

Information Management Technology Library

---

---

**PS TEXT™ EDIT**

---

---

**and**

---

---

**PS TEXT FORMAT**

---

---

**User's Guide**

---

---

---

---

---

---

Part Number 058060

Tandem Computers Incorporated

---

Document History	Edition	Part Number	Product Version	OS Version	Date
	First Edition	84084	PS TEXT EDIT C00 PS TEXT FORMAT C00	GUARDIAN 90 C00 GUARDIAN 90 C00	November 1987
	Second Edition	058060	PS TEXT EDIT C00 PS TEXT FORMAT C00	GUARDIAN 90 C00 GUARDIAN 90 C00	April 1991

---

**Note** The second edition of this manual was reformatted in April 1991; no changes were made to the manual's content at that time.

New editions incorporate any updates issued since the previous edition.

**Trademarks or Service Marks**

The following are trademarks or service marks of Tandem Computers Incorporated:

Batch-Plus, CCS, CD READ, CLX, Cyclone, DB-Batch-FE, DDNAM, DNS, Dynabus+, Enable, Encompass, Enform, Envoy, Exchange, Expand, EXT, FaxLink, FOX, Guardian, Guardian 90, HLX, InfoWay, Inspect, Integrity, Integrity S2, IXF, Laser-LX, Lighthouse, Lighthouse Keeper, LYN, Measure, MHX, Multifan, NOX, No Batch, NonStop, NonStop-UX, NonStop V+, Pathmaker, PCFormat, PS Mail, PS Text, PSX, RDF, Safeguard, Safe-T-Net, SeeView, SNAK, ST-1000, SysWay, TACL, Tandem, Tandem logo, TGAL, THL, TIL, T.I.M.E., TMF, Transfer, TransWay, TSCE, TSCE-1000, TSCP, TSCP-1000, TSMS, TSMS-1000, TTSI-NET, Tunex, Twinac, Twincos, Twinpro, TXP, V8, V80, ViewPoint, ViewSys, VLM, VLX, WFLink, XL8, XL80

All brand and product names are trademarks or registered trademarks of their respective companies.

**Copyright** All rights reserved. No part of this document may be reproduced in any form, including photocopying or translation to another language, without the prior written consent of Tandem Computers Incorporated. Copyright © 1991 Tandem Computers Incorporated.

