USING THE PROFESSIONAL EDITOR

Contents

Beginning the Edit Session	5-3
Professional Editor Modes	5-5
Edit Mode	5-5
Input Mode	5-6
Scrolling	5-6
Vertical Scrolling	5-6
Horizontal Scrolling	5-7
Automatic Scrolling	5-8
Editing Lines	5-9
Changing Text	5-9
Inserting Characters	5-11
Deleting Characters	5-11
Recovering Deleted Characters	5-12
Splitting a Line into Multiple Lines ...	5-13
Inserting New Lines	5-13
Deleting Lines	5-14
Recovering Deleted Lines	5-14
Ending the Edit Session	5-15
With Save (END Function Key)	5-15
Without Save (QUIT Function Key) ...	5-16
From the Primary Options Menu	5-16

Beginning the Edit Session

To start the Professional Editor from DOS command mode, enter the command:

A>EDIT [*input filename*]

(Enter **EDIT64** if you are using the minimum version and have named your program **EDIT64**.)

The input filename operand is optional. If present, the Professional Editor places it on the Primary Options Menu as the input filename. It is useful to specify the filename here when the filename is displayed above on the screen from a **DIR** command. The Professional Editor obtains all other needed information from the menus. When you enter the Professional Editor, DOS loads the editor program and gives it control. After initialization, the Professional Editor presents the Primary Options Menu and waits for your input.

When all required options have been entered on the Primary Options Menu, the Professional Editor enables the **EDIT** function key (F9). To begin editing the input file, press the **EDIT** function key.

If the READ INPUT option is selected, the Professional Editor reads the input file into the text buffer. If the file is too large to fit in the available text buffer, reading stops when the buffer reserve is exceeded. This allows a portion of the text buffer to remain available for changes to be made to the file. You can use the PUT command to write a selected portion of the text to the output file if more room is needed.

When editing of the first segment is complete, use the NEXT command to write the segment to the output file and read the next segment from the input file.

If you do not select the READ INPUT option, you must use the GET command to read text from the input file into the text buffer.

If a new file is being created, the Professional Editor enters input mode immediately to allow new data to be entered.

Professional Editor Modes

The Professional Editor has four different modes of operation for editing a file:

- Command Mode
- Edit Mode
- Input Mode
- Line Command Mode

Command mode is described in Chapter 7, “Editor Commands.” Line Command Mode is described in Chapter 6, “Line Commands.”

Edit Mode

You use edit mode to modify the file data displayed on lines 1-24 of the screen. You enter edit mode from command mode by pressing the SWAP function key (F2). The command input and status areas on line 24 are replaced by the 24th data line. The cursor is set to the beginning of the current line of text in lines 1-24, or to line 1 if no current line has been defined.

You can change (replace) file data by entering new data on top of the old data on the screen. You use the insert and delete character functions to insert and delete characters from the line being edited.

To return to command mode, press the SWAP function key (F2).

Input Mode

Input mode is used to insert new lines between existing lines on the screen or to insert lines into a new file.

Scrolling

Two types of scrolling may be performed, vertical and horizontal.

Vertical Scrolling

Vertical scrolling permits you to roll the data up and down the screen continuously from the beginning of the text to the end. Press the Scroll Down keypad key to scroll forward toward the end of the text. Press the Scroll Up keypad key to scroll backward toward the beginning of the file.

When you press the Scroll Down key, lines 13-24 are moved upward into lines 1-12, and the next 12 lines of file data are displayed in lines 13-24. When you press the Scroll Up key, lines 1-12 are moved downward into lines 13-24, and the previous 12 lines of file data are displayed in lines 1-12. You can change the number of lines moved in a vertical scroll operation (Vertical Scroll Increment) with the SET command. The default scroll increment is 12 lines (half-screen).

In scrolling vertically, the cursor follows the movement of data so that the cursor remains positioned on the same line of text as long as that

line remains on the screen. If the line drops off the screen, the cursor is set to line 1.

The **Scroll Top** and **Scroll Bottom** keys position the screen to the top and bottom of the text, respectively.

You may use scrolling in command, edit, and line command modes. If you press a vertical scroll key in input mode, input mode is ended before the scrolling is performed.

Vertical scrolling functions may also be performed using the editor commands **UP**, **DOWN**, **TOP**, and **BOTTOM**. (See Chapter 7.)

Horizontal Scrolling

Horizontal scrolling permits you to move the data on the screen horizontally in order to bring the desired columns of data into the screen window.

Press the **Scroll Right** key to move to the right within the lines on the screen (the data actually moves left). The screen window moves by the number of columns specified by the **Horizontal Scroll Increment**. When the last column of data (column 140) appears on the screen, further scrolling to the right is prohibited.

Press the **Scroll Left** key to move to the left within the lines on the screen (the data actually moves right). The screen window moves by the number of columns specified by the **Horizontal Scroll Increment**. When the first column of data (column 1) appears on the screen, further scrolling to the left is prohibited.

You can change the number of columns moved in a horizontal scroll operation (**Horizontal Scroll Increment**) with the **SET** command.

Automatic Scrolling

The automatic scrolling feature (AUTO SCROLL) permits vertical and horizontal scrolling to occur automatically when the cursor moves off the edge of the screen. If the cursor moves off the bottom of the screen when you press the Cursor Down key, a scroll down operation is performed. In the same way, movement of the cursor off the top, left, or right edge causes a scroll up, left, or right operation to occur.

You can toggle automatic scrolling on or off by pressing the SCROLL function key (F19). The character **S** appears in the MD status display area when it is on.

When you insert new lines of data in input mode, an automatic scroll right always occurs if you type off the right edge of the screen, regardless of the status of AUTO SCROLL. This allows you to continuously enter a line of data that is wider than the screen without stopping to scroll. If AUTO SCROLL is enabled, an automatic scroll left to the beginning of the line occurs when you press the Enter key to allow the next line to be started at the beginning.

Editing Lines

When editing file data, you position the file so that the text to be changed is in the screen window by using the scrolling functions. If it is not already in edit mode, you place the program in edit mode by using the SWAP function key (F2). This places the cursor in the data area on the screen and allows the file text to be directly modified when you enter new data from the keyboard.

Changing Text

You can change file data by positioning the cursor to the line to be changed using the Cursor Up or Cursor Down keys. The horizontal cursor position is maintained during the upward or downward movement as long as the cursor does not become positioned in a tab space. When the desired line is found, you may then position the cursor horizontally to the text to be changed using the Cursor Right or Cursor Left keys. If necessary, you may use the Scroll Left or Scroll Right keys to bring the text to be modified into the screen window. You can then change the data by entering new characters on top of the existing characters. When the data is changed, the program automatically adjusts the spacing of the data on the screen to account for tabs that are added, deleted, or modified.

During the editing of the line, you may use the Cursor Left or Cursor Right keys to position the cursor to any character within the line. If the cursor is positioned beyond the current end of the line, the line is extended and filled with space characters if needed.

As the cursor moves horizontally over the line, it skips over blanks on the screen that represent spaces created by tab skips. These spaces are automatically inserted and deleted when tab characters are inserted or deleted. Tab characters are not overlaid in this mode except by another tab character. You must use the Del key to delete a tab.

You may use the Cursor to BOL key to reposition the cursor to the beginning of the line or left edge of the screen.

When you have finished editing the line, press the Enter key to advance to the next line. The cursor is then positioned at the beginning of the next line.

If you make an error while editing a line of text, you may restore the line from the text buffer by pressing the Restore Field key.

Note: When lines of text are changed, any trailing space or tab characters are removed from the line before it is written back to the text buffer. This is done to conserve diskette space.

Inserting Characters

To insert characters between existing characters in a line, press the Ins key. The letter I is then displayed in the keyboard status area. Characters entered from the keyboard are then inserted before the character over the cursor, pushing the remainder of the line to the right. Correct tab spacing is maintained as the data is shifted on the screen.

If the line becomes full, you receive the message:

Line is FULL Insert character(s) NOT accepted.

When this occurs, you must delete characters before any more characters may be inserted into the line. This prevents data from being lost accidentally by being pushed off the end of a line.

To return to normal operation, press the Ins key again. Insert Character mode is *not* turned off automatically when you press the Cursor Up or Cursor Down key. Insert Character mode *is* turned off when you press the Enter key.

Deleting Characters

To delete a character within a line, position the cursor to the character to be deleted and press the Del key. The character is deleted, and the line shifts to the left. Correct tab spacing is maintained as the data is shifted on the screen. You can use the Del key while in insert character mode.

You may also delete all characters from the current cursor position to the end of the line by pressing the Delete to EOL key. If the cursor is positioned at the beginning of the line, a null line is left, which contains only the end of line sequence (CR-LF) characters.

When you use either of these methods, the characters deleted are saved (up to a maximum of 140 characters) and may be recovered as explained below.

Recovering Deleted Characters

Characters that you delete with the Del or Delete to EOL key are saved by the Professional Editor until a maximum of 140 characters have been accumulated. You may recover these characters in the reverse order in which they were deleted, by doing the following:

- Position the cursor to the point where the characters are to be re-inserted. If you have just deleted characters, the cursor is already correctly positioned. The characters will be inserted to the right of the cursor.
- Press the Ins key to enter insert character mode.
- Press the Esc key once for each character to be recovered.
- Characters are recovered in the reverse order in which they were deleted.
- Press the Ins key again to exit from insert character mode (if desired) and continue editing.

When the number of saved characters exceeds 140, the oldest characters are discarded as new characters are added.

Splitting a Line into Multiple Lines

It is sometimes convenient to split a line into two or more consecutive lines. To do this, use the SPLIT function key (F17).

First, position the cursor to the point in the line where you want the division to be made. This position may not be at or beyond the current end of the line, or an error message is displayed. While at this position, press the SPLIT function key. The line is split at the cursor position. The character above the cursor becomes the first character of a new line.

Note: The SPLIT function key is enabled *only* while in edit mode.

Inserting New Lines

To insert new lines into the text, position the cursor to the beginning of the line which immediately precedes the insert point and press the INSERT function key. If you want to insert new lines at the beginning of the text, place the cursor on the START OF TEXT special message line on the screen. The existing lines on the screen following the insert point are then pushed down one line, and a blank line is created, with the cursor placed at the beginning of the line.

The prompt **INSERT>>** is displayed in the line command input area at the left edge of the screen to show where the new line is being inserted.

Enter the new line and press the Enter key. Another blank line is created for the next line to be inserted. Continue to insert lines by entering the data and pressing the Enter key.

If you type off the right edge of the screen, an automatic scroll right is performed to allow you to continue typing on the same line. The Professional Editor automatically scrolls back to the beginning of the line when you press the Enter key.

To get out of input mode, press the Enter key without entering any data. The program deletes the unused line and returns to the mode from which you entered input mode. You also end input mode if you press the Cursor Up or Cursor Down key.

Note: To create a completely empty line (one which contains no characters other than the ending CR-LF), type a single space and press the Enter key.

Deleting Lines

You delete lines of text by positioning the cursor to the line to be deleted and pressing the DELETE function key. The line is deleted from the screen, and subsequent lines are pushed up to fill the gap, with a new line appearing on line 24.

Recovering Deleted Lines

The Professional Editor maintains a stack of up to five lines which you have deleted with the DELETE function key. As lines are added to the top of the stack, the oldest line at the bottom of the stack is discarded. If you accidentally delete lines with the DELETE function key, the last five lines are available for recovery from the stack. To recover a line, press the UNDELE function key. The last line deleted is then placed in front of the current line. The restored line is removed from the

stack, and the next deleted line is then ready to be restored. To recover another line, press the UNDELE function key again.

Notes:

1. The UNDELE function key is enabled *only* after you press the DELETE function key.
2. The UNDELE function key works differently when you delete the *last* line of a file. (See “Recovering Deleted Data,” in Chapter 2 for details.)

Ending the Edit Session

With Save (END Function Key)

When you have finished editing the file, you may save the file on a diskette and end the session by pressing the END function key or by entering the END command. If RECYCLE=YES, control returns to initialization after the file has been written to diskette.

If the file has not been modified when you enter the END command, you see the following message:

File has NOT been modified. Save anyway? <Y or N> >

To save the file, reply Y. Reply N to cancel the END command and continue editing.

Without Save (QUIT Function Key)

To end the edit session without saving the file being edited, press the QUIT function key or enter the QUIT command. The edited file is discarded, and the original file remains as it was.

If the file was modified, you see the following:

File has been modified. QUIT anyway? <Y or N> >

This message ensures that you really want to throw away the edited file. If you do, reply Y. If not, reply N, and the editing session continues.

The main purpose of the QUIT function is to allow you to throw away a file that is accidentally damaged during the edit and start over again. If RECYCLE=YES, control returns to initialization.

Note: When you use the Edit in Place option, you cannot QUIT a session once data has been written into the file being processed. The Professional Editor issues a message to inform you when you can no longer QUIT.

From the Primary Options Menu

In initialization, you can end the Professional Editor by pressing the RETURN function key (F10). Control returns to DOS command mode.

CHAPTER 6. LINE COMMANDS

Contents

Line Commands	6-3
Entering Line Commands	6-3
Cancelling an Incomplete (Pending) Line Command	6-4
Performing Other Editing Functions ...	6-4
Syntax of Line Commands	6-5
Count Operand	6-5
Operation Sub-Commands	6-5
Destination Sub-Commands	6-6
AFTER (A) Sub-Command	6-6
BEFORE (B) Sub-Command	6-6
Copy Operations	6-7
Copying a Single Line	6-7
Copying a Group of Lines	6-7
Example	6-8
Example	6-9
Copy and Repeat Operations	6-10
Delete Operations	6-11
Deleting a Single Line	6-11
Deleting a Group of Lines	6-11
Deleting to End of Segment	6-12
Deleting from Beginning of Segment	6-12

Extract Operations	6-13
Opening the Extract File	6-13
Extracting a Single Line	6-14
Extracting a Group of Lines	6-14
Extracting to End of Segment	6-15
Extracting from Beginning of Segment	6-15
Closing the Extract File	6-16
Gather Operations	6-16
Gathering a Group of Lines	6-17
Gathering to End of Segment	6-18
Gathering from Beginning of Segment	6-18
Move Operations	6-19
Moving a Single Line	6-19
Moving a Group of Lines	6-19
Print Operations	6-21
Printing a Group of Lines	6-21
Printing to End of Segment	6-22
Printing from Beginning of Segment ...	6-22
Uppercase and Lowercase Translation Operations	6-23
Translating a Single Line	6-23
Translating a Group of Lines	6-23
Translating to End of Segment	6-24
Translating from Beginning of Segment	6-24
Displaying the Column Grid	6-25

This chapter describes the line commands and their use. (See Chapter 9 for additional line commands used for text processing.)

Line Commands

Line commands are used to manipulate an entire line of text or a group of consecutive lines called a *block*. A *line command* consists of one or more sub-commands which identify the operation to be performed (move, copy, etc.), define the starting and ending lines of the block of data, and, when required, define the destination for the data.

Line sub-commands differ from the regular editor commands in that you enter them into the *line command area* at the left of the data line to be referenced, rather than in the command input area in line 24. The position of the line sub-commands on the screen, in relation to the lines of text, determines which lines of text are affected by the command being entered.

Entering Line Commands

You can enter line commands from command or edit mode by pressing the LINE function key (F3). When you press this key, the cursor moves to the line command area of the current line. If you press the LINE function key while in line mode, you return to edit mode, and the cursor moves to the beginning of the line of data.

You enter line sub-commands by typing the sub-command verb in the line command input area of the line to be manipulated. The sub-command is displayed in reverse video so that it may be quickly located on the screen. Some line commands require several sub-commands to be entered on different lines to complete the entire command before it is executed. When you have entered all the required sub-commands, the line command is executed and the sub-commands are removed from the screen.

When a line command requires more than one sub-command for completion, you may enter the sub-commands in any order. You do not need to enter all of the sub-commands on the same display page. You may scroll freely over the entire file and enter sub-commands at any point.