---

# Contents

---

Preface xix

---

## Section 1 Creating a Simple Document

Overview 1-1

Starting TEDIT 1-2

    If You Want to Create the File 1-3

    If You Do Not Want to Create the File 1-3

Highlight: Where You Store Your Documents 1-4

Using the Status Line 1-6

Putting Text in the New File 1-7

    Going to the HOME Location 1-8

    Moving Forward and Backward 1-8

    Moving Left, Right, Up, and Down 1-8

    Returning to Start a New Line 1-8

    Inserting Characters 1-9

    Deleting Characters 1-10

    Deleting and Inserting Lines 1-11

Highlight: Using TEDIT Function Keys 1-13

Highlight: Asking for Help 1-15

Exiting From the File 1-16

Summary 1-17

---

## **Section 2 Revising a Simple Document**

Overview 2-1

Reopening the File 2-3

Inserting Lines and Characters 2-4

Moving a Line and a Sentence 2-6

    Moving a Line 2-6

    Highlight: How TEDIT Moves Text 2-8

    Moving a Sentence 2-10

    Locating the New Position for the Sentence 2-11

    Retrieving the Sentence 2-12

Moving Words 2-13

    Highlight: Giving Written Commands 2-15

    Using Function Keys Along With a Written Command 2-15

Breaking a Line 2-18

Replacing Words and Inserting Text 2-19

Deleting a Sentence 2-20

Highlight: Using and Canceling Persistence 2-20

Adjusting the Line Width 2-21

Undoing Mistakes You Have Made 2-23

Exiting 2-24

Summary 2-24

---

## **Section 3 Customizing TEDIT for a Columned List**

Overview 3-1

Creating a New File 3-1

Typing in Columns 3-2

    Using the Ruler to Determine Column Widths 3-2

    Setting the Tab Stops 3-3

    Highlight: The Options Part of the Profile 3-3

    Tabbing to the Stops and Typing the Agenda 3-6

Moving and Copying Text 3-8

    Moving Names Into Alphabetical Order 3-8

    Emptying the Scratchpad 3-9

    Copying 3-10

Highlight: Knowing All the Available Functions 3-11

Making Words Uppercase Without Retyping 3-14

Saving and Using the Profile 3-15

    Identifying the Profile 3-15

    Saving the Profile 3-16

    Using the Profile 3-17

    Highlight: Using the Default Profile 3-18

    Deleting the Profile 3-18

Summary 3-19

---

## **Section 4 Advanced Editing Features**

Overview 4-1

Transferring Text to the New File 4-1

    Creating the File 4-2

    Opening a Second Window 4-2

    Copying a Paragraph From One Window to Another 4-4

    Closing the Second Window 4-5

Setting Up the Profile 4-6

    Setting Line Width and Tabs 4-7

    Adjusting Line Width as You Type 4-8

    Highlight: The Function-Keys Part of the Profile 4-9

    Saving the Profile 4-11

Editing the Document 4-12

    Inserting Six Blank Lines With One Command 4-13

    Highlight: Changing the Size of a Paragraph 4-14

    Breaking Lines and Inserting Sentences 4-15

    Making a List From a Series in a Sentence 4-17

    Changing End Punctuation and Finishing Up 4-18

Personalizing a Form Letter 4-20

    Searching for Words 4-21

        Highlight: Using Ranges for Searching 4-23

        Highlight: Using Patterns for Searching 4-24

    Replacing Words 4-25

Summary 4-27

---

## Section 5 Creating an Oversized Table

- Overview 5-1
- Transferring Text to the New File 5-2
  - Creating the File 5-2
  - Three Ways to Open the Second Window 5-2
  - Copying a Region From One Window to Another 5-4
  - Closing the Second Window 5-7
- Changing the Agenda Into a Table 5-9
  - Establishing Row Headings 5-10
    - Putting Two Columns Into One 5-10
    - Moving a Column 5-11
    - Highlight: Setting Up Function Keys to Move Columns 5-12
    - Reversing Lines and Realphabetizing 5-15
    - Giving a Series of Commands in One Step 5-17
  - Highlight: Three Ways to Repeat Commands 5-19
  - Highlight: Giving a Series of Commands With an OBEY File 5-21
  - Establishing Column Headings 5-22
    - Using RULER to Determine Placement 5-23
    - Horizontal Scrolling 5-24
    - Setting the Tab Stops in the Profile 5-26
    - Highlight: Using the SETPROFILE Command to Set Tab Stops 5-26
    - Using TAB and BACKTAB for Horizontal Scrolling 5-28
    - Putting in Horizontal and Vertical Rules 5-28
    - Highlight: Inserting Vertical Rules With a Function Key 5-32
- Creating and Saving the Profile 5-35
- Summary 5-37

---

**Section 6 Working With Larger Files**

Overview 6-1

Combining Files 6-2

Highlight: Turning Off Auto Word Wrap When Using RETRIEVE 6-3

Finding and Moving Large Blocks of Text 6-5

Using Section Markers 6-5

Determining the Section Marker 6-5

Marking the Sections 6-6

Moving a Section 6-7

Using Line and Column Numbers 6-9

Displaying All the Line Numbers 6-9

Displaying Individual Lines by Number 6-11

Displaying Columns by Number 6-11

Renumbering Lines 6-12

Highlight: Renumbering and Compressing Files for Storage 6-13

Copying a Range of Lines to Another File 6-14

Using Marked Positions 6-16

Summary 6-20



---

## **Section 7 Introduction to TFORM**

- Overview 7-1
- What is a Formatter? 7-2
- What TFORM Can Do for You 7-4
- Entering TFORM Commands 7-5
  - The Backslash 7-5
  - The Command Name 7-5
  - Qualifiers and Options 7-6
  - Command Shortcuts 7-7
  - Default Settings 7-7
- Getting Help 7-9
- Summary 7-11

---

## **Section 8 Designing the Page Layout**

- Overview 8-1
- Using Page Layout Commands 8-3
- Changing the Paper Size 8-4
- Setting New Margins 8-7
  - Entering the Value for Margins 8-7
  - Using Relative Margin Settings 8-11
- Determining Spacing in a Document 8-14
  - Setting Line Spacing 8-14
  - Setting the Spacing Between Sentences 8-16
  - Setting the Spacing Between Paragraphs 8-20

---

Handling Page Breaks	8-23
Starting a New Page	8-23
Starting on the Next Even or Odd Page	8-24
Keeping Lines Together	8-27
Splitting Paragraphs	8-29
Summary	8-30