Cancelling an Incomplete (Pending) Line Command

When you enter the first sub-command of a line operation, the CANCEL function key (F7) is enabled. You may then use this key to cancel all sub-commands of an incomplete line command. You can cancel individual sub-commands by typing spaces over them.

Performing Other Editing Functions

While a line command is pending, you may perform other editing operations. If you move the cursor to the right into the data area, you may make modifications to the text in the normal ways. You may also use the SWAP function key to enter command mode and execute regular editor commands. If you delete a line on which a line sub-command was entered (and is still pending), the sub-command is cancelled automatically.

Syntax of Line Commands

A line command may consist of one or more sub-commands. The sub-command is entered in either uppercase or lowercase in the line command area of the line being manipulated.

Count Operand

Some sub-commands have an optional count operand field. When you use a count operand, you type it following the verb, with or without optional space characters. A *count operand* is an integer whose value is between 1 and 65535. You use a minus sign (-) to indicate a negative count when appropriate. You can use an * to denote all lines (* represents an “infinite” value of 65535). If you do not enter the count, a default value of 1 is assumed.

Operation Sub-Commands

Operation sub-commands specify the operation to be performed and identify the limits of the block of data. Two forms of sub-commands may be used.

One form consists of a *single letter* verb followed by an optional count operand. The *letter* defines the operation to be performed. The *count* defines the block of data in relation to the current line:

- If you enter a positive count n , the block begins with the current line and includes the following $n-1$ lines.
- If you enter a negative count n , the block ends with the current line and includes the preceding $n-1$ lines.

- If you enter the value *, the block begins with the current line and includes all lines to the end of the current segment.
- If you enter the value —*, the block ends with the current line and includes all lines from the beginning of the current segment.

The second form consists of a *2-character* verb and marks the current line as the start or end of the block. You must enter the same sub-command again on another line to complete the set and define the block. The first occurrence of the verb in relation to the text defines the start of the block. The second occurrence of the verb in relation to the text defines the end of the block and includes that line.

Destination Sub-Commands

Some line commands that move and copy data require that the *destination* for the block be defined. The destination sub-commands **A** and **B** define the point at which the data block is to be inserted. If these sub-commands are placed within the block being manipulated, a sub-command overlap error occurs.

AFTER (A) Sub-Command

The sub-command **A** designates that the block selected is to be inserted *after* the line on which **A** appears.

BEFORE (B) Sub-Command

The sub-command **B** designates that the block selected is to be inserted *before* the line on which **B** appears.

Copy Operations

The line copy commands insert a copy of one or more consecutive lines at a new location in the text file. A copy operation requires the use of the sub-commands **C** or **CC** to define the block, and a destination sub-command **A** or **B** to define the destination.

Copying a Single Line

To copy a single line of text, position the cursor to the line command input area of the line to be copied and enter the sub-command **C**. Then position the cursor to the line command input area of the line at the point where the line is to be inserted, and enter the sub-command **A** if the line is to be inserted *after* the current line, or **B** if the line is to be inserted *before* the current line. The original line is then copied before or after the current line.

Copying a Group of Lines

To use one method of copying a group of consecutive lines, position the cursor to the line command input area of the first line of the group to be copied and enter the sub-command:

C *nnnnn*

where *nnnnn* is the number of consecutive lines to be copied, including the current line. If you do not enter a count, a default value of 1 is used. Then, position the cursor to the line command input area at the point where the group is to be inserted, and

enter the sub-command **A** if the group is to be inserted *after* the current line, or **B** if the group is to be inserted *before* the current line. The original line and the *next* $n-1$ lines are then copied.

The sub-command:

C -*nnnn*

copies the current line and the *previous* $n-1$ lines.

Example

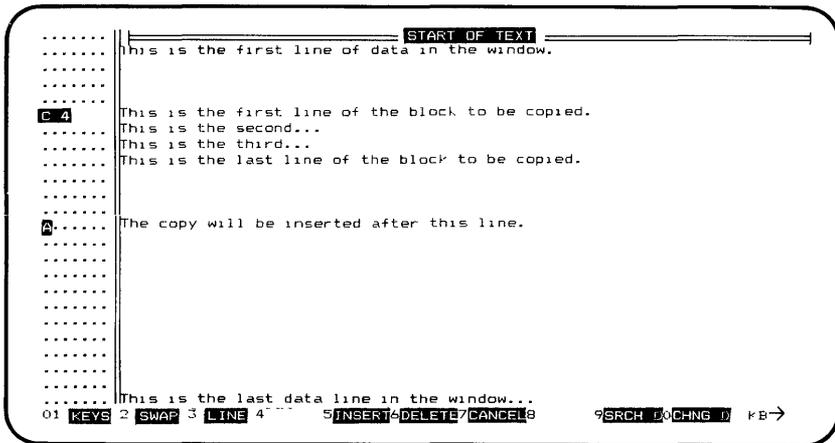


Figure 12. Example 1 of a Line Copy Operation

To use the second method, position the cursor to the line command input area of the first (or last) line of the group of consecutive lines to be copied and enter the sub-command:

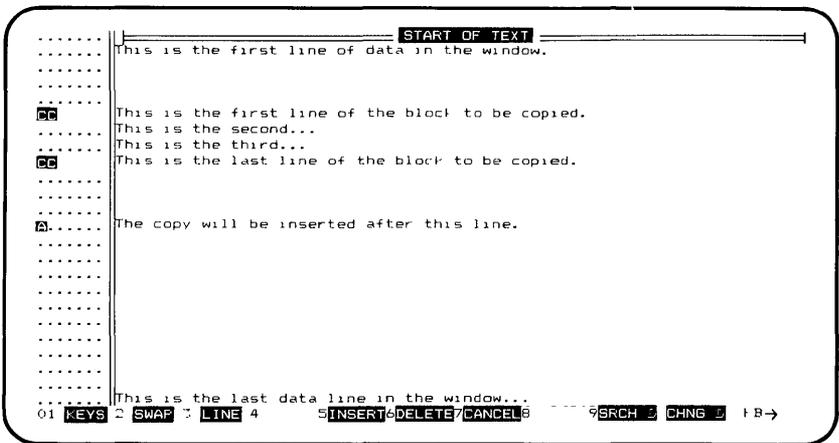
CC

Then position the cursor to the line command input area of the last (or first) line of the group to be copied and enter the sub-command:

CC

Then position the cursor to the line command input area at the point where the group is to be inserted, and enter the sub-command **A** if the group is to be inserted *after* the current line, or **B** if the group is to be inserted *before* the current line. The lines beginning with the first **CC** through the last **CC** line are then copied.

Example



```
..... START OF TEXT .....
..... This is the first line of data in the window.
.....
..... This is the first line of the block to be copied.
..... This is the second...
..... This is the third...
..... This is the last line of the block to be copied.
.....
..... The copy will be inserted after this line.
.....
.....
.....
..... This is the last data line in the window...
01 KEYS 2 SWAP 3 LINE 4 5 INSERT 6 DELETE 7 CANCEL 8 9 SRCH 0 CHNG 1 B->
```

Figure 13. Example 2 of a Line Copy Operation

Delete Operations

The line delete commands delete one or more consecutive lines of text. A delete operation requires the use of the sub-commands **D** or **DD** to define the block. A destination sub-command is not used.

Note: Lines that you delete with line delete commands are not saved and cannot be recovered.

Deleting a Single Line

To delete a single line of text, position the cursor to the line command input area of the line to be deleted and enter the sub-command **D**.

Deleting a Group of Lines

To use one method of deleting a group of consecutive lines, position the cursor to the line command input area of the first line of the group to be deleted and enter the sub-command:

D *nnnnn*

where *nnnnn* is the number of consecutive lines to be deleted, including the current line. If you do not enter a count, a default value of 1 is used. The current line and the *next* $n-1$ lines are then deleted.

The sub-command:

D-*nnnn*

deletes the current line and the *previous n-1* lines.

To use the second method, position the cursor to the line command input area of the first (or last) line of the group of consecutive lines to be deleted and enter the sub-command:

DD

Then position the cursor to the line command input area of the last (or first) line of the group to be deleted and enter the sub-command:

DD

The lines beginning with the first **DD** line through the last **DD** line are then deleted.

Deleting to End of Segment

To delete all lines from the current line to the end of the segment, position the cursor to the line command input area of the first line to be deleted and enter the sub-command:

D *

Deleting from Beginning of Segment

To delete all lines from the beginning of the segment through the current line, position the cursor to the line command input area of the last line to be deleted and enter the sub-command:

D -*

Extract Operations

The line extract operation creates an external file and copies blocks of data from the file being edited into the new file. An extract operation requires the use of the sub-commands **E** or **EE** to define the block. A destination sub-command is not used.

You must open the extract file with the **EOPEN** command before any data can be written to it.

Opening the Extract File

The **EOPEN** command opens a file to hold the extracted portion of the file being edited. The format of the **EOPEN** command is:

EOPEN|EO *filename*

where *filename* is the complete name of the extract file to be used. If you do not enter the device specification, the current default device is used. If the file named already exists, you receive the following message:

Extract file already exists. Want to OVER WRITE? (Y or N)

If you reply **Y**, the existing file is replaced by the new extract file. If you do not want to replace the existing file, reply **N**, and the **EOPEN** command is cancelled.

Once the extract file has been opened, you use the extract line commands (**E** and **EE**) to select the data to be placed in the file. The file remains open and available for use by subsequent extract commands until you close it by using the **ECLOSE** command. If the file is still open when you enter an **END** or **QUIT** command, the file is automatically closed.

Extracting a Single Line

To extract a single line of text, position the cursor to the line command input area of the line to be extracted and enter the sub-command **E**. The line is then written to the extract file.

Extracting a Group of Lines

To use one method of extracting a group of consecutive lines, position the cursor to the line command input area of the first line of the group to be extracted and enter the sub-command:

E *nnnnn*

where *nnnnn* is the number of consecutive lines to be extracted, including the current line. If you do not enter a count, a default value of 1 is used. The current line and the *next n-1* lines are then written to the extract file.

The sub-command:

E *-nnnnn*

extracts the current line and the *previous n-1* lines.

To use the second method, position the cursor to the line command input area of the first (or last) line of the group of consecutive lines to be extracted and enter the sub-command:

EE

Then position the cursor to the line command input area of the last (or first) line of the group to be extracted and enter the sub-command:

EE

The lines beginning with the first **EE** line through the last **EE** line are then written to the extract file.

Extracting to End of Segment

To extract all lines from the current line to the end of the segment, position the cursor to the line command input area of the first line to be extracted and enter the sub-command:

E *

Extracting from Beginning of Segment

To extract all lines from the beginning of the segment through the current line, position the cursor to the line command input area of the last line to be extracted and enter the sub-command:

E -*

Closing the Extract File

You use the ECLOSE command to close the extract file which you previously opened with the EOPEN command. The format of the ECLOSE command is:

```
ECLOSE|EC
```

If you have written any text to the extract file, the file is closed in a normal manner. If no text is written to the extract file, an empty file is created.

Gather Operations

The line gather operation consolidates several consecutive lines of text into a single line. You do this by replacing the terminating CR-LF characters that separate the individual lines of the block with a space, leaving one line. The length of the resultant line must not exceed the maximum line length of 140 bytes, or an error occurs. A gather operation requires the use of the sub-commands **G** or **GG** to define the block. A destination sub-command is not used.

Gathering a Group of Lines

To use one method of gathering a group of consecutive lines, position the cursor to the line command input area of the first line of the group to be gathered and enter the sub-command:

G *nnnnn*

where *nnnnn* is the number of consecutive lines to be gathered, including the current line. If you do not enter a count, no operation is performed. The current line and the *next* $n-1$ lines are then gathered.

The sub-command:

G *-nnnnn*

gathers the current line and the *previous* $n-1$ lines.

To use the second method, position the cursor to the line command input area of the first (or last) line of the group of consecutive lines to be gathered and enter the sub-command:

GG

Then position the cursor to the line command input area of the last (or first) line of the group to be gathered and enter the sub-command:

GG

The lines beginning with the first **GG** line through the last **GG** line are then gathered.

Gathering to End of Segment

To gather all lines from the current line to the end of the segment, position the cursor to the line command input area of the first line to be gathered and enter the sub-command:

G *

Note: This use is likely to cause the maximum line length to be exceeded, in which case an error message is displayed.

Gathering from Beginning of Segment

To gather all lines from the beginning of the segment through the current line, position the cursor to the line command input area of the last line to be gathered and enter the sub-command:

G -*

Note: This use is likely to cause the maximum line length to be exceeded, in which case an error message is displayed.

Move Operations

You use the line move operations to move one or more consecutive lines to a new location in the text file. A move operation requires the use of the sub-commands **M** or **MM** to define the block, and a destination sub-command **A** or **B** to define the destination.

Moving a Single Line

To move a single line of text, position the cursor to the line command input area of the line to be moved and enter the sub-command **M**. Then position the cursor to the line command input area of the line at the point where the line is to be inserted, and enter the sub-command **A** if the line is to be inserted *after* the current line, or **B** if the line is to be inserted *before* the current line.

The line is then inserted before or after the current line and deleted from its original position in the file.

Moving a Group of Lines

To use one method of moving a group of consecutive lines, position the cursor to the line command input area of the first line of the group to be moved and enter the sub-command:

M *nnnnn*

where *nnnnn* is the number of consecutive lines to be moved, including the current line. If you do not

enter a count, a default value of 1 is used. Then position the cursor to the line command input area at the point where the group is to be inserted, and enter the sub-command **A** if the group is to be inserted *after* the current line, or **B** if the group is to be inserted *before* the current line. The current line and the *next n-1* lines are then moved.

The sub-command:

M -*nnnn*

moves the current line and the *previous n-1* lines.

To use the second method, position the cursor to the line command input area of the first (or last) line of the group of consecutive lines to be moved and enter the sub-command:

MM

Then position the cursor to the line command input area of the last (or first) line of the group to be moved and enter the sub-command:

MM

Then position the cursor to the line command input area at the point where the group is to be inserted, and enter the sub-command **A** if the group is to be inserted *after* the current line, or **B** if the group is to be inserted *before* the current line. The lines beginning with the first **MM** line through the last **MM** line are then moved.

Print Operations

You use the line print operations to print a group of consecutive lines on the line printer. Only text lines are printed, and the lines are formatted using the current tab settings. A destination sub-command is not used.

Printing a Group of Lines

To use one method of printing a group of consecutive lines, position the cursor to the line command input area of the first line of the group to be printed and enter the sub-command:

P *nnnnn*

where *nnnnn* is the number of consecutive lines to be printed, including the current line. If you do not enter a count, one line is assumed. The current line and the *next* $n-1$ lines are then printed.

The sub-command:

P *-nnnnn*

prints the current line and the *previous* $n-1$ lines.

To use the second method, position the cursor to the line command input area of the first (or last) line of the group of consecutive lines to be printed and enter the sub-command:

PP

Then position the cursor to the line command input area of the last (or first) line of the group to be printed and enter the sub-command:

PP

The lines beginning with the first **PP** line through the last **PP** line are then printed.

Printing to End of Segment

To print all lines from the current line to the end of the segment, position the cursor to the line command input area of the first line to be printed and enter the sub-command:

P*

Printing from Beginning of Segment

To print all lines from the beginning of the segment through the current line, position the cursor to the line command input area of the last line to be printed and enter the sub-command:

P_*

Uppercase and Lowercase Translation Operations

You use the line translation commands to translate one or more consecutive lines of text to uppercase (U or UU) or lowercase (L or LL). A destination sub-command A or B is not used.

Translating a Single Line

To translate a single line of text, position the cursor to the line command input area of the line to be translated, and enter the sub-command L to translate to lowercase, or U to translate to uppercase.

Translating a Group of Lines

To use one method of translating a group of consecutive lines, position the cursor to the line command input area of the first line of the group to be translated and enter the sub-command:

L *nnnnn* or **U** *nnnnn*

where *nnnnn* is the number of consecutive lines to be translated, including the current line. If you do not enter a count, a default value of 1 is used. The current line and the *next n-1* lines are then translated.

The sub-command:

L-*nnnnn* or **U**-*nnnnn*

translates the current line and the *previous n-1* lines.

To use the second method, position the cursor to the line command input area of the first (or last) line of the group of consecutive lines to be translated and enter the sub-command:

LL or **UU**

Then position the cursor to the line command input area of the last (or first) line of the group to be translated and enter the sub-command:

LL or **UU**

The lines beginning with the first **LL** or **UU** line through the last **LL** or **UU** line are then translated.

Translating to End of Segment

To translate all lines from the current line to the end of the segment, position the cursor to the line command input area of the first line to be translated and enter the sub-command:

L* or **U***

Translating from Beginning of Segment

To translate all lines from the beginning of the segment through the current line, position the cursor to the line command input area of the last line to be translated and enter the sub-command:

L-* or **U-***

Displaying the Column Grid

You may display the column grid to determine the column positions of the characters in a line. To display the grid, position the cursor to the line command input area of a line and enter the command:

COL

The grid is displayed on the line which precedes the current line, and all preceding text lines on the screen are pushed upward by one line. Current boundaries are shown on the grid as left and right arrow symbols. The column grid has the format:

```
<-----1-----2-----3-----4-----5-----6-----7-----8
```

The column grid remains on the screen until it rolls off the screen by a scrolling operation or until you enter a line command. You may delete the grid by positioning the cursor to the beginning of the grid line and pressing the DELETE function key. The column grid does not become part of the text file.

CHAPTER 7. EDITOR COMMANDS

Contents

Using Editor Commands	7-3
Using the Command Stack	7-4
Command Syntax	7-5
BOTTOM Command	7-6
BOUNDS Command	7-7
CHANGE Command	7-8
COLUMN Command	7-14
DOWN Command	7-15
ECLOSE Command	7-16
END Command	7-17
EOPEN Command	7-18
FILES Command	7-19
GET Command	7-20
LEFT Command	7-21
LINENO Command	7-22
MACRO Command	7-23

MARGIN Command	7-25
MASK Command	7-26
MERGE Command	7-27
NEXT Command	7-28
PUT Command	7-30
QUIT Command	7-31
RIGHT Command	7-32
SEARCH Command	7-33
SET Commands	7-40
TABS Command	7-44
TOP Command	7-45
UP Command	7-46

This chapter provides a detailed reference to editor commands and their syntax. For easy reference, commands are described in alphabetical order.

Using Editor Commands

You use command mode to enter commands via the command input area on line 24. You enter command mode from edit mode by pressing the SWAP function key (F2). The command prompt **COMMAND>** is displayed in the left on line 24 followed by the cursor.

To enter a command, type the command verb and operands and press the Enter key. When the command has been executed, the prompt is reissued, and the program waits for the next command. You may use the function and keypad keys to perform their respective functions while in command mode.

To return to edit mode, press the SWAP function key. If the text buffer is currently empty, the Professional Editor enters input mode instead of edit mode to allow new lines to be entered.

Note: If tabs are entered into a command line, they are not expanded and are displayed as the circle character. Tabs are normally used only in arguments for the SEARCH and CHANGE commands.

Using the Command Stack

The Professional Editor maintains a Last-In First-Out (LIFO) stack, on which commands are placed as entered. The stack has four levels, so the last four commands entered are available on the stack at any given time. Stacking of commands saves you key strokes, because a previous command may be recalled from the stack and reentered with or without modification. If you make an error in a command entry and receive an error message, you may recall the command, correct the error, and reenter the command without having to retype the entire command.

You recall commands from the stack by using the RECALL function key (F4). When you press the RECALL function key, the command at level 1 (top) of the stack, which is the most recent command, is placed in the command input area. You may modify the command by moving the cursor horizontally and typing in new data. When you are ready to enter the command, press the Enter key. The command will be executed and placed back on the stack at level 1.

You may also recall less recent commands at levels 2, 3, and 4 by pressing the RECALL key repeatedly. The first use of the RECALL key obtains the command at level 1. By pressing the RECALL key again without pressing Enter, the command at level 2 is recalled. If you continue to press the RECALL function key without pressing Enter, you recall the third and then the fourth level command. After you recall the level 4 command, the next use of the RECALL function key wraps around to level 1 again.

After you recall a command (and perhaps modify it), pressing the Enter key causes the command to

be executed. At the same time, the command is placed back on the stack at the *same* level from which it was recalled. Thus, you can execute a sequence of up to four commands repeatedly with or without modification by using the RECALL function key and the Enter key.

Command Syntax

Each command in this chapter is shown with its format. The format descriptions follow these rules:

- Words in capital letters are *keywords* and must be input as shown. You may enter them in any combination of uppercase and/or lowercase. The Professional Editor will convert keywords to uppercase.
- You must supply any items in lowercase *italic* letters. (For example, *filename*).
- Items in square brackets ([]) are optional.
- When two or more items are separated by a vertical bar (|), you must select only one of the items. (For example, most of the commands can be entered as either the entire word or as a 1-3 character abbreviation.)

BOTTOM Command

Purpose: The BOTTOM command positions the file to the end of text within the current segment.

Format: BOTTOM|B

Remarks: The command is terminated at the bottom of text and does not continue into the next segment.

BOUNDS Command

Purpose: The BOUNDS command defines left and right boundaries which limit the scope of SEARCH and CHANGE operations.

Format: BOUNDS|BOU [*lbound* |OFF] [*rbound*]

Remarks: *lbound* is the column number of the left boundary. *rbound* is the column number of the right boundary. Valid values are $1 \leq lbound \leq rbound \leq 140$. Default values of 1 and 140 are used when no boundaries have been defined with the BOUNDS command.

When a search operation is being performed for a SEARCH or CHANGE command, the left boundary is the starting point for the search within a given line. The right boundary is the ending point within the line. If a string of characters in a line is partly or completely outside of the current boundaries, it will not be found by a search operation.

Example: The command:

BOUNDS OFF

may be used to reset the boundaries to their default values of 1 and 140. This command changes the bounds status to display BN → 0:0.

CHANGE Command

Purpose: The CHANGE command searches the text buffer for a given string and replaces that string with another string. The operation is performed within the limits of any boundaries defined by the BOUNDS command.

Format: CHANGE |C /*string1*/*string2*/*n* [*lbound* [*rbound*]]

Remarks: *string1* is the search string. *string2* is the change string. *n* is the repeat count. *lbound* and *rbound* are optional local boundary limits.

The operands *string1*, *string2*, and *n* are separated by an arbitrary non-blank delimiter, shown as / above. The first non-blank character following the blank which follows the command verb (CHANGE or C) is used as the delimiter in separating the operands. The local bounds *lbound* and *rbound*, if specified, are separated from the preceding operands by one or more blanks.

The search string (*string1*) must be a non-null string and may have a maximum length of 40 characters.

The change string (*string2*) is used to replace the search string in the text. It may be a null string and may have a maximum length of 40 characters. A change argument with zero characters (a null string) replaces the search argument found in the text with nothing, deleting the occurrence from the file text. On entry, the change argument is saved along with

CHANGE Command

the search argument and may be used in future search and change operations.

The repeat count n may have a value of 0 through 65535. If you do not enter a repeat count, a default value of zero is assumed. You can set the repeat count to an infinite value by using the character *. If the count is 1, the change operation looks for the first occurrence (the next occurrence) of the search argument and, if found, makes the change. If it is greater than 1, the change operation locates the next n occurrences of the search argument in the file and, if found, makes the change.

If the repeat count is omitted or 0, the search and change arguments are defined, but no change operation is performed. The SRCH X and CHNG X function keys are then enabled, allowing you to perform repeated search and/or change operations by pressing the function keys. This allows you to make selective changes throughout the file text by using SRCH to find the next occurrence. If you want to make the change, press the CHNG D key. If not, press a SRCH key to continue the scan in the desired direction.

You can specify the optional local boundaries as numeric values from 1-140. If included, these boundaries override the current boundaries established by the BOUNDS command. If *lbound* is specified but *rbound* is omitted, *rbound* defaults to $lbound + (\text{length of } string1) - 1$, thus confining the search to a string starting at column *lbound*.

CHANGE Command

When a change operation is limited by boundaries, an occurrence of the search argument is not detected if any portion of the matching string in the file text lies outside either the left or right boundaries.

If you do not specify local boundaries in a CHANGE command, the global boundaries (if any) will limit the scope of the change operation.

Specifying the Starting Position and Direction of Change

A change operation always has a starting point, which may be:

- The current position, which is the character above the cursor on the screen.
- The top of the text buffer if the change is a forward change.
- The bottom of the text buffer if the change is a backward change.

CHANGE Command

In addition to a starting point, a change operation also has a direction which may be either forward (toward the bottom of the text buffer) or backward (toward the beginning of the text buffer).

When you enter a CHANGE command, you specify the starting point and direction by using the appropriate verb or abbreviation for the CHANGE command as follows:

- | | |
|-------------------------------------|---|
| CHANGE or C or CHANGED or CD | Performed in the forward direction (downward), starting at the current position. |
| CHANGEU or CU | Performed in the backward direction (upward), starting at the current position. |
| CHANGEF or CF | Performed in the forward direction (downward), starting at the start of the text buffer. This operation changes the first occurrence if the count is 1. |
| CHANGEL or CL | Performed in the backward direction (upward), starting at the end of the text buffer. This operation changes the last occurrence if the count is 1. |

CHANGE Command

Repeating a Change Operation

Once you have defined both arguments in a CHANGE command, you can perform additional change operations by entering one of the above CHANGE commands *without any operands* or by using the change function keys.

Once you enter a SEARCH or CHANGE, the search function keys SRCH D, SRCH U, SRCH F, and SRCH L and change function keys CHNG D, CHNG U, CHNG F, and CHNG L remain enabled until you exit to DOS. After the arguments are defined, you may use them to quickly repeat a search and change operation.

To repeat the change forward (downward) from the current position, enter the command CHANGE, C, CHANGED, or CD, or press the CHNG D function key.

To repeat the change backward (upward) from the current position, enter the command CHANGEU or CU, or press the CHNG U function key.

To change the first occurrence, enter the command CHANGEF or CF, or press the CHNG F function key.

To change the last occurrence, enter the command CHANGEL or CL, or press the CHNG L function key.

CHANGE Command

Example: **CHANGE /print/type/5 1 80**

This command changes the next five occurrences of the word **print** to the word **type**, but only in columns 1-80. Occurrences outside of those boundaries are not changed.

COLUMN Command

Purpose: The COLUMN command displays a column grid which may be used to determine the column positions of characters within lines. To display the column grid, move the cursor to the command line or to the desired line in line command mode and enter the COLUMN command (COL in line command mode).

Format: COLUMN|COL

Remarks: The grid is displayed on the line above the current line and all preceding lines on the screen are moved upward one line. The column grid extends from column 1 to 140 and has the format:

<.....1.....2.....3.....4.....5.....6.....7.....8

If boundaries are in effect, they are displayed on the grid as left and right arrow symbols.

The column grid remains on the screen until it rolls off as the result of a vertical scrolling operation or until you enter a line command. You can delete the grid by using the DELETE function key. The column grid does not become part of the text file.

DOWN Command

Purpose: The DOWN command positions the file forward toward the end of text within the current segment.

Format: DOWN|D [*n*]

Remarks: *n* is optional and is the number of lines to move downward ($0 < n < 65536$). If not specified, *n* defaults to the current vertical scroll increment. The command is terminated if the bottom of text is encountered and does not continue into the next segment.

Example: **DOWN 5**

ECLOSE Command

Purpose: The ECLOSE command closes the extract file that was previously opened with the EOPEN command.

Format: ECLOSE|EC

Remarks: If any text has been written to the extract file, the file is closed in a normal manner. If no text is written to the extract file, an empty file is created.

END Command

Purpose: The END command saves the file being edited and terminates the edit.

Format: END|E

Remarks: The file being edited is written to the output file and control returns to initialization. The END command is equivalent to the END function key.

EOPEN Command

Purpose: The EOPEN command opens the extract file. The extract file is used to hold a portion of the file being edited.

Format: EOPENIEO *filename*

Remarks: *Filename* is the complete name of the extract file to be used. If the device is omitted, the current default device will be used. If the file named already exists, it is replaced by the data extracted from the current file.

Once the extract file has been opened, you use the extract line commands (E and EE) to select the data to be placed in the file. The file remains available for use by subsequent extract commands until you close it by using the ECLOSE command. If the file is still open when an END or QUIT command is entered, the file is automatically closed and saved.

FILES Command

Purpose: The FILES command displays the names of active files and their current status.

Format: FILES|F

Remarks: When you enter the FILES command, the Professional Editor responds with the message:

```
IN> filename ST> EOF R/O EIP OUT> filename EXT> filename
```

This message shows the name of the input file (IN), the current file status (ST), the name of the output file (OUT), and the name of the extract file (EXT). A name is shown only if the file is active. The indicators in the status field (ST) have the following meanings:

- EOF indicates that end of file has occurred on the input file, all data has been read, and the input file is now closed.
- R/O indicates that the input file is read-only.
- EIP indicates that the Edit in Place option is being used, and the input and output files are the same physical file.

If a new file is being created, the input filename field shows **New File**. The name of the extract file (EXT) is shown only if the file is currently open.

GET Command

Purpose: If the input file is too large to fit into the text buffer, the GET command reads a selected number of lines into the text buffer.

Format: GET|G [*nnnnn*|*]

Remarks: *nnnnn* is the number of lines to be read, and * indicates that data is to be read until the buffer reserve limit is exceeded. If no count is specified, * is assumed as a default.

If the buffer currently contains text, the new lines are added to the end of the buffer so that the file is maintained in proper sequence. If end of file has been encountered on the input file or the buffer reserve limit is currently exceeded, the command is rejected.

Example: **GET 24**

This command gets the next 24 lines (the size of one screen).

LEFT Command

Purpose: The LEFT command performs a scroll left operation by the number of columns specified.

Format: LEFT|L [*nnnnn*]

Remarks: *nnnnn* is the number of columns to scroll. If the number is omitted, the current horizontal scroll increment is used.

Example: **LEFT 5**

LINENO

Command

Purpose: The LINENO command displays the number of the current line relative to the first line in the text buffer.

Format: LINENO|LI

Remarks: The response to this command is the message:

Current line is: *nnnn*

MACRO Command

Purpose: The MACRO command defines or deletes a data macro definition during the editing session.

Format: MACRO|MAC [*key*] [P|F|D] /*macro data string*/

Remarks: *Key* identifies the keyboard key to be associated with the macro definition. It may be any of the available alternate keys A, B, . . . Z, 0, 1, . . . 9.

The second parameter specifies the macro type:

P Phrase macro

F File macro

D Existing definition is to be deleted

The *macro data string* is the data string to be used when the macro is called. This string is surrounded by delimiter characters (shown as / above), which do not appear anywhere else in the string. This convention is identical to that used in the arguments of the SEARCH and CHANGE commands. If the macro is a phrase macro, the character string is inserted whenever the macro is called. If the macro is a file macro, the character string must be a valid file specification for the file that contains the data structure to be inserted.

Refer to Chapter 8, “Using Macros,” for a complete description of these features.

MACRO Command

Examples: The following command assigns the phrase macro **PRINT USING** to the keyboard key **A** (**PRINT USING** is a BASIC programming statement):

MACRO A P /PRINT USING/

The following command assigns the phrase macro **SET VS 24** to the keyboard key **B** (to change the vertical scroll amount while editing):

MACRO B P /SET VS 24/

MARGIN Command

Purpose: The MARGIN command defines or modifies margins during the editing session.

Format: MARGIN|MAR [*leftmargin*]=|OFF] [*rightmargin*]=|OFF]

Remarks: *Leftmargin* may have a value of 0-132. *Rightmargin* may have a value of 0 or 5-137. If defined, the value of the right margin must always be at least 5 greater than the left margin. The character = may be used to leave the current value as is. OFF sets the respective margin inactive (with a value of 0).