---

## **Section 9 Adjusting the Text**

Overview	9-1
How TFORM Works with Your Edited Text	9-2
Joining Text	9-4
Turning Joining On and Off	9-6
Hints on Using Joining	9-8
When Joining Should Be On	9-8
When Joining Should Be Off	9-9
Justifying Text	9-10
Turning Justification On and Off	9-10
Using Joining and Justification Together	9-12
Indenting Text	9-13
Indenting the First Line of Paragraphs	9-13
Indenting a Single Line	9-16
Indenting Blocks of Text	9-16
Turning Indentation On and Off	9-19
Using Relative Values	9-21
Centering Text	9-24
Adding White Space to the Text	9-26
Summary	9-29

---

## **Section 10 Adding Headers and Footers**

Overview 10-1

Entering Titles 10-3

Changing Titles 10-8

Including Page Numbers, Date, and Time 10-10

    Displaying Page Numbers 10-10

    Displaying the Date 10-12

    Displaying the Time 10-14

Using Multiple Headers and Footers 10-16

Putting Headers and Footers on Specific Pages 10-18

    Printing on Even and Odd Pages 10-18

    Printing on the First Page 10-24

    Printing on the Current Page 10-24

    Printing After a Specified Page Number 10-24

Setting Margins for Headers and Footers 10-25

    Entering Side Margins 10-25

    Changing the Gap 10-27

Summary 10-29

---

## **Section 11 Using Special TFORM Features**

Overview 11-1

Styling Text 11-3

    Entering Styling Commands 11-3

    Using Styling Commands 11-4

---

Making Lists	11-7
Starting the List	11-7
Entering Items in a List	11-8
Ending the List	11-8
Nesting Lists	11-8
Drawing Boxes	11-10
Defining Levels in Your Document	11-12
Entering Level Headings	11-12
Printing Level Headings	11-13
Generating a Table of Contents	11-17
Producing Form Letters	11-19
Creating the Body of the Letter	11-21
Creating the Variables File	11-22
Entering the Address	11-23
Printing Form Letters	11-24
Adding Comments to a File	11-27
Summary	11-28

---

## **Section 12 Printing Your Document**

Overview	12-1
Running TFORM	12-2
Sending the Document to a Printer	12-3
Previewing the Document on Your Screen	12-4
Sending the Document to Another File	12-5
Printing Selected Pages	12-6
Shifting the Text on the Page	12-7
Summary	12-11

---

**Appendix A Command and Function Key Equivalents**

Command and Function Key Equivalents A-1

---

**Glossary** Glossary-1

---

**Index** Index-1

---

**Figures**

- Figure 1-1. Storing Your Documents 1-5
- Figure 1-2. Using Shifted and Unshifted Function Keys 1-14
- Figure 2-1. Editing a Simple Memo 2-2
- Figure 2-2. Using the Scratchpad to Move Text 2-9/10
- Figure 3-1. The Options Part of the Profile (Page 1) 3-4
- Figure 3-2. Editing the Agenda 3-7
- Figure 4-1. The Function-Keys Part of the Profile (Page 1) 4-10
- Figure 4-2. Editing the Memo to Create a Form Letter 4-13
- Figure 5-1. Defining a Region 5-5
- Figure 5-2. Editing the Agenda to Create a Table 5-9
- Figure 5-3. The Completed Table—an Expense Form 5-34
- Figure 7-1. Text and TFORM Commands in an Edit File 7-3
- Figure 7-2. Formatted Text in a Printed Document 7-3
- Figure 8-1. Using TFORM Commands for Page Layout 8-2
- Figure 8-2. Setting the Form Size 8-5
- Figure 8-3. Printed Documents in Different Page Sizes 8-6

---

Figure 8-4. Setting Margins	8-9
Figure 8-5. Printed Documents Using Different Margin Settings	8-10
Figure 8-6. Using Relative Margin Settings	8-12
Figure 8-7. Printed Document Using Relative Margin Settings	8-13
Figure 8-8. Setting Line Spacing	8-15
Figure 8-9. Printed Document Using Different Spacings	8-15
Figure 8-10. Setting Sentence Spacing	8-18
Figure 8-11. Printed Document Using Different Sentence Spacings	8-19
Figure 8-12. Setting the Paragraph Gap	8-21
Figure 8-13. How Interline Spacing Effects Paragraph Gap	8-22
Figure 8-14. Starting New Pages	8-25
Figure 8-15. Printed Results Using the NEW Command	8-26
Figure 8-16. Keeping Lines Together	8-28
Figure 9-1. Using TFORM Commands to Format Text	9-1
Figure 9-2. Three Stages of a Document	9-3
Figure 9-3. Joining Text	9-5
Figure 9-4. Setting Joining	9-7
Figure 9-5. Printed Document Using Joining	9-7
Figure 9-6. Setting Justification	9-11
Figure 9-7. Printed Document Using Justification	9-11
Figure 9-8. Using Joining and Justification	9-12
Figure 9-9. Indenting Paragraphs	9-14