Examples: The following command sets the left margin at 1 and the right margin at 80:

MARGIN 1 80

The following command leaves the left margin at its current value and changes the right margin to 0:

MAR = OFF

MASK

Command

Purpose: The MASK command displays and modifies the current input mask data during the editing session.

Format: MASK|MA

Remarks: When you enter the command, the column grid is displayed on line 24 of the screen, and the current mask data is displayed on line 25 immediately below. You can then modify the mask data as desired.

Press the Enter key to return to the command mode.

MERGE Command

Purpose: The MERGE command merges the contents of a file residing on diskette into the file being edited at the specified point. To use the command, first position the cursor to the line which precedes the insert point. The text from the file will be inserted *after* this line. Then, enter the MERGE command.

Format: MERGE|M *filename*

Remarks: *Filename* is the complete name (in DOS format) of the file to be merged. If the device is omitted, the current default disk is used.

If the file cannot be located, an error message is displayed, and the command is cancelled. If an error occurs while reading the file, or if the text buffer overflows before the complete file is merged, an error message is displayed and the merged text is deleted from the text buffer.

NEXT Command

Purpose: When a multi-segment file is being processed, you can use the NEXT command to write a portion or all of the current segment to the output file and to read a portion or all of the next segment into the text buffer.

Format: NEXT|N [*nnnnn*|*]

Remarks: If *nnnnn* is specified, text from the beginning of the text buffer up to and including the current line is written, and *nnnnn* new lines of text are read and appended to the end of the text buffer. This usage is logically equivalent to using the PUT command followed by the command GET *nnnnn*.

If * is specified, the entire text buffer is written out, and new text is read into the text buffer until the end of file is encountered or the buffer reserve limit is exceeded. If the segment currently in the text buffer is the last segment of the file, or if the file is not segmented, the current contents of the text buffer are written to the output file and the text buffer is cleared. Additional text may then be added to the end of the file. This usage is logically equivalent to using the PUT * c command followed by the command GET *.

NEXT Command

If no count is specified, text from the beginning of the text buffer up to and including the current line is written, and new text is read into the text buffer until the end of file is encountered or the buffer reserve limit is exceeded. This usage is logically equivalent to using the PUT command followed by the command GET *.

Example: **NEXT 24**

PUT Command

Purpose: The PUT command writes a selected number of lines from the text buffer to the output file.

Format: PUT|P [*]

Remarks: *, if included, indicates that the entire text buffer is to be written. If * is not specified, all lines from the beginning of the text buffer up to and including the current line are written. Move the cursor to the correct line before you enter the PUT command.

Example: **PUT 24**

QUIT Command

Purpose: The QUIT command terminates editing without saving the edited file.

Format: QUIT|Q

Remarks: The output file is scratched, and control returns to initialization. The input file remains unchanged.

RIGHT Command

Purpose: The RIGHT command performs a scroll right operation by the number of columns specified.

Format: RIGHT|R [*nnnnn*]

Remarks: *nnnnn* is the number of columns to scroll. If the count is omitted, the current horizontal scroll increment is used.

Example: **RIGHT 5**

SEARCH Command

Purpose: The SEARCH command searches the text buffer for a character string that matches the string specified in the command.

Format: SEARCH|S /*string*/[*nnnnn*]* [*lbound* *rbound*]

Remarks: The search is limited to columns which are within the current boundaries defined by the BOUNDS command. The search starts at the current cursor position and is terminated when a match is found or the end of the text buffer is reached. Thus, only the current segment is searched.

The *string* must be a non-null string and may have a maximum length of 40 characters. If the string is found, the cursor is set to the beginning of the line in which the string was found, and the string is highlighted.

The operands *string* and *nnnnn* are separated by an arbitrary non-blank delimiter, shown as / above. The first non-blank character following the blank which follows the command verb (SEARCH or S) is used as the delimiter in separating the operands. The local bounds *lbound* and *rbound*, if specified, are separated from the preceding operands by one or more blanks.

SEARCH Command

nnnnn is the occurrence count. It specifies the occurrence of the search argument to be found by the search operation. If *nnnnn* is 1 (or is omitted), the search operation looks for the next occurrence of the search argument. If *nnnnn* is greater than 1, the search operation locates the *n*th occurrence of the search argument in the file.

If the occurrence count is *, the search operation counts the number of occurrences found from the starting point to the bottom (or top of the file) and reports the results. The file position remains unchanged.

Specifying the Starting Position and Direction of Search

A search operation always has a starting point which may be:

- The current position, which is the character under the cursor on the screen.
- The top of the text buffer if the search is a forward search.
- The bottom of the text buffer if the search is a backward search.

SEARCH Command

In addition to a starting point, a search operation also has a direction, which may be either forward (toward the bottom of the text buffer), or backward (toward the beginning of the text buffer).

When you enter a SEARCH command, you specify the starting point and direction by using the appropriate verb or abbreviation for the SEARCH command as follows:

SEARCH or S or SEARCHD or SD Performed in the forward direction (downward), starting at the current position.

SEARCHU or SU Performed in the backward direction (upward), starting at the current position.

SEARCHF or SF Performed in the forward direction (downward), starting at the start of the text buffer. This operation finds the first occurrence if the count is 1.

SEARCHL or SL Performed in the backward direction (upward), starting at the end of the text buffer. This operation finds the last occurrence if the count is 1.

SEARCH

Command

Repeating a Search Operation

Once a search argument has been defined by a SEARCH command, additional search operations may be performed by entering one of the above SEARCH commands *without any operands* or by using the search function keys.

Once you enter a SEARCH or CHANGE, the search function keys SRCH D, SRCH U, SRCH F, and SRCH L remain enabled until you exit to DOS. After the arguments are defined, you may use them to quickly repeat a search operation.

Any local boundaries specified in the SEARCH command will be in effect for repeat search operations.

To repeat the search forward (downward) from the current position, enter the command SEARCH, S, SEARCHD, or SD, or press the SRCH D function key.

To repeat the search backward (upward) from the current position, enter the command SEARCHU or SU, or press the SRCH U function key.

To repeat the search for the first occurrence, enter the command SEARCHF or SF, or press the SRCH F function key.

SEARCH Command

To repeat the search for the last occurrence, enter the command SEARCHL or SL, or press the SRCH L function key.

You may specify the optional local boundaries *lbound* (1-139) and *rbound* (2-140). They are separated from the search operand and count by one or more blanks. If both *lbound* and *rbound* are omitted, the current boundaries defined by the BOUNDS command are used.

Examples: The command:

SEARCH /word/*

searches forward to the end of the text buffer for the number of occurrences of the string **word** and displays the message:

nnnnn **occurrences were found**

If the requested string is not found, the following message is displayed.

Search argument not found

The command:

S /print/5 1 80

searches forward for the next five occurrences of the word **print** in columns 1-80. Occurrences outside of those boundaries are ignored.

SET Commands

Purpose: The SET command may be used to change the value of various editor options during editing. It performs the following functions:

SET ASCII Character Mode:

Purpose: This command changes the ASCII character mode.

Format: SET ASCII [EXTENDED|STANDARD]

SET Audio Prompting Level:

Purpose: This command changes the level of audio prompting.

Format: SET AUDIO [0|1|2]

Remarks: The level may be 0, 1, or 2:

Level	Meaning
0	Only error message prompts are issued.
1	Only message and error message prompts are issued.
2	All acknowledgement and message prompts are issued.

SET Commands

SET Buffer Reserve:

Purpose: This command changes the current buffer reserve.

Format: SET RESERVE [*nn*]

Remarks: *nn* is the number of K bytes (K = 1024) of the text buffer to be reserved for changes made to the file during editing. The value of *nn* must be greater than the minimum buffer requirement (2K) and less than the current size of the text buffer, maximum 63K. If you use this minimum value, you will have 1024 bytes in which to add new data to the current segment before you reach the critical limit of 2048 bytes. The maximum value of *nn* is 63 (64512 bytes), or the current size of the text buffer.

Although the new value is set immediately, you will not notice its effect until you issue a GET command.

Example: SET RESERVE 6

SET Commands

SET Default Disk:

Purpose: This command changes the current DOS default diskette device.

Format: SET DISK [A|B]

Remarks: The diskette device specified must be both valid *and* installed.

SET Horizontal Scroll Increment:

Purpose: This command changes the horizontal scroll increment.

Format: SET HSCROLL|HS [*n*|H|P]

Remarks: *n* is the new value of the scroll increment ($0 < n < 72$). **H** indicates half-page (40 columns), and **P** indicates full-page (72 columns).

Example: **SET HS 72**

SET Paragraph Indentation:

Purpose: This command changes the paragraph indentation value.

Format: SET PARA [*nn*]

Remarks: *nn* is a value in the range 0-80.

SET Printer Line Spacing:

Purpose: This command changes the printer line spacing.

Format: SET LPI [6|8]

Remarks: The 6 indicates 6 lines per inch and 8 indicates 8 lines per inch.

SET Printer Page Width:

Purpose: This command changes the printer page width.

Format: SET WIDTH [80|132]

Remarks: The 80 indicates 80 characters per line (normal size characters), and 132 indicates 132 characters per line (condensed characters).

SET Commands

SET Printer Unit:

Purpose: This command changes the printer unit to be used in printing.

Format: SET UNIT [0|1|2]

Remarks: The printer unit you select must be installed for printing to occur. The default printer is unit 0.

SET Screen Mode:

Purpose: Use this command to switch between display modes.

Format: SET SCREEN [M80|B80|C80]

Remarks: The modes are:

- M80** Monochrome Display Adapter and Monochrome Display
- B80** Color Graphics Display Adapter and black and white or color display in black and white mode
- C80** Color Graphics Display Adapter and RGB color monitor in color mode

SET Commands

Both the designated display adapter and display must be installed and operational.

When this command is executed, the Professional Editor begins using the designated display and leaves the current data as is on the previous display.

Note: The Professional Editor does not support 40 character display modes.

SET Vertical Scroll Increment:

Purpose: This command changes the vertical scroll increment.

Format: SET VSCROLL|VS [*n*|H|P]

Remarks: *n* is the new value of the scroll increment ($0 < n < 25$). **H** indicates half-page (12 lines), and **P** indicates full-page (24 lines).

Example: SET VS 24

TABS Command

Purpose: The TABS command displays and modifies the current tab stop settings during the editing session.

Format: TABS|TA

Remarks: When you enter the TABS command, the column grid is displayed on line 24 of the screen, and the current tab settings are displayed on line 25 immediately below. You can then modify the tab settings as desired.

Press the Enter key to return to Command mode.

TOP Command

Purpose: The TOP command positions the file backward to the beginning of text within the current segment.

Format: TOP|T

Remarks: The command is terminated at the top of text and does not continue into the previous segment.

UP Command

Purpose: The UP command positions the file backward toward the beginning of text within the current segment.

Format: UP|U [*n*]

Remarks: *n* is optional and is the number of lines to move upward ($0 < n < 65536$). If not specified, *n* defaults to the current scroll count. The command is terminated if the top of text is encountered and does not continue into the previous segment.

Example: **UP 5**

CHAPTER 8. USING MACROS

Contents

Data Macros	8-3
Phrase Macros	8-4
File Macros	8-4
Defining Data Macros	
During Initialization	8-5
Defining a Phrase Macro	8-6
Defining a File Macro	8-7
Deleting a Macro Definition	8-7
Saving the Macro Definitions	8-7
Calling Macros During Editing	8-8
Defining and Deleting Macro Definitions	
During Editing	8-9
Examples of Data Macro Usage	8-10
Example 1 - BASIC Language	
Data Macros	8-10
Example 2 - GML Language	
Data Macros	8-11

This chapter describes the methods for defining data macros and using them during the edit session.

Data Macros

Data macros are predefined sequences of data characters or lines which you may insert into the file being edited by using the alternate alphabetic and numeric keys of the IBM Personal Computer keyboard. The data contained in a macro may be a frequently used sequence of characters, referred to as a *phrase macro*, or a data structure which contains one or more lines of data, referred to as a *file macro*.

You may define data macros during initialization, and they will be saved as part of the profile for the type of data being edited. You may also define, redefine, and delete data macros during the edit session as you require them. A data macro is always associated with a specific keyboard key. When you press that key while holding the Alternate (Alt) key down, the macro is retrieved and inserted in the text. This process is referred to as *calling a macro*.

Phrase Macros

A *phrase macro* is a string of data which may contain from 1 to 30 characters, including tab characters. When a phrase macro is called, the data string is inserted into the line being edited at the current cursor position. The character above the cursor and all characters to the right are shifted to the right to allow the insertion of the phrase.

The primary use of phrase macros is to quickly create a commonly used string of characters with a single keystroke. Since you may use phrases in commands as well as text lines, you may use phrase macros to define your own function keys. Phrase macros are also useful for using the extended ASCII characters, foreign language characters, mathematical symbols, and line graphics. Examples of phrase macros are given later.

File Macros

A *file macro* is a set of one or more lines of text residing in an external file. The macro definition is just the name of the external file. When the macro is called, the external file is merged into the file being edited after the current line.

The primary use of file macros is to create data structures in the file being edited from a prototype structure stored in an external file. This saves time, since the prototype structure can be quickly inserted and then modified as required. Examples of file macros are given later.

Defining Data Macros During Initialization

In most uses, data macros are related to the type of data being edited. It may be convenient to define a set of macros for use when editing BASIC programs, another set for assembly language programs, another for PASCAL programs, another for script, etc. In these cases, it is convenient to define these macros once and have the definitions saved as part of the edit profile file. The macros are then recalled as part of the profile each time you edit that type of data.

In order to have data macros included in the profile, you must define them during initialization using the following Data Macro Definitions Menu:

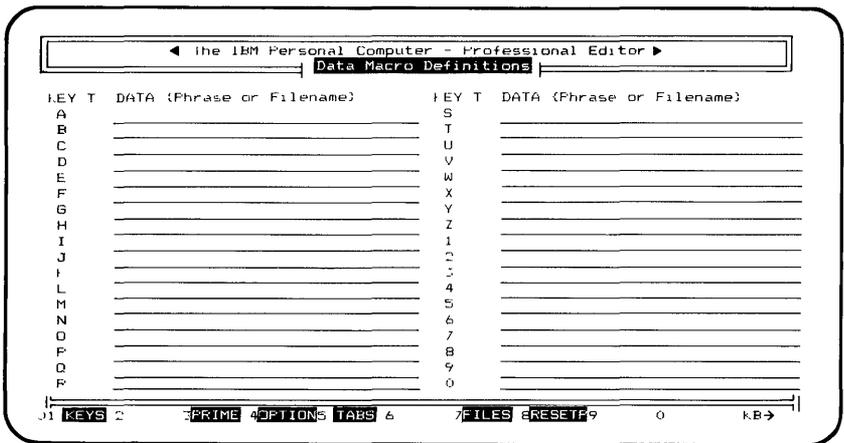


Figure 15. Data Macro Definition Menu Panel

To request this menu, press the **MACROS** function key (F6) while the Primary Options Menu is being displayed. Before doing so, you should specify the input, output, and profile files so that the existing profile (if any) will be read and any previously defined macros made available.

The menu is divided into two sets of fields on the left and right halves of the screen. Three fields are present for each possible macro definition. The first field, **KEY**, shows the keyboard key (A, B,...Z, 0,...9) associated with the macro definition to its right. The second field, **T**, shows the type of macro:

P Phrase macro

F File macro

Blank Not defined for the key.

The third field, **DATA**, shows the data associated with the macro. It is blank if the macro is not defined. This field is displayed with underscore highlighting to make the spacing of the data more readable. (The underscore does not appear on a color monitor.) If the macro is a phrase macro (P), this field contains the data string. Tab characters are not expanded and are represented by the circle character. If the macro is a file macro, the data field contains the file specification for the external file which contains the actual data.