---

Figure 9-10. Printed Documents Showing Indented Paragraphs	9-15
Figure 9-11. Indenting Commands in the Edit File	9-18
Figure 9-12. Unexpected Indentation in the Printed Document	9-18
Figure 9-13. Indenting Text	9-20
Figure 9-14. Indented Text in a Printed Document	9-21
Figure 9-15. Using Relative Values to Indent Text	9-22
Figure 9-16. Nested Indentation in a Printed Document	9-23
Figure 9-17. Centering Text	9-25
Figure 9-18. Centered Text in a Printed Document	9-25
Figure 9-19. Using the SPACE Command	9-27
Figure 9-20. White Space in a Printed Document	9-28
Figure 10-1. Using TFORM Commands to Enter Headers and Footers	10-2
Figure 10-2. Entering Headers and Footers	10-5
Figure 10-3. Printed Headers and Footers	10-6
Figure 10-4. Titles Carry Over in Headers and Footers	10-9
Figure 10-5. Entering Page Numbers in a Header or Footer	10-11
Figure 10-6. Displaying the Page Number in a Header and Footer	10-11
Figure 10-7. Entering the Date in a Header or Footer	10-13
Figure 10-8. Displaying the Date in a Header or Footer	10-13
Figure 10-9. Entering the Time in a Header or Footer	10-14

---

Figure 10-10. Displaying the Time in a Header or Footer	10-15
Figure 10-11. Using Multiple Headers and Footers	10-17
Figure 10-12. Printed Multiple Headers and Footers	10-17
Figure 10-13. Entering Alternating Headers and Footers	10-21
Figure 10-14. Displaying Alternating Headers and Footers	10-22/23
Figure 10-15. Changing Header Margins	10-26
Figure 10-16. Displaying Different Header and Text Margins	10-26
Figure 10-17. Changing the Header Gap	10-27
Figure 10-18. Results of Increasing the Header Gap	10-28
Figure 11-1. Using TFORM Commands for Special Effects	11-2
Figure 11-2. Styling Text	11-5
Figure 11-3. How Styled Text Looks When Printed	11-6
Figure 11-4. Creating Lists	11-9
Figure 11-5. How Lists Look When Printed	11-9
Figure 11-6. Drawing Boxes	11-11
Figure 11-7. How Boxes Look When Printed	11-11
Figure 11-8. Entering Level Headings	11-13
Figure 11-9. Level Headings Appear in Different Styles	11-14
Figure 11-10. Defining Level Headings in an Edit File	11-15
Figure 11-11. Levels Are Grouped Together	11-16
Figure 11-12. Level Headings Generate a Table of Contents	11-17



---

Figure 11-13. Printed Table of Contents 11-18  
Figure 11-14. How TFORM Produces Form Letters 11-20  
Figure 11-15. The Body File for a Form Letter 11-21  
Figure 11-16. The List File for a Form Letter 11-23  
Figure 11-17. Printed Form Letters 11-26  
Figure 12-1. Offsetting the Text 12-8  
Figure 12-2. Entering Print Offset Commands 12-9  
Figure 12-3. The Shifted Text 12-10

---

**Tables** Table 7-1. TFORM Units of Measurement 7-6  
Table 7-2. TFORM Default Settings 7-8  
Table 8-1. Commands Used in Page Layout 8-2  
Table 9-1. Commands Used in Adjusting Text 9-29  
Table 10-1. Commands Used to Add Headers and Footers 10-29  
Table 11-1. Commands Used for Special TFORM Features 11-28  
Table 12-1. Commands Used for Printing 12-11

**Contents**

---

---

---

# Preface

---

**What is  
PS TEXT EDIT?**

PS TEXT EDIT (TEDIT) is a full-screen editing program that allows you to create a document and store it on the computer. You store the document in a file, which is a particular space that exists in the computer's memory and is physically located on a disk.

TEDIT is a full-screen (or block mode) editor; with this kind of editor, you can change text anywhere on the screen at any time. You can use the editor to enter, modify, delete, and rearrange text easily. (Other types of editors—such as line editors—allow you to change text on only one line at a time, enter the line, then go on to the next line.)

---

**What is  
PS TEXT FORMAT?**

PS TEXT FORMAT (TFORM) is a text formatting program. You use a text formatter to control the printed appearance of the text stored in an edit file. TFORM is an optional program available from Tandem, which you may not have on your system. If you do not have TFORM, you can use TEDIT with another formatter that is available to you.

Some of the TFORM commands determine page layout by setting margins and text spacing. Other commands control text alignment, justification, indentation, and paragraph display. You can also use TFORM to add special touches to your text, including headers, footers, boxes, lists, and styling features such as bolding, underlining, and shadow printing.

---

**What This Manual Is  
About**

The *PS TEXT EDIT and PS TEXT FORMAT User's Guide* is a task-oriented, step-by-step guide to using TEDIT and TFORM. Part One, "Using PS TEXT EDIT" consists of Sections 1 through 6. Part Two, "Using PS TEXT FORMAT" consists of Sections 7 through 12.

**Part One: Using PS TEXT EDIT** Part One of this manual is designed to give you hands-on experience in using most of the features of TEDIT. You start with the easiest, most straightforward, and most commonly used functions and progress to the more complex functions. In this way, you develop your understanding of a concept at the time you need it to make use of a given feature. Each feature is illustrated with detailed examples showing how you communicate with TEDIT and how TEDIT responds. The examples are shown as a shaded area in the text; what you type is highlighted in white.

This manual goes into more depth than the *PS TEXT EDIT Quick Start*, but it does not discuss everything that TEDIT can do. The more complex features are covered in the *PS TEXT EDIT Reference Manual*.

**Part Two: Using PS TEXT FORMAT** Part Two of this manual describes the TFORM commands you use in everyday printing tasks. Each section covers a specific area of formatting, beginning with an explanation of the task and including a complete description of the relevant TFORM commands. Throughout the manual you'll find detailed examples showing you how to enter the TFORM commands; then you'll see illustrations of the resulting effect on a printed document.

This manual expands on the material introduced in the *PS TEXT FORMAT Quick Start*. It covers some of the TFORM commands in depth, but does not include all of the commands and their options. For information on more complex formatting tasks, or to explore all the capabilities of TFORM, see the *PS TEXT FORMAT Reference Manual*.

- How to Use This Manual** The Table of Contents, the Glossary, and the Index contain entries for both Part One, “Using PS TEXT EDIT” and Part Two, “Using PS TEXT FORMAT.” Use the Table of Contents and the Index to locate sections or pages that discuss specific functions. Refer to the tab dividers for lists of the functions covered in each section or read the overview or summary of each section for a reminder of what that section contains. Use the Glossary as a quick reference; it contains definitions for the terms shown in *italic* in the text.
- Part One: Using PS TEXT EDIT** Work through Part One of the manual from beginning to end. You use the manual this way for two reasons: 1) you accumulate the knowledge you need for later sections as you go along and 2) you create documents in the beginning that you reuse later on. The TEDIT sections are set up according to this scheme:
- In Section 1 you create a simple document—a one-paragraph memo announcing a quarterly budget meeting. You learn how to create a file, move around on the screen, use the online help, and exit from TEDIT.
  - In Section 2 you revise the memo, familiarizing yourself with the terminal keys and the TEDIT function keys, and using the command line for functions not associated with function keys. You also learn how to undo mistakes.
  - In Section 3 you create a three-column agenda for the budget meeting. You learn how to customize TEDIT to facilitate creating a columned list like an agenda.
  - In Section 4 you create a form letter, using the memo from Section 2 as rough-draft copy. The form letter gives personalized instructions to people attending the budget meeting. You use more of the customizing features of TEDIT, learn how to copy text from one file to another, and use the TEDIT search and replace features.

- In Section 5 you create a tabular expense form, using the agenda from Section 2 as the rough-draft copy. You get more practice with windows for transferring text between files, learn more advanced customizing features, and work with a file that is both longer and wider than one screen.
- In Section 6 you put the form letter, the agenda, and the tabular expense form into a single file, which simulates how you would distribute the material to the budget meeting attendees. Section 6 gives you practice in moving around in and editing a larger file.