Defining a Phrase Macro

To define a phrase macro, move the cursor to the type field (T) of the key to be used and enter the character **P**. Press the Enter key to move the cursor to the start of the data field. Enter the data characters to be inserted when the macro is called.

Leading spaces will be preserved, trailing spaces will be deleted. If you want, you may use the Alt keypad keys to define extended ASCII characters by holding down the Alt key while entering the decimal character code on the numeric keypad. When the data is correct, press the Enter key.

Defining a File Macro

To define a file macro, move the cursor to the type field (T) of the key to be used and enter the character F. Press the Enter key to move the cursor to the start of the data field. Enter the file specification for the file to be used in the form *d:filename.ext*. If you do not specify the device, the current default device will be assumed when the macro is called. No test is made to insure that the filename is correct or that the file exists until the macro is called. When the data is correct, press the Enter key.

Deleting a Macro Definition

To delete a macro definition, move the cursor to the type field and type a space over the current type. The macro definition is then deleted, though it remains on the screen.

Saving the Macro Definitions

When all macros have been defined, press the PRIME function key (F3) to return to the Primary Options Menu. When you press the EDIT function key to begin the edit session, all macro definitions are saved in the profile file.

Calling Macros During Editing

While editing a file, you may request (call) a macro by holding down the Alt key and pressing the corresponding alphabetic (A-Z) or numeric (0-9) keyboard key. If the cursor is positioned in the data area of the screen (edit mode), calling a phrase macro causes the associated data string to be inserted at the cursor position in the current line. If the insertion of the phrase causes characters to be shifted off the end of the line, you receive the message:

Line is FULL. Insert character(s) NOT accepted.

The insertion will not be made. In order to continue inserting characters, you must first delete characters from the line. This prevents the accidental loss of data from a line. Calling a file macro causes the associated external file to be merged into the text after the current line.

If the program is in command mode with the cursor positioned in the command input area, calling a phrase macro causes the associated data string to be inserted into the command. Thus, phrase macros may be used to create frequently-used commands. File macros cannot be called in command mode, and such requests are ignored.

Neither phrase or file macros may be called in line command mode with the cursor positioned in the line command input area. Such requests will be ignored.

Defining and Deleting Macro Definitions During Editing

During the edit session, you may create new macro definitions and modify or delete existing ones by using the MACRO command:

MACRO|MAC [*key*] [P|F|D] /*macro data string*/

Key is the keyboard key (A, B,...Z, 0, 1,...9) to be associated with the data macro. The second parameter specifies the type of macro:

- P A phrase macro.
- F A file macro.
- D Any existing definition should be cleared.

Macro data string is the data associated with the macro. The data must be enclosed in delimiter characters (indicated above as /). The delimiter convention is the same as is used in the SEARCH and CHANGE commands: the delimiter may be any character which does not appear in the data string.

For example, the command:

MACRO Q P /this is the Q key/

creates a phrase macro associated with the keyboard key Q, whose data is the phrase **this is the Q key**. If a macro is already defined for the Q key, it will be replaced by the new definition.

Examples of Data Macro Usage

The following two examples illustrate possible uses of data macros. The program distribution diskette contains sample profile and data files used in these examples. If you wish to experiment with them, copy the files with extensions **PRF** and **GML** to a work diskette before invoking the Professional Editor.

Example 1 – BASIC Language Data Macros

The IBM Personal Computer BASIC Interpreter allows the alternate alphabetic keys to be used to create some of the keywords of the BASIC language. This feature has been used as the basis of the following example. In addition to just creating keywords, some complete BASIC statements are created with prototype syntax. This allows you to create statements without having to look up the statement syntax in the manual. After you create the prototype statement with the macro key, you can edit it immediately to replace the prototype fields with real data.

The profile file for this example is stored on the distribution diskette as file **BAS.PRF**.

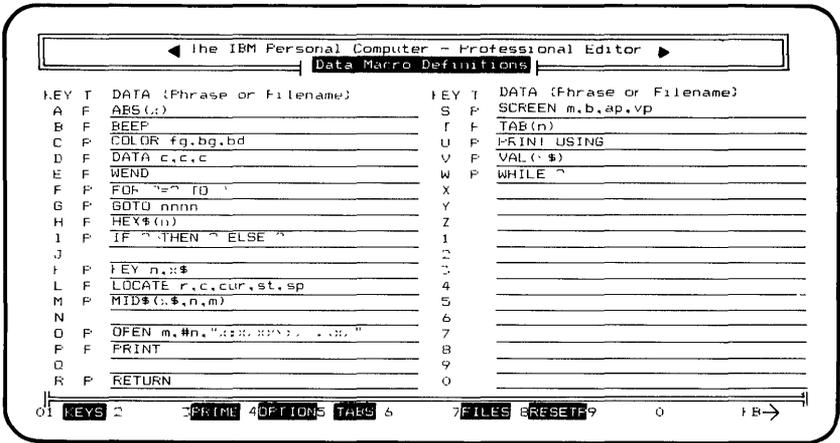


Figure 16. Example 1: BASIC Language Macros

Example 2 – GML Language Data Macros

This example illustrates how data macros might be useful to someone creating text for documents using the IBM Generalized Markup Language (GML) and SCRIPT/VS. This example assumes that you are familiar with GML tags and SCRIPT/VS commands. By defining these data macros, you can create frequently used GML tags, SCRIPT/VS commands, and document structures. Macros which create data structures are file type macros. The contents of these external files are illustrated in the figures which follow the macro definitions.

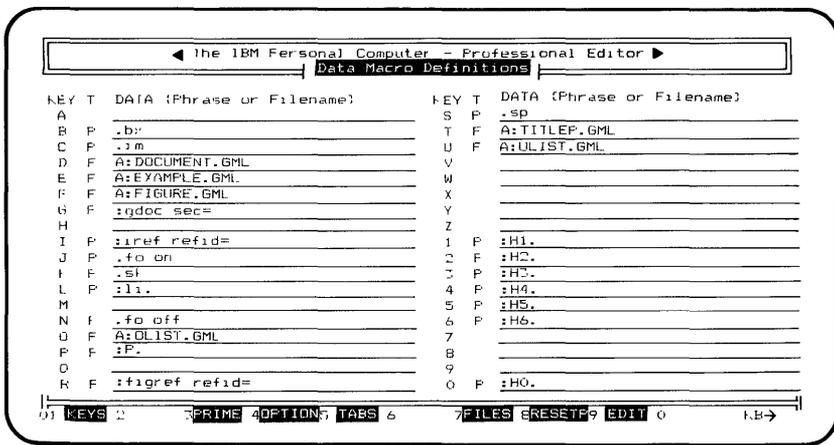


Figure 17. Example of GML and SCRIPT/VS Data Macros

The profile file for this example is stored on the distribution diskette as file SCR.PRF. The diskette also contains the GML files referred to in the various data macros. Figure 18 illustrates the structure of the external file EXAMPLE.GML. This data structure creates all of the major GML tags that define the structure of an example. This structure is created in the file being edited when you press the Alt-E key.

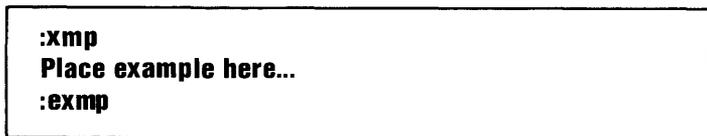


Figure 18. Sample File Macro EXAMPLE.GML

Figure 19 illustrates the structure of the external file DOCUMENT.GML. This data structure creates all of the major GML tags that define the structure of a complete document. This structure is created in the file being edited when you press the Alt-D key.

```

:GDOC SEC='???'
:FRONTM
:TITLEP
:TITLE.
:DATE.
:AUTHOR.
:ADDRESS
:ALINE.
:ALINE.
:ALINE.
:ALINE.
:EADDRESS
:ETITLEP
:PREFACE
Insert preface here...
:TOC
:FIGLIST
:BODY
.fo on
:H1.
Insert body of document here...
:APPENDIX
Insert appendices here...
:BACKM

```

Figure 19. Sample File Macro DOCUMENT.GML

Figure 20 illustrates the structure of the external file FIGURE.GML. This data structure creates all of the major GML tags that define the structure of a figure. This structure is created in the file being edited when you press the Alt-F key.

```

:fig id='?????' place='inline' frame='box'
.sp 2
.fo off
Insert the figure here...
.fo on
.sp 3
:figcap.
:efig

```

Figure 20. Sample File Macro FIGURE.GML

Figure 21 illustrates the structure of the external file OLIST.GML. This data structure creates all of the major GML tags that define the structure of an ordered list. This structure is created in the file being edited when you press the Alt-O key.

```
:ol  
:li.  
:eol
```

Figure 21. Sample File Macro OLIST.GML

Figure 22 illustrates the structure of the external file TITLEP.GML. This data structure creates all of the major GML tags that define the structure of the title page. This structure is created in the file being edited when you press the Alt-T key.

```
:TITLEP  
:TITLE.  
:DOCNUM.  
:DATE  
:AUTHOR.  
:ADDRESS  
:ALINE.  
:ALINE.  
:ALINE.  
:EADDRESS  
:ETITLEP
```

Figure 22. Sample File Macro TITLEP.GML

Figure 23 illustrates the structure of the external file ULIST.GML. This data structure creates all of the major GML tags that define the structure of an unordered list. This structure is created in the file being edited when you press the Alt-U key.

```
:ul  
:li.  
:eul
```

Figure 23. Sample File Macro ULIST.GML

CHAPTER 9. TEXT PROCESSING FEATURES

Contents

Using Margins	9-5
Defining Margins	9-6
Display of the Right Margin	9-6
Using Automatic Word Wrapping	9-7
Paragraph Indentation	9-8
Text Formatting Operations	9-9
Formatting a Block of Text	9-10
Justifying a Block of Text	9-12
Centering Lines	9-13
Removing Tab Characters	9-14

This chapter describes the text (word) processing features of the Professional Editor. The Professional Editor is not intended to be a complete word processing system. However, some simple word processing features are included which permit you to perform simple text formatting operations on a file as you edit it. These features may be used to prepare simple documents such as memos and minor reports and to print them on the IBM Matrix Printer.

With the exception of automatic word wrapping, which will be described later, you must invoke all text formatting operations explicitly. No automatic formatting of the file occurs. This gives you complete control over the formatting operations.

For applications which require complex text processing, we recommend the use of an external text formatter. The data macro facilities of the Professional Editor can be conveniently used to insert tags and control sequences for most text formatters that are capable of processing standard ASCII files.

For the purposes of the discussion which follows, several terms need to be defined.

- A *word* is a consecutive sequence of characters which are not spaces or tabs. Words are separated by the beginning of the line, spaces, tabs, and/or the end of the line.
- A *sentence* is a sequence of words. The end of a sentence occurs when the last word ends with an end of sentence character, which may be a period (.), a colon (:), a semicolon (;), or a question mark (?).
- A *paragraph* is a sequence of consecutive sentences. A paragraph begins at the first word of a block selected for formatting or following a null line. A paragraph ends at the last word of a block selected for formatting, or when a null line is encountered.
- A *null line* is a line in the file being edited which contains no characters or only space and/or tab characters followed by CR-LF.

Using Margins

You use margins in insertion and formatting operations to specify the columns of the page between which the words of text will be placed.

The *left margin* determines how much space is inserted at the left edge of each formatted line. The value of the left margin is the column in which the first word of a line should begin. A value of 0 or 1 places the first word in the first column. For values greater than 1, space characters are inserted at the beginning of the line as required. The left margin may have a value in the range of 0-132.

The *right margin* defines the right edge of the page. New lines being inserted are limited in length to the value of the right margin. Words in a formatted line will not extend beyond the right margin. If the text is being justified, the last character of the last word of each line appears in the right margin column. When you enter a value of 137, text is aligned to column 137. The right margin also controls automatic word wrapping when new lines are being entered in input mode. The right margin may have a value in the range of 0 or 5-137.

The left margin must always have a value that is at least five columns less than the right margin, and margins may never overlap. Either margin may be given a value of zero (0), which indicates that the margin is undefined. An undefined left margin behaves as though the value was 1. An undefined right margin behaves as though the value was 137.

Defining Margins

You can define margins in two ways. First, you can specify these values during initialization on the Profile Options Menu. When defined at this point, the values become part of the profile and are saved when the edit session begins. This method is described in “Selecting Options” in Chapter 4.

The second method allows you to set or modify margin values during the edit session. These values exist only for the duration of the session. These are not saved in the profile file.

You can define and modify margins during the session by using the MARGIN command:

```
MARGIN|MAR [leftmargin]=|OFF]
[rightmargin]=|OFF]
```

Leftmargin may have a value of 0-132. *Rightmargin* may have a value of 0 or 5-137. If defined, the value of the right margin must always be at least 5 greater than the left margin. You use the character = to leave the current value as is. OFF sets the respective margin inactive (with a value of 0).

Display of the Right Margin

When you define a right margin, that column in the data area of the screen is highlighted on each line of the screen. On the IBM Monochrome Display, the column is displayed in reverse video. With a color display, the column is displayed with blue characters on a red background. Margin highlighting is visible only when the corresponding column is in the display window. The margin highlighting permits you to determine the relation of the right margin to the current position in the text.

The left margin is not displayed, since it is not directly involved in text input operations.

Using Automatic Word Wrapping

Automatic word wrapping allows text to be rapidly entered and roughly formatted with respect to the right margin. When you are entering text *in input mode*, the following actions occur when you type across the right margin:

- If the character that was typed beyond the right margin is a space or tab character, the current line is terminated at the right margin and a new line is started.
- If the character that was typed beyond the right margin is not a space or tab, all characters from the last space or TAB character in the line are moved to the beginning of a new line.

This feature allows you to type at full speed without being concerned with the right margin.

For some applications (such as entering program source), you do not want to wrap words between lines. Automatic word wrapping is an optional mode. You can turn it on and off during the session as desired. When word wrapping is turned off, text is never moved to the next line. When the right margin is encountered, the current line is terminated and a new line is started with the character which caused the line to overflow.

You specify word wrapping as on or off on the Profile Options Menu. The value is then stored as part of the profile. During the session, automatic word wrapping may be toggled from on to off or from off to on by pressing the WRAP function key (F15).

Paragraph Indentation

The paragraph indentation parameter allows you to specify how many spaces the first word of a paragraph will be shifted to the right of the left margin. These spaces are then inserted when the text is formatted. This value is used only during formatting operations and has no effect on text entry and editing operations. A value of zero (0), causes no indentation in the first line of a paragraph.

You can specify the value of the paragraph indentation parameter on the Profile Options Menu, where it becomes a part of the profile. During the edit session, you can change the value by entering the command:

SET PARA [*value*]

where *value* may be a number between 0 and 80.

Text Formatting Operations

Text formatting operations permit words and sentences in consecutive lines of a file to be grouped together and spaced to form paragraphs. Two types of formatting are available: simple and justified.

In *simple* formatting operations, words are grouped within lines so that the first word of a line begins in the left margin column and the last word of a line never exceeds the right margin. A single space is inserted between each word and a double space is inserted between sentences. Multiple spaces between words in the original text are removed.

The first line of the block is treated as the first line of a paragraph. A null line appearing within the block of text being formatted causes the current paragraph to be ended and a new paragraph to be started with the next non-null line. The first line of a paragraph is indented to the right of the left margin if paragraph indentation is specified.

In *justified* formatting, additional spaces are added as required between the words of a line so that the last character of the last word appears in the right margin column. This produces smooth right and left margins, often desired in more formal documents.

Figure 25 and Figure 27 show examples of simple and justified formatting. Text formatting operations are performed using line commands which indicate the beginning and end of the group of lines to be formatted.

Note: If any line in the block being formatted contains a word that is longer than the space between the left and right margins, the line is left unformatted and an error message is displayed.

All of the following are formatted with the left margin = 4, right margin = 43, and paragraph indentation = 2.