At the back of the manual is an Appendix containing a table of command and function key equivalents for TEDIT. Use the Appendix to show you several different ways to accomplish a certain task.

The examples in Part One refer to a Tandem 6530 terminal. If you are using a different kind of terminal, you may notice that some features do not work in exactly the same way as described, although most do. Also, the examples in the manual may not look the same as what you see on your screen. Some of the keys on other terminals are labeled differently as well; for example, the `RETURN` key on the Tandem terminal is labeled `ENTER` on the IBM 3270 terminal. If you have difficulty establishing a correspondence between the functions discussed in this manual and the way your terminal works, refer to the manual that comes with your terminal or see your *TEDIT* contact about any functions you have trouble using.

**Part Two: Using  
PS TEXT FORMAT**

Part Two of this manual serves as a quick reference for the most frequently used TFORM commands. Each formatting task is explained through the use of several examples chosen to reflect the options available for the TFORM commands. You aren't expected to try out every example. Instead, find the example that meets your needs and enter that command in your text file. Part Two is divided into the following sections:

- Section 7 provides a brief introduction to text formatting and explains the design of TFORM commands. It also tells you how to use online help.
- Section 8 discusses page layout. This section presents commands for setting page size and margins, and for controlling spacing and page breaks in the printed document.
- Section 9 presents the commands you use to adjust the text display. Commands used for joining, justifying, and indenting paragraphs are covered in this section.
- Section 10 shows you how to add headers and footers to a document.
- Section 11 explains some of the special formatting abilities of TFORM. These features include styling text, creating lists, drawing boxes, and formatting form letters.
- Section 12 details the TFORM printing command. In addition to providing information on running the TFORM program, it also tells you how to preview a document on your terminal screen, send the document to another file, print selected pages, and print form letters.

When you first start using TFORM, you'll probably read through all the sections completely to familiarize yourself with how TFORM works and how you use TFORM commands. After you're more familiar with TFORM, the Table of Contents and Index can serve as fast pointers to exactly the information you need.

---

**Who Should Use This Manual**

The *PS TEXT EDIT and PS TEXT FORMAT User's Guide* is for you if you are a person who uses and produces information as part of your job. You do not need to have a technical background, be very familiar with computers, or have used a text editor or a text formatter before. The *User's Guide* is directed toward those who can spend more time and who want to learn TEDIT or TFORM in more depth than those who just want the basic overviews provided in the *Quick Starts*. The *User's Guide* assumes that:

- You know how to use your terminal.
- You are interested in exploring what TEDIT can do.
- You want to use the editor efficiently and with ease.
- You have the option of using TFORM to format your printed documents.

---

**Where to Go for More Information**

There are other sources of information about TEDIT:

- The *PS TEXT EDIT Quick Start* is a short tutorial that helps you learn the basics of TEDIT in 20 minutes.
- The *PS TEXT EDIT Reference Manual* gives you a complete view of TEDIT. The manual includes a conceptual overview, detailed descriptions of all the commands and concepts, and gives examples for each. It also includes descriptions of advice messages and how TEDIT interacts with various terminals.
- The *PS TEXT EDIT Reference Card* gives you a handy summary of TEDIT functions and commands. The *EDIT/VS to TEDIT Reference Card* compares EDIT functions to those of TEDIT.



- The *PS TEXT EDIT Independent Study Program* is a self-paced approach to learning the default TEDIT functions assigned to the sixteen function keys on the Tandem 6530 terminal. The ISP is accompanied by audio/video cassettes and includes exercises to facilitate learning the features and capabilities of TEDIT.
- The TEDIT Online Help can function as an online user's guide. It has information for new users of TEDIT about how to create a file, put text in it, and manipulate the text. If you type TEDIT at your command interpreter and then press `RETURN`, TEDIT displays a "WELCOME" screen and tells you what to do next. It also explains how to get online help on specific subjects.

There are other sources of information about TFORM:

- The *PS TEXT FORMAT Quick Start* presents a brief overview of the most frequently used TFORM commands. This 20-minute tutorial helps you learn the basics of TFORM and is especially useful if you are familiar with text formatters.
- The *PS TEXT FORMAT Reference Manual* presents a complete description of TFORM. The manual presents an overview of the TFORM program, and details the complete syntax of all TFORM commands. It also discusses using TFORM escape sequences, functions, and system parameters.
- The *PS TEXT FORMAT Reference Card* gives you a handy summary of TFORM functions and commands. The *PS TEXT FORMAT TGAL to TFORM Reference Card* compares TGAL functions to those of TFORM.
- The TFORM Online Help provides online summaries of all TFORM commands.

For information about disk volumes and files or about using printers, refer to the *GUARDIAN 90 Operating System Utilities Reference Manual*, Volumes 1 through 3.

To order more manuals or reference cards, contact your Tandem sales office.

---

# 1 Creating a Simple Document

---

**Overview** PS TEXT EDIT (TEDIT) is an *editing program* that allows you to create a *document* and store it on the computer. Each document can be stored in its own *file*. You can use TEDIT on a file that already exists or you can have TEDIT create a new file.

In this section you will learn how to:

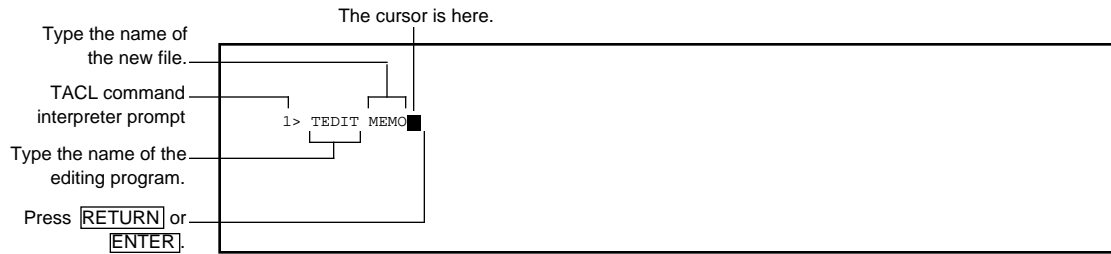
- Start TEDIT and create a new file.
- Put text in the file.
- Use your terminal's function keys.
- Use TEDIT function keys.
- Ask TEDIT for help.
- Exit from the file.

**Starting TEDIT** Before you begin, be sure the following conditions are met:

1. You are *logged on* to your terminal.
2. You have a *template* that tells you what each of the *function keys* on your terminal does. Tandem makes a template that fits your terminal. Check with your TEDIT contact to make sure you get a template that identifies the function keys you are using. A template for a Tandem 6530 terminal fits across the top of the keyboard; it identifies the *functions* associated with each of the keys labeled **F1** through **F16**.
3. You must be using the *default profile*. If you have never used TEDIT before, you automatically get the default profile when you start TEDIT. If you are not using the original, system default profile, some of the examples in this manual will look different from what you see on your *screen*.

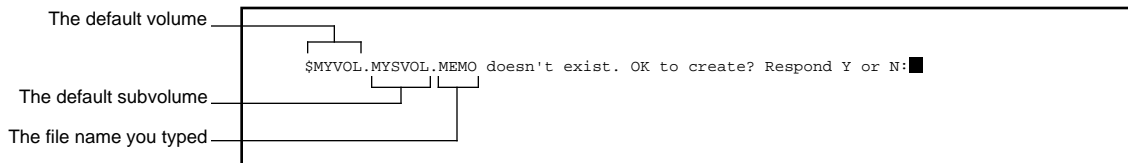
Now, let's get started.

To start TEDIT and *create* a new file named MEMO at the same time, type TEDIT followed by the file name at the *command interpreter* prompt. Then press **RETURN** if you are using a Tandem terminal. Press **ENTER** if you are using a IBM 3270 terminal. See the following example.



**Note** A file name contains up to eight characters that must be either letters or numbers. The first character must be a letter.

Your screen may blink once or twice. Then you see the following line at the top of the screen:



TEDIT asks you if it is OK to create the new file for these reasons: 1) to show you the name and location of the new file and 2) to allow you to change it in case you misspelled the file name or you want it in a different location.

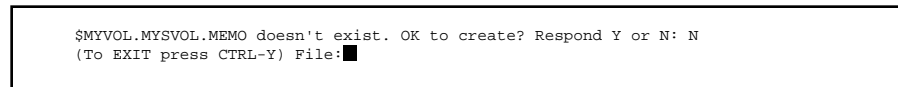
**If You Want to Create the File**

If you typed the file name correctly and want to create the file, now type Y (meaning yes) and press `RETURN`, TEDIT creates the new file in the *default volume and subvolume*, which are displayed on the line at the top of the screen.