Formatting a Block of Text

You can perform simple formatting on a block of text using the line commands:

- | | |
|-----------------|--|
| F nnnnn | Format the next <i>nnnnn</i> lines |
| F -nnnnn | Format the previous <i>nnnnn</i> lines |
| F * | Format to the end of the text |
| FF | Denotes the starting and ending lines of the block to be formatted |

```

..... X This is the left margin column X
..... This is the right margin column X
..... X
FF This example shows how X
..... unformatted text is formatted and or justified X
..... using the line commands X
..... F, FF, J, and JJ. X
..... The left and right margin columns are shown above. On X
..... the actual display screen, the X
..... right margin column would appear highlighted. X
..... Note that when JJ is used, extra spaces are inserted between X
..... words to align each line with the right margin. X
..... This is the last sentence of the first paragraph. X
..... X
..... This is the first sentence of the second paragraph. X
..... Note that the paragraphs are separated by a null line. X
..... After formatting, X
..... the first line of each paragraph will be indented an additional X
FF 2 columns since the FARA parameter is set to 2. X
..... X
..... This data is not included in the block and will remain X
..... unformatted. X
..... X
..... X
01 KEYS 2 SWAP 3 LINE 4 5 INSERT 6 DELETE 7 CANCEL 8 9 0 1B →

```

Figure 24. Example of Text Prior to Simple Formatting

```

..... X This is the left margin column X
..... This is the right margin column X
..... X
..... This example shows how unformatted X
..... text is formatted and or justified using X
..... line commands F, FF, J, and JJ. The X
..... left and right margin columns are shown X
..... above. On the actual display screen, X
..... the right margin column would appear X
..... highlighted. Note that when JJ is used X
..... extra spaces are inserted between words X
..... to align each line with the right X
..... margin. This is the last sentence of X
..... the first paragraph. X
..... X
..... This is the first sentence of the X
..... second paragraph. Note that the X
..... paragraphs are separated by a null line. X
..... After formatting, the first line of each X
..... paragraph will be indented an additional X
..... 2 columns since the FARA parameter is X
..... set to 2. X
..... X
..... This data is not included in the block and will remain X
11 KEYS 2 3 END 4 QUIT 5 WRAP 6 7 SPLIT 8 MASK 9 SCROLL 0 INDENT KB →

```

Figure 25. Example of Text After Simple Formatting

Justifying a Block of Text

You can justify a block of text using the line commands:

- J nnnnn** Justify the next *nnnnn* lines
- J-*nnnnn*** Justify the previous *nnnnn* lines
- J*** Justify to the end of the text
- J-*** Justify from the beginning of text
- JJ** Denotes the starting and ending lines of the block to be justified

```
..... X. This is the left margin column
..... This is the right margin column X
..... X
..... JJ This example shows how X
..... unformatted text as formatted and or justified X
..... using the line commands X
..... F, FF, J, and JJ. X
..... The left and right margin columns are shown above. On
..... the actual display screen, the Y
..... right margin column would appear highlighted.
..... Note that when JJ is used, extra spaces are inserted between
..... words to align each line with the right margin.
..... This is the last sentence of the first paragraph.
..... X
..... This is the first sentence of the second paragraph.
..... Note that the paragraphs are separated by a null line.
..... After formatting, X
..... the first line of each paragraph will be indented an additional
..... 2 columns since the FARA parameter is set to 2.
..... JJ
..... This data is not included in the block and will remain
..... unformatted. X
..... X
..... X
..... X
01 KEYS 2 SWAP 3 LINE 4 5 INSERT 6 DELETE 7 CANCEL 8 9 0 FB→
```

Figure 26. Example of Text Prior to Justified Formatting

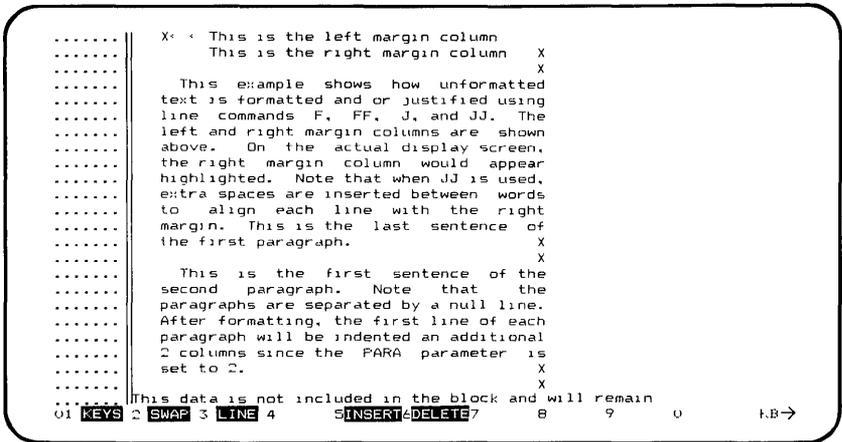


Figure 27. Example of Text After Justified Formatting

Centering Lines

You may want to center one or more lines of text so that equal blank space is provided at the beginning and end of each line. Centering is performed with respect to the left and right margins. Before a line is centered, it is formatted to provide uniform spacing between words. When a block of consecutive lines is centered, each line is treated independently, and words are not normally moved from one line to another. However, if a given line exceeds the right margin, it is split into multiple lines which are each centered respectively.

You can center a block of text using the line commands:

- K nnnnn** Center the next *nnnnn* lines
- K -nnnnn** Center the previous *nnnnn* lines
- K *** Center to the end of the text
- K -*** Center from the beginning of text
- KK** Denotes the starting and ending lines of the block to be centered

Removing Tab Characters

In some circumstances, you may want to remove all tab characters from a portion of text while retaining the present column alignment. You use the line commands **T** and **TT** to replace tab characters with spaces and insert any additional spaces required to maintain the current column alignment of the data.

You can remove tabs by using the line commands:

T nnnnn	Remove tabs from the next <i>nnnnn</i> lines
T-<i>nnnnn</i>	Remove tabs from the previous <i>nnnnn</i> lines
T*	Remove tabs to the end of the text
T-*	Remove tabs from the beginning of text
TT	Denotes the starting and ending lines of the block from which tabs are to be removed

CHAPTER 10. RECOVERING FROM ERROR SITUATIONS

Contents

Using the Backup File Option	10-3
Recovering from Disk I/O Errors	10-5
Recovering from Disk Full Conditions	10-9
Switching Drives (Option 1)	10-10
Changing Diskettes (Option 2)	10-11
Ending the Session (Option 3)	10-11
Picking Up the Pieces	10-11
Combining the Pieces with DOS	10-12
Errors During Print Operations	10-13

This chapter describes the methods for recovering from several types of errors which may be encountered during an editing session. To provide protection for your important files, we highly recommend that you make backup copies of such files on another diskette at frequent intervals.

Using the Backup File Option

You use the backup file option to maintain on the same diskette a backup copy of the file being edited. You select the backup file option at the beginning of a session on the Profile Options Menu. Unlike most other options, the backup file option may only be set during initialization and may not be changed during the session.

Use the backup file option only when the output file being created during the session will replace the input file to that session. The input and output filenames are thus identical, and the files will reside on the same diskette.

When you select the backup file option, the Professional Editor retains a copy of the original input file on the same diskette when the output file is saved with the END command. The Professional Editor does this by renaming the extension of the input file to BAK before the output file is saved. After the backup file is renamed, the output file is renamed to the final filename and extension. If any problems occur

during this process, the new output file is left with a file extension of TMP. This allows you to recover both files, if necessary, by copying them to another diskette.

When a session begins, any currently existing backup file (extension BAK) on the same diskette is deleted so that the space can be reused. If you end the session with a QUIT command, the backup file is gone. The penalty for using the backup file option is that twice as much disk space is needed to store both the backup and current copy of the file.

Recovering from Disk I/O Errors

Disk I/O errors may occur during an editing session for a number of reasons. Depending on the cause of the error, it may be possible to recover from the error and continue with the session. If recovery is not possible, the session ends as though a QUIT command had been entered. The output file is lost, but the input file remains intact.

When a disk I/O error occurs, the following is displayed on lines 24 and 25 of the screen:

DISK ERROR *operation* **File** *filename* **Area** *n* **Type** *description*
Reply R to Retry, P to Process Error

Operation describes the operation being performed: Reading or Writing. *Filename* gives the device and name of the file involved. *Area* describes the diskette data area involved as follows:

- 0 The error occurred in a system file. This should not occur during an edit session.
- 1 The error occurred in the diskette file allocation table (usually when a file is being opened).
- 2 The error occurred in the diskette directory (usually when a file is being opened or closed).
- 3 The error occurred in the data area of the file (usually when a file is being read or written).
- 4 The error occurred in the file allocation table in memory.

Description shows what type of error occurred:

- **Write Protected** – The diskette write protect tab is installed, and a file on the diskette is being opened for output. If you wish the diskette to be written, remove the tab and reply **R**. If not, end the session by replying **P**.
- **Drive NOT Ready** – The diskette drive is not ready (the door is open or no diskette is installed). Ready the drive and reply **R**, or if the drive has failed, reply **P** to end the session.
- **Data Error, Seek Error, Sector Not Found, Write Fault, and General Disk Failure** – You have either a media or drive failure. Reply **R** to retry, or **P** to process the error.

If you reply **R** to the request, DOS retries the operation. If the error is corrected, the session continues. If not, you receive another message.

If you reply **P** to process, the Professional Editor processes the error and takes appropriate action. The following table lists the actions which will be taken when you reply **P**.

The Professional Editor File Error Recovery Actions		
Input	Reading	The read operation ends at the current point. The session continues, but it is not possible to read any more text from the file.
Output	Writing	The output file is closed at its present position. The session ends as though a QUIT command had been issued.
Merge	Reading	The merge operation ends at the current point. The session continues.
Extract	Writing	The extract operation is terminated at the current point. The session continues.
Checkpoint	Writing	The checkpoint is not written. When the session ends, control returns to DOS instead of to the Primary Options Menu.
	Reading	Control returns to DOS instead of to the Primary Options Menu.

Figure 28. File Error Recovery Actions

Recovering from Reopen Errors

With the exception of Edit in Place operations, all files are opened and closed around all I/O operations. If an attempt is made to reopen a file and it fails, the diskette containing the file probably has been inadvertently swapped out of the drive. When this condition is detected, the following message is displayed:

**Reopen of operation file filename FAILED.
Verify that correct diskette is in drive #.
Reply "R" to retry operation, "P" to accept error and proceed.**

Operation describes the operation being performed. **INPUT** indicates that the input file was being reopened for a read operation. **OUTPUT** indicates that the output file was being reopened for a write operation. **EXTRACT** indicates that the extract file was being reopened for a write operation.

If you reply **R** to the request, the operation is retried. If the error is corrected, the session continues. If not, you receive another message.

If you reply **P** to the message, the error is processed according to the operation being attempted, as follows:

- **INPUT/OUTPUT** operations - The edit session is ended as if a **QUIT** command had been issued.
- **EXTRACT** operations - The extract operation is ended as if an **ECLOSE** command had been issued, except no end-of-file mark is placed in the file.

Recovering from Disk Full Conditions

This section describes procedures and options that you can use when you run out of space on a diskette. These options apply only to the file being edited. If you run out of space while performing an EXTRACT operation, the operation ends with a message, and no recovery options are available.

You may encounter the out of space condition in two situations:

- During the execution of a NEXT or PUT command where text is being written to the output file.
- During the execution of an END command where all text is being written to the output file prior to ending the session.

The three recovery options for both situations are identical. If the checkpoint file resides on the same diskette, it is deleted automatically before any recovery actions are taken. The space allocated to this file can then be used.

When the out of space condition is detected while writing to the output file, the following Recovery Options Menu is displayed at the bottom of the screen. (At the time the menu is displayed, the current output file has been closed at its current position.)

Changing Diskettes (Option 2)

Option 2 permits you to remove the current diskette and insert a new formatted diskette in the same drive. If the input file resides on the current diskette, option 2 is available *only* if all data has been read from the input file. When you select option 2, you receive the message:

Press the ENTER key when new diskette has been inserted.

Insert the new diskette and press the Enter key. A new file is opened on the new diskette, and the session continues.

Ending the Session (Option 3)

Selecting option 3 immediately ends the edit session. Any file data left in the text buffer is lost.

Picking Up the Pieces

If you select Option 1 or 2, the output file from the session consists of two files residing on separate diskettes. If the same filename was used for both input and output files, then both output files are left with an extension of TMP. Before proceeding, use the DOS RENAME command to rename the files to an extension other than TMP. For example, rename the first part of the file to PT1 as follows:

RENAME A:filename.TMP A:filename.PT1

Then, rename the second part of the file to PT2 as follows

RENAME B:filename.TMP B:filename.PT2

Combining the Pieces with DOS

In order to combine the files, you must have enough room to hold both the pieces and the final output on a diskette. If necessary, copy the pieces onto a freshly formatted diskette. Be sure that you have enough space on the diskette to contain the final output file.

If you are using DOS Version 1.10, you can use the COPY command to combine the pieces into a single file. For example, you would enter the following DOS command:

COPY *filename.PT1*+*filename.PT2*

If you are using DOS Version 1.00, then you can use the Professional Editor, entering *filename.PT1* as the name of the input file, and specify the combined file as the output file.

If the second piece is larger than the amount of free buffer space, use the PUT command to write the buffer to the output file as follows:

PUT *

With either version of DOS, use the MERGE command to read in the second piece as follows:

MERGE A:*filename.PT2*

Save the file by entering the END command. Re-edit the file, and repair any lines that may have been split between the two pieces with the Gather line command. If the size of the second piece is greater than the amount of text buffer space, you cannot use the MERGE command to combine the pieces. In this case, you must use the DOS 1.10 procedure described above to combine the files.

Errors During Print Operations

If the printer becomes not ready during a print operation, a time-out occurs and one of the following messages is displayed:

Printer TIME-OUT. Reply: R to retry or C to cancel.

or

Printer is OUT of paper. Reply: R to retry or C to cancel.

After correcting the problem, reply **R** to continue with the print operation. If the error cannot be corrected, reply **C** to cancel the print operation.

APPENDIXES

Contents

APPENDIX A. MESSAGES A-3

APPENDIX A. MESSAGES

This appendix lists all messages produced by the Professional Editor, describes the conditions where they occur, and suggests corrective actions when appropriate. Messages are listed in alphabetical order.

The Professional Editor messages are displayed at the bottom of the screen in line 25, unless otherwise noted. After a message is displayed, it remains on the screen until you press another key.

Already at END of text. Request cancelled.

A search down operation was requested and the current starting position is at the end of the text buffer. Scroll up or move the cursor up and repeat the operation.

Already at START of text. Request cancelled.

A search up operation was requested and the current starting position is at the start of the text buffer. Scroll down or move the cursor down and repeat the operation.

Block boundary extends past end of text.

The destination sub-command just entered is within the data block being manipulated. The sub-commands remain pending, and you must delete the destination sub-command (by typing spaces over it) and place it on a valid line.

Block sub-commands OVERLAP.

The destination sub-command previously entered is within the data block being manipulated. All sub-commands remain pending and you must delete the destination sub-command (by typing spaces over it) and place it on a valid line.

CAUTION: Conflicting options have been CHANGED.

This message indicates that the program has discovered one or more options which are in conflict with each other or with the filename options. Conflicts are resolved by changing the option. For example, if an output filename is specified and the READ ONLY option is selected, a conflict is detected. The Professional Editor resets the READ ONLY option to NO and leaves the filename as is.

Change argument is INVALID.

The change argument string in a CHANGE command is not recognized. This generally indicates a syntax error. The command is ignored. You may recall the command and make corrections.

Change argument is NOT defined.

A CHANGE command with no operands was entered and no change argument is currently defined. Enter a CHANGE command with search and change arguments first.

Change argument NOT found.

A CHANGE operation failed to locate the argument to be changed in the text. No changes were made. If **Only current segment processed** is added to the end of this message, then a multi-segment file is being processed, and only that portion of the file in memory was processed.

Change argument is TOO LONG.

The length of the change argument in a CHANGE command is greater than 40 bytes. The command is ignored. You may recall the command and make corrections.

Checkpoint failed, recycle NOT AVAILABLE. Press any key to continue.

An I/O error or disk full condition was encountered while writing the checkpoint file. The recycle option is cancelled and the session continues.

Command is NOT recognized.

The verb of the command just entered is not recognized. The command is ignored. You may recall the command and make corrections.

Complete file has been read into buffer.

This message is displayed at the beginning of a session to inform you that all data from the input file has now been loaded into the text buffer.

Complete file has now been read.

This message is displayed after a GET command is executed to inform you that all data from the input file has now been loaded into the text buffer.

Count NOT allowed with this sub-command.

The sub-command just entered does not permit the use of a count, such as a block command (CC, DD, etc.). The command is ignored.

Current line is *nnnnn*.

This message is the response to the LINENO command and gives the number of the current line.

Destination is ALREADY defined.

A destination sub-command (A or B) was just entered and a previously entered destination sub-command is still pending. The sub-command is ignored.

Destination NOT allowed with this line command.

The line command currently pending does not use a destination, such as delete (D or DD). The destination sub-command just entered is ignored.

Destination OVERLAPS data block.

The destination sub-command just entered is within the data block being manipulated. The sub-commands remain pending, and you must delete the destination sub-command (by typing spaces over it) and place it on a valid line.

Directory ERROR on drive α , file may be LOST!!

A directory error has occurred while the program was trying to close either an extract file or the output file. If you have changed the diskette on which the file resided while the file was open, data on the diskette currently in the drive may have been destroyed.

Directory full, profile not written. Press any key to continue.

The directory on the diskette to which the profile file is being written is full. The profile file is not updated and the session continues. If this diskette contains your output file, we advise you to QUIT the session immediately, since there is no room to store the output file.

Directory search terminated by I/O error. Press ENTER to continue.

An I/O error has occurred while the directory was being read to display a file list. The list is terminated at the point where the error occurred and may not include all of the files requested. Press the Enter key to continue.

DISK ERROR.

See “Recovering from Disk I/O Errors,” in Chapter 10 for a complete description of this message.

Disk *x*: is currently selected.

This message is issued when a SET DISK command is entered with no operands. It tells you which diskette is the current default diskette.

Disk *x*: is now the default disk.

This message is issued when a SET DISK command is used to select a new default diskette device. It verifies that the new diskette device is now the current default.

Diskette directory on drive *x*: is FULL!!

This message occurs when a file is opened and no space exists in the diskette directory. If you are in initialization, exit to DOS and change diskettes. If the file being opened is an extract file, the open fails and the session continues.

Diskette full, profile not written. Press any key to continue.

The diskette to which the profile file is being written is full. The profile file is not updated and the session continues. If this diskette contains your output file, we advise you to QUIT the session immediately, since there is no room to store the output file.

Diskette on drive x: is FULL!!

All available free space on the diskette has been used while writing the output file or an extract file. If the file is the extract file, the extract operation is terminated and the session continues. If the file is the output file, you receive an options menu. See “Recovering from Disk Full Conditions,” in Chapter 10 for a description of the options available.

Display configuration NOT supported – Editor terminated.

The Professional Editor was not able to reset the display mode to one of the three supported modes: Monochrome 80x25, Black & White 80x25, or Color 80x25.

Extract file already exists. Want to OVERWRITE? <Y or N>

An EOPEN command has been entered, and the extract file specified already exists. Reply Y to replace the existing extract file with the new one. Reply N to cancel the command.

Extract file is NOT open.

An ECLOSE command has been entered and the extract file is not open. The command is ignored.

EXTRACT file is NOT open. Request cancelled.

An extract operation is being performed and the extract file has not been defined with the EOPEN command. The operation is cancelled. Use the EOPEN command to define the extract file, then repeat the operation.

File extension TMP is not allowed.

The file extension TMP is reserved for use by the editor and may not be used. To edit the file, rename the extension.

File has been modified. QUIT anyway? <Y or N>

This message is displayed when you enter a QUIT command after modifying the file being edited. Reply Y to discard the changes and end the session. Reply N to cancel the command and continue the session.

File has been MODIFIED. QUIT no longer possible.

This message occurs when a QUIT command is entered while using the Edit in Place option, and data has already been written to the file. The command is ignored. You must use the END command to end the session.

**File has NOT been modified. SAVE anyway?
<Y or N>**

This message is displayed when you enter an END command without modifying the file being edited. Reply **Y** to save the file and end the session. Reply **N** to cancel the command and continue the session.

File is READ-ONLY. Cannot be written.

A PUT or NEXT command has been entered and the Read Only option is in effect. The command is ignored.

File is READ-ONLY. Use QUIT command to terminate.

An END command has been entered and the Read Only option is in effect. The command is ignored.

Filename must be UNIQUE.

An ambiguous filename has been entered in a menu field or as the operand of a command. A unique filename is required. Ambiguous names may designate more than one file, such as *.ASM, or PROG*.BAS. The command is ignored. You may recall the command and make corrections.

Filename REQUIRED with this command.

A MERGE or EOPEN command has been entered without including the filename as the first operand. The command is ignored. You may recall the command and make corrections.

File size exceeds available memory. Cannot MERGE.

A MERGE command has been entered which specified a file which is larger than the size of the text buffer. The command is ignored.

File size exceeds buffer. First segment read.

This message is displayed at the beginning of a session to inform you that the input file is larger than the available space in the text buffer. After editing the first segment, use the GET and PUT commands or the NEXT command to write out the buffer and read in the next segment.

File size is > 64K bytes. Cannot MERGE.

A MERGE command has been entered and the requested file is larger than 64K (65536) bytes. This file can never be merged. The command is ignored.

File specification is NOT defined.

An invalid file specification was entered in the LIST FILES option field of the Primary Options Menu. The FILES function key has been pressed before correcting the erroneous specification. Correct the file list specification and press the FILES function key again to display the file list.

Formatting terminated by buffer space shortage.

The amount of buffer free space has been reduced below the critical minimum during a format, justify, or center operation. The operation ends at the current line.

**GATHER will exceed maximum line length.
Request cancelled.**

A gather operation for a block will produce a line whose length is greater than 140 bytes. The line operation is cancelled.

Input file is at END OF FILE.

A GET command has been entered and there is no more data in the input file. The command is ignored.

Input file is NEW. Cannot be read.

A GET command has been entered and the input file is a new file. The command is ignored.

Input file is READ-ONLY, cannot modify.

A command or other operation which will modify the file text has been attempted and the Read Only option is in effect. The command or operation is ignored.

INVALID audio level given.

The value of the audio level in the Profile Options Menu is not 0, 1, or 2. The current value is retained.

INVALID buffer reserve value.

A buffer reserve value which is less than 2, greater than 63, or contains non-numeric characters was entered. The entry is ignored.

INVALID disk drive specified.

The diskette drive specified was not recognized by DOS. The only drives recognized are A: and B:. Correct the drive code and press Enter.

INVALID count value.

You entered an invalid repeat count. The count must be a number in the range 0-65535.

INVALID filename specified.

The filename entered did not conform to DOS naming restrictions. Correct the filename and press Enter.

INVALID option value. Must be "Y" or "N".

A question was answered with a value other than Y, N, YES, or NO. The entry is ignored.

INVALID left boundary given.

The value of the left boundary operand of a BOUNDS command is invalid, non-numeric, or greater than 139. The command is ignored. You may recall the command and make corrections.

INVALID left margin given.

The value of the left margin operand of a MARGIN command is invalid, non-numeric, or greater than 135. The command is ignored. You may recall the command and make corrections.

INVALID macro key specified.

The key specified in a **MACRO** command is not a valid alternate keyboard key. It must be A, B,..., Z, 0, 1,... 9. You may recall the command and make corrections.

INVALID macro type given.

The type specified in a **MACRO** command is not recognized. Use **P** to define a phrase macro, **F** to define a file macro, or **D** to delete a macro key definition. You may recall the command and make corrections.

INVALID option value reset to default.

An invalid numeric value was entered in an option field which requires a numeric value.

INVALID paragraph indent.

The value for the paragraph indentation in the Profile Options Menu is invalid, non-numeric, or greater than 80. The current value is reset to the default value of 0.

INVALID printer line spacing given.

The value of the printer line spacing in the Profile Options Menu is not 6 or 8. The current value is retained.

INVALID printer unit given.

The value of the printer unit in the Profile Options Menu is not 0, 1, or 2. The current value is retained.

INVALID printer width given.

The value of the printer width in the Profile Options Menu is not 80 or 132. The current value is retained.

INVALID right boundary given.

The value of the right boundary operand of a BOUNDS command is invalid, non-numeric, or greater than 140. The command is ignored. You may recall the command and make corrections.

INVALID right margin given.

The value of the right margin operand of a MARGIN command is invalid, non-numeric, or greater than 139. The command is ignored. You may recall the command and make corrections.

INVALID write verify option given.

The value of the write verify option in the Profile Options Menu is not 0, 1, or 2. The current value is retained.

Keyword “OFF” expected.

A BOUNDS command was entered which did not contain a value for the right boundary. The command is ignored. You may recall the command and make corrections.

Left & right boundaries OVERLAP.

The value of the right boundary specified in a BOUNDS command is less than or equal to the left boundary. Boundaries have been turned off. You may recall the command and make corrections.

Left & right margins OVERLAP.

The value of the right margin specified in a MARGIN command is less than or equal to the left margin. The command is ignored. You may recall the command and make corrections.

Line cannot be SPLIT at this point.

The SPLIT function key has been pressed with the cursor positioned at the beginning or end of the line. The request is ignored.

Line command NOT recognized.

The verb of a line sub-command just entered is not recognized. The sub-command is ignored.

Line is FULL. Insert character(s) NOT accepted.

This message occurs when characters are being inserted into a line in insert character mode or while calling a phrase macro. It also occurs when you press the Tab key and no more tabs are defined. It indicates that the line is full and the insertion of additional characters would cause data to be lost from the end of the line. In order to continue to insert characters, you must first delete some characters from the line to make room.

Macro data string is REQUIRED.

A MACRO command has been entered which defined a macro and did not include a data string. The command is ignored. You may recall the command and make corrections.

Macro data string is TOO LONG, has been truncated.

A MACRO command has been entered which defined a macro and the data string contained more than the maximum of 30 characters. You may recall the command and make corrections.

***nnnnn* changes were made.**

This message reports the number of changes that were made by a change operation. If **Only current segment processed** is added to the end of this message, then a multi-segment file is being processed, and only that portion of the file in memory was processed.

***nnnnn* lines left unformatted due to excess word length.**

During a formatting operation (format, justify, or center), *nnnnn* lines were found which contained words longer than the space between margins. These lines were left unformatted and exceed the right margin. Correct the text and repeat the formatting operation.

***nnnnn* occurrences were found.**

This message reports the number of occurrences which were found by a search for all occurrences when the operation is completed. If **Only current segment processed** is added to the end of this message, then a multi-segment file is being processed, and only that portion of the file in memory was processed.

New File. Editor has entered input mode.

This message occurs when the edit session begins and a new file is being created. It informs you that the program is in input mode ready to accept new lines of data.

Not enough free space for MERGE. Use PUT command first.

A MERGE command has been entered and the requested file is larger than the free space available in the text buffer. The command is ignored. Use the PUT command to write some of the text in the text buffer to the output file and retry the command.

**NOT enough free space for this operation.
Request cancelled.**

A line copy or move operation is being executed and there is not enough free space in the text buffer to insert the copied data. The operation is cancelled.

Null macro data string NOT ALLOWED.

The data string parameter in a MACRO command has been found to have zero length. Be sure that you have enclosed the string in delimiter characters. You may recall the command and make corrections.

Output file already exists. Want to OVER WRITE? (Y or N)

During initialization, you specified a new file, and the output file already exists or the editor created a .TMP file which you must rename. If you respond **Y**, the new file will overlay the existing file. If you respond **N**, you must then enter a new output filename.

Output file REQUIRED with NEW input file.

The input file has been specified as a new file and no name has been entered for the output file. You must enter a valid name for the output file before the edit session may begin.

OVERLAPPING left or right boundary.

A right boundary value has been entered which is less than or equal to the left boundary. The command is rejected.

Part of file has been read. More data remains.

This message occurs on the completion of a GET command to inform you that more data remains to be read from the input file. The GET command may not have read all of the data requested if the buffer reserve was exceeded.

POSITIVE value is required.

An operand of a command contained a negative value (*-nnnnn* or *-**), and a positive value is required. The command is ignored. You may recall the command and make corrections.

Printer is OUT of Paper. Reply: R to retry or C to cancel.

The printer has run out of paper during a print operation. Install more paper in the printer and reply **R**. If the printer cannot be made ready, reply **C** to cancel the print request.

Printer TIME-OUT. Reply: R to retry or C to cancel.

The printer became not ready during a print operation. Ready the printer and reply **R** to continue the print operation. If the printer cannot be made ready, reply **C** to cancel the print request.

PUT terminated by in-place overlap.

This message occurs when using the Edit in Place option. It indicates that the write position of the file has reached the current read position, and writing any more data to the file would overlay data which has not been read yet. The PUT command will be terminated at this point. Use the GET command to read more data from the input file, then reenter the PUT command. If you reach an impass, you must issue an END command to end the session.

**QUIT will NO LONGER be possible if
PUT/NEXT is executed. Proceed? (Y or N)**

A PUT or NEXT command has been entered while using the Edit in Place option. If executed, the file will be written and subsequent use of the QUIT command will be prohibited. If you wish to proceed, reply Y. If you wish to stop the command, reply N (or anything other than Y).

Recycle FAILED. Returning to DOS

The editor checkpoint file could not be found and recycle was requested. The program exits to DOS.

Rename of final output file FAILED.

An error occurred while the output file was being renamed from *filename.TMP* to the final name. The output file is left with the name *filename.TMP*. This error can only occur when the input and output files are on the same diskette and have the same name.

Rename of input file to BAK FAILED.

An error occurred while the input file was being renamed to *filename.BAK*. The output file is left with the name *filename.TMP*.

Reopen of *operation* file *filename* FAILED.
Verify that correct diskette is in drive #.
Reply “R” to retry operation “P” to accept error
and proceed.

An attempt to reopen the file *filename* for the
operation (INPUT, OUTPUT, or EXTRACT) has
failed. See “Recovering from Reopen Errors” in
Chapter 10 for recovery options.

Requested display mode INVALID.

A value other than 1, 2, or 3 has been entered for
the display mode on the Primary Options Menu.
The current display mode is retained.

Requested file NOT found.

A MERGE command has been entered and the
requested file cannot be located on the specified
diskette. The command is ignored. You may recall
the command and make corrections.

**Requested function is NOT ALLOWED on this
line.**

A function has been requested on a line which is a
special message (such as the column grid or start of
text line). The function may be a line command or
a command such as MERGE. The requested
function is not performed.

Requested scroll is OUT OF RANGE.

A scrolling command (UP, DOWN, LEFT, or RIGHT) has been entered and the count specified is either negative (*-nnnn*) or infinite (*). The command is ignored. You may recall the command and make corrections.

Right margin is not defined.

A format, justify, or center operation has been requested and the right margin has not been defined. The command is cancelled. Use the MARGIN command to define the right margin and repeat the command.

Search argument is INVALID.

The search string in a SEARCH or CHANGE command is not recognized or is of zero length (null). This generally indicates a syntax error in the command. The command is ignored. You may recall the command and make corrections.

Search argument is NOT defined.

A SEARCH or CHANGE command with no operands has been entered and no change argument is currently defined. The command is ignored. You may recall the command and make corrections by entering a SEARCH or CHANGE command with a search argument first.

Search argument NOT found.

A SEARCH operation was performed and the requested occurrence of the search string was not found before the top (or bottom) of the text buffer was encountered. If **Only current segment processed** is added to the end of this message, then a multi-segment file is being processed, and only that portion of the file in memory was processed.

Search argument NOT found. NO changes made.

A CHANGE operation was performed and the requested occurrence of the search string was not found before the top (or bottom) of the text buffer was encountered. No changes were made. If **Only current segment processed** is added to the end of this message, then a multi-segment file is being processed, and only that portion of the file in memory was processed.

Search argument is TOO LONG.

The length of the search argument in a SEARCH or CHANGE command is greater than 40 bytes. The command is ignored. You may recall the command and make corrections.

SET command option is NOT recognized.

The first operand in a SET command is not recognized. The command is ignored. You may recall the command and make corrections.

Source and destination OVERLAP.

The destination point has been defined within the block defined by a line operation. The sub-commands remain pending and you must delete the destination sub-command (by typing spaces over it) and place it on a valid line.

Sub-command ALREADY defined.

A sub-command is currently pending and another sub-command with the same verb has been entered. The new sub-command is ignored.

Sub-command verb CONFLICTS with existing command.

A sub-command is currently pending and a new sub-command with a different verb has been entered. The new sub-command is ignored.

Text buffer empty. Editor has entered INPUT mode.

The SWAP function key was pressed while the text buffer is empty. Instead of entering edit mode, the Professional Editor enters input mode so that new text may be entered.

Text buffer free space BELOW MINIMUM. NO MORE Changes.

The amount of free space in the text buffer has been reduced to less than the critical minimum value of 2048 bytes. No more changes can be made to the file until buffer space is freed. You may make more space available by using the PUT command to write part of the text to the output file or by deleting lines of text.

Text buffer is EMPTY. Request cancelled.

The text buffer is empty and an operation has been requested which refers to text (such as SEARCH), or an extract operation is being performed and there is no data in the text buffer. The request is cancelled.

TOO FEW parameters in this command.

More operands are required by the command just entered than were supplied. The command is ignored. You may recall the command and make corrections.

TOO MANY parameters in this command.

More operands were supplied in the command just entered than are used by the command. The command is ignored. You may recall the command and make corrections.

Unable to delete input file.

An error has occurred while the input file is being deleted so that the output file can be renamed. The output file is left with the name *filename.TMP*. This error can only occur when the input and output files are on the same diskette and have the same name.

Use GET command to load text.

This message is displayed when an edit session begins and the Read Input option is set to NO. The text buffer is empty and you must use the GET command to read text from the input file.

Value of SET option is INVALID.

The second operand of a SET command contains a value which is either not recognized or is invalid. The command is ignored. You may recall the command and make corrections.

WARNING *nnnnn* lines were SPLIT due to excess length. Press any key to continue.

On reading text lines from the input file or merge file, the Professional Editor has encountered *nnnnn* lines which exceed 140 bytes in length. These lines have been split at 140 bytes. You may need to fix these lines before you save the file. Press any key to continue the session.

INDEX

A

active files 7-19
active function keys 2-6
AFTER (A)
 sub-command 6-6
ASCII character
 mode 7-38
ASCII files 1-3
Audio Level option 4-20
audio operation 3-14
audio operations 4-20
audio prompting
 level 7-38
Auto Indent option 4-18
Auto Scroll option 4-19
automatic functions 3-7
automatic indentation 3-12
automatic line
 indentation 3-35
automatic modes 3-7
automatic scrolling 3-34,
 5-8
automatic word
 wrapping 9-7

B

Backspace key 3-15
Backup File option 4-15,
 10-3
BAK file extension 10-3
BASIC language data
 macros 8-10

BEFORE (B)

 sub-command 6-6
block of text 6-3
BN> 3-8
BOTTOM Command 7-6
boundaries 3-8
BOUNDS Command 7-7
Bounds option 4-21
buffer reserve 3-41, 4-21,
 7-39

C

calling macros 8-3, 8-8
CANCEL function key
 (F7) 3-32
cancelling a line
 command 6-4
CAPS LOCK mode 3-28
centering lines 9-13
CHANGE Command 7-8
changing diskettes 3-43,
 10-11
changing text 2-17, 5-9
Character Operations 3-24
character set
 restrictions 3-40
Characters per Line
 option 4-22
checkpoint file 4-16
CHNG D function key
 (F29) 2-18, 3-37
CHNG F function key
 (F28) 2-18, 3-37

- CHNG L function key (F30) 3-37
- CHNG U function key (F27) 3-36
- color display 3-5
- color graphics monitor adapter 3-5
- COLUMN Command 7-14
- column grid 7-14, 7-26, 7-44
- command input area 3-5
- command line 2-17
- command mode 7-3
- command prompt 3-7
- command stack 7-4
- command syntax 7-5
- COMMAND.COM 1-9
- commands, using 7-3
- condensed font 4-22
- control and numeric keypad 3-16
- control mode operation 3-18
- copy and repeat operation 6-10
- copy operations 6-7
- copying lines 2-21
- count operand in line command 6-5
- CR-LF character 5-12
- creating a file 2-5
- current line 3-6
- cursor display 3-10
- Cursor Down 3-22, 5-9
- Cursor Left 3-23, 5-9
- cursor operations 3-22
- Cursor Right 3-23, 5-9

- Cursor to BOL 2-8, 3-24, 5-10
- Cursor to EOL 3-24
- Cursor Up 2-8, 3-22, 5-9
- Cursor Word Left 3-23
- Cursor Word Right 3-24
- cursor, moving the 2-8

D

- data display area 2-17, 3-6
- data macros 8-3
- data mode operation 3-21
- default disk 4-12, 7-40
- default drive 4-12
- default output file 4-9
- defining your own function keys 8-4
- Del key 2-12, 5-11
- delete character 3-25
- DELETE function key (F6) 3-32, 5-14
- delete operations 6-11
- Delete to EOL 3-25
- deleting a character 2-12
- deleting a line 2-14
- deleting characters 5-11
- deleting lines 2-22, 5-14
- destination sub-command 6-6
- disk I/O errors 10-5
- diskettes, changing 3-43
- display mode, selecting 4-11
- display modes 3-5, 4-11

- display screen
 - format 3-5
 - operation 3-5
 - window 3-6
- distribution diskette 1-9
- DOS prompt 2-4
- DOWN Command 7-15
- drive, default 4-12
- drives, switching 10-10

E

- ECLOSE Command 7-16
- \$\$EDIT\$\$CKP file 4-16
- EDIT function key
 - (F9) 2-7, 4-4
- Edit in Place option 4-14
- edit mode 5-5
- edit session, starting 5-3
- editing a file 5-3
- editing lines 5-9
- editor commands,
 - using 7-3
- EDIT64 1-7
- EJECT function key
 - (F34) 3-38
- END command 5-15, 7-17
- END function key
 - (F13) 2-9, 3-33, 5-15
- end of line character 5-10
- END OF TEXT 3-8
- ending the edit
 - session 5-15, 5-16
- ending the session 10-11
- Enter key 3-16
- entering a file 2-7
- EOPEN Command 7-18
- error messages A-3
- error recovery 10-3

- errors during print 10-13
- Esc key 5-12
- Exchange Scroll 3-27
- EXE file extension 1-9
- Extended ASCII
 - option 4-18
- extract file 7-17
- extract operations 6-13

F

- features of the Professional Editor 1-4
- file
 - character set
 - restrictions 3-40
 - error recovery
 - actions 10-7
 - format 3-39
 - integrity 3-42
 - maximum line
 - length 3-39
 - multi-segment 3-41
 - operations 3-39
 - specifications 3-43
 - status 7-19
 - too big 5-4
- File List Menu 4-26
- file macros 8-4
- file position 4-15
- file specifications 3-43
- FILES Command 7-19
- FILES function key
 - (F7) 4-4
- font size 4-22
- format of screen 3-5
- formatting text 9-9
- free space 3-8
- full line condition 5-11, 8-8

- full screen editor 1-4
- function keys 3-30
- function keys on screen 2-6, 3-9
- function keys, defining your own 8-4
- functions of the Professional Editor 1-4
- functions, automatic 3-7

G

- gather operations 6-16
- GET Command 7-20
- GML language data macros 8-11
- grid, column 7-14, 7-26, 7-44

H

- hardware requirements 1-8
- highlighted line 2-8
- highlighting 3-5
- horizontal scroll 3-7
- horizontal scroll increment 4-21, 7-40
- horizontal scrolling 5-7
- HS> 3-7

I

- I/O errors 10-5
- incomplete line command 6-4

- INDENT function key (F20) 3-13, 3-35
- indentation 3-12
- indentation, paragraphs 9-8
- initialization menus 4-6
- inplace overlap 4-15
- input file 4-8
- input mask 7-26
- input mask data 4-24
- input mode 3-32, 5-6
- Ins key 2-12
- insert character 3-25
- INSERT function key (F5) 2-13, 3-32
- insert mode 2-12
- inserting a character 2-12
- inserting a line 2-13
- inserting characters 5-11
- inserting new lines 5-13
- installing the program 1-9
- invoking the Professional Editor 5-3

J

- justified formatting 9-9
- justifying text 9-12

K

- key set 0 3-31
- key set 1 3-33
- key set 2 3-35
- key set 3 3-38
- key sets 3-30

keyboard mode 3-9
keyboard mode
 selection 3-28
keyboard operation 3-15
keyboard status
 display 3-29
KEYS function key
 (F1, F11, F21, F31) 3-31,
 4-3

L

labels, line 25 3-30
LEFT Command 7-21
left margin 4-16, 9-2
LIFO 7-4
line command
 area 3-10, 3-31, 6-3
 cancelling 6-4
 count operand 6-5
 destination
 sub-command 6-6
 entering 6-3
 operation
 sub-command 6-5
 syntax 6-5
line copy commands 6-7
line full condition 5-11,
 8-8
LINE function key
 (F3) 2-21, 3-31, 6-3
line length, file 3-39
line operations 3-25
line 24 3-5
line 25 2-6, 3-5
LINENO Command 7-22
Lines per Inch option 4-22
list files 4-11
logo screen 2-4
lowercase translation
 operations 6-23

M

MACRO Command 7-23
Macro Definition
 Menu 4-25
 macros 8-3
MACROS function key
 (F6) 4-4
MARGIN Command 7-25
margins 9-5
MASK Command 7-26
MASK function key
 (F18) 3-34
Mask Input option 4-18
MD> 3-7
menus
 Data Macro
 Definition 4-25
 entering data on 4-5
 File List 4-26
 initialization 4-6
 Primary Options 4-7
 Profile Options 4-13
 Tabs and Mask
 Options 4-23
MERGE Command 7-27
messages A-1
messages on screen 3-9
messages, special 3-8
minimum version of
 program 1-7
mode selection,
 keyboard 3-28
modes of operation 5-5
monochrome display 3-5
monochrome display
 adapter 3-5
move operations 6-19
moving lines 2-23
moving the cursor 2-8
multi-segment files 3-41

N

NEXT command 5-4, 7-28
non-ASCII files 1-3
normal mode
 operation 3-17
null line 3-32, 9-4
NUM LOCK mode 3-29
numeric keypad 3-16
numeric mode
 operation 3-20

O

operation
 sub-command 6-5
OPTION function key
 (F4) 2-6, 4-4
out of space
 condition 10-9
output file 4-9
output file, default 4-9

P

paragraph 9-4
paragraph indentation
 7-41, 9-8
Paragraph Indentation
 option 4-17
pending line command 6-4
phrase macros 8-3
picking up the
 pieces 10-11
Primary Options
 Menu 2-4, 4-7
PRIME function key
 (F3) 4-3
print errors 10-13

PRINT function key
 (F32) 3-38
print operations 6-21
printer
 line spacing 7-41
 page width 7-41
 unit 4-22, 7-42
profile file 4-10
Profile Options Menu 4-13
program features 1-4
program function
 keys 3-30
program versions 1-7
PUT Command 7-30

Q

QUIT command 5-16,
 7-31
QUIT function key
 (F14) 3-33, 5-16

R

Re-Cycle option 4-16
Read Input option 4-16
Read Only option 4-14
RECALL function key
 (F4) 3-31
recovering deleted
 characters 5-12
recovering deleted
 lines 2-14, 5-14
recovering from
 errors 10-3
Recovery Options
 Menu 10-10
replacing a character 2-11

required hardware and software 1-8
RESETP function key (F8) 4-4
Restore Field key 3-25, 5-10
restricted characters 3-40
RETURN function key (F10) 4-4, 5-16
return to initialization 4-16
RIGHT Command 7-32
right margin 2-6, 3-13, 4-17, 9-5
rubout key 3-15, 4-5

S

save a file 7-17
Save Scroll 3-27
screen
 format 3-5
 operation 3-5
 window 3-6
screen mode 7-42
Scroll Bottom 3-27
Scroll Down 3-26
SCROLL function key (F19) 3-34
Scroll Left 3-26, 5-7
SCROLL LOCK
 mode 3-29
scroll mode
 operation 3-19
Scroll Right 3-27, 5-7
Scroll Top 3-27
Scroll Up 3-26
scrolling 5-6
scrolling operations 3-26
SEARCH Command 7-33
sentence 9-4

SET Commands 7-40
simple formatting 9-9
software requirements 1-8
sound, use of 3-14
SP> 3-8
SPACE function key (F33) 3-38
space, free 3-8
special messages 3-8
SPLIT function key (F17) 3-34, 5-13
split lines 3-34
splitting a line 5-13
SRCH D function key (F25) 2-18, 3-36
SRCH D function key (F9) 3-32
SRCH F function key (F24) 2-17, 3-35
SRCH L function key (F26) 3-36
SRCH U function key (F23) 3-35
stack 1-4
standard font 4-22
standard version of program 1-7
START OF TEXT 3-8
starting the Professional Editor 2-4, 5-3
starting the edit session 5-3
status display 3-7
status, file 7-19
sub-command overlap error 6-6
SWAP function key (F2) 2-17, 3-6, 3-31, 5-5
switching drives 10-10
syntax, editor commands 7-5
syntax, line commands 6-5

T

- tab characters 3-10
- tab characters,
 - removing 9-14
- tab settings 4-23
- tab space 3-11
- tab stops 3-11
- Tabs and Mask Options
 - Menu 4-23
- TABS Command 7-44
- TABS function key (F5) 4-4
- temporary file 4-9
- terminating edit 5-15
- terminating the
 - Professional Editor 5-15
- text buffer 3-41
- text buffer size 4-12
- text processing 9-3
- TMP file extension 4-9, 10-4
- TOP Command 7-45
- types of files 1-3

U

- UNDELE function key (F16) 2-14, 3-34, 5-14
- undelete line 3-34
- UP Command 7-46

- update file in place 4-14
- Uppercase option 4-17
- uppercase translation operations 6-23
- uses of the Professional Editor 1-4

V

- versions of program 1-7
- vertical scroll 3-7
- vertical scroll
 - increment 4-21, 7-43
- vertical scrolling 5-6
- VS> 3-7

W

- warm start 4-16
- WI> 3-8
- width of page 9-2
- window 3-6, 3-8
- word 9-4
- word processing 9-3
- word wrapping 2-6, 9-7
- Word Wrapping
 - option 4-19
- working diskette 1-9
- WRAP function key (F15) 3-33
- Write Verify option 4-20

Reader's Comment Form

Professional Editor

6936795

Your comments assist us in improving the usefulness of our publication; they are an important part of the input used for revisions.

IBM may use and distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever. You may, of course, continue to use the information you supply.

Please do not use this form for technical questions regarding the IBM Personal Computer or programs for the IBM Personal Computer, or for requests for additional publications; this only delays the response. Instead, direct your inquiries or request to your Authorized IBM Personal Computer Dealer

Comments:



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 321 BOCA RATON, FLORIDA 33432

POSTAGE WILL BE PAID BY ADDRESSEE

IBM PERSONAL COMPUTER
SALES & SERVICE
P.O. BOX 1328-C
BOCA RATON, FLORIDA 33432



.....
Fold here

Tape

Please do not staple

Tape



International Business Machines Corporation

**P.O. Box 1328-C
Boca Raton, Florida 33432**