After you type Y and press `RETURN`, TEDIT creates the new file. While TEDIT is creating the file, the screen may blink once or twice again. Then you see a blank screen with a line of information at the bottom; this line is called the *status line*. You can skip over the next part about not creating the file and go directly to “Using the Status Line.”

**If You Do Not Want to Create the File**

If you do not want the file you just named, type N (meaning no) after the colon and press `RETURN`. TEDIT then asks you to type another file name or to exit. The screen looks like this:



## Creating a Simple Document

---

### Where You Store Your Documents

If you want to leave TEDIT, hold down the `CTRL` key and type Y. EOF! appears at the cursor position (EOF stands for end of file), and TEDIT returns you to the command interpreter.

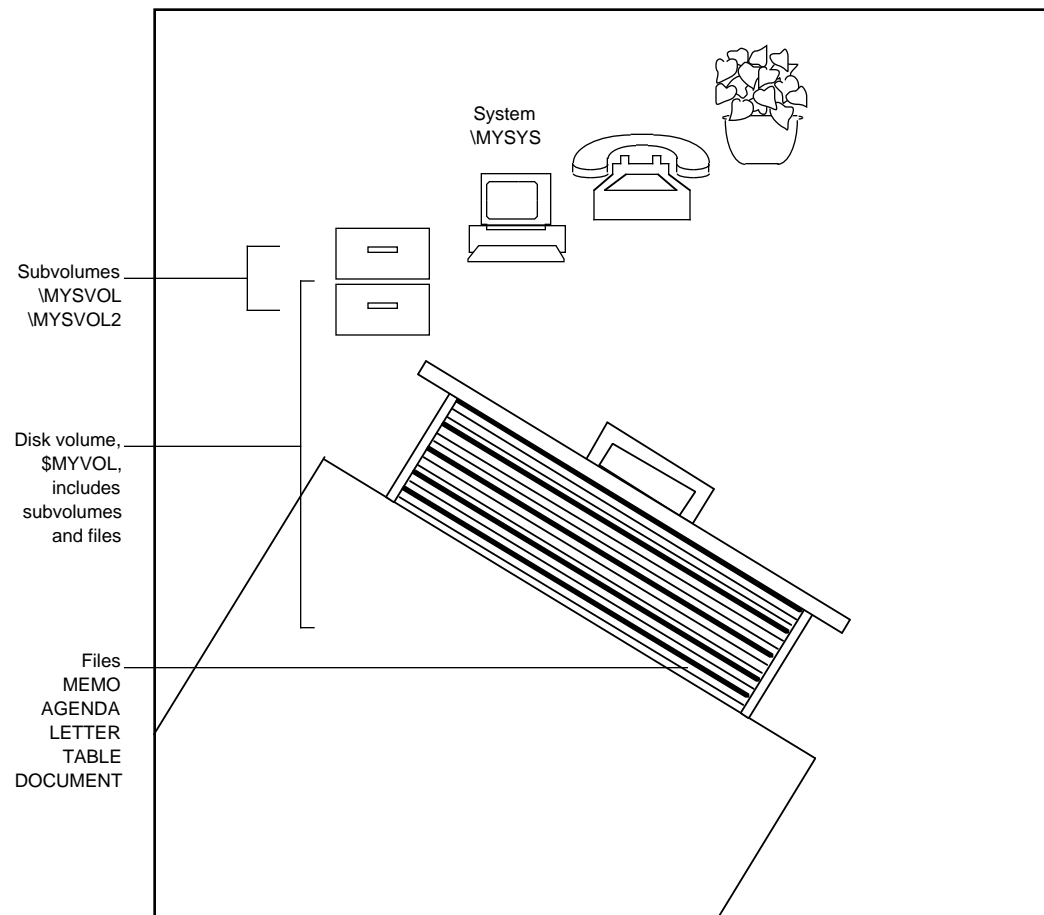
If you want to change the name of the file or the volume or the subvolume, type the new name after the word "File:" and press `RETURN`. TEDIT again shows you the file name and asks you if it is OK to create it. If you want the file, type Y and press `RETURN`.

---

### Highlight Where You Store Your Documents

Creating a file means setting up a specific location on a *disk* where you can store your document. Tandem refers to the disk as a volume. Within the volume are subvolumes, and within each subvolume are files, as shown in Figure 1-1. Naming the volume, subvolume, and file identifies a unique location. From now on you can always find the file in that same location.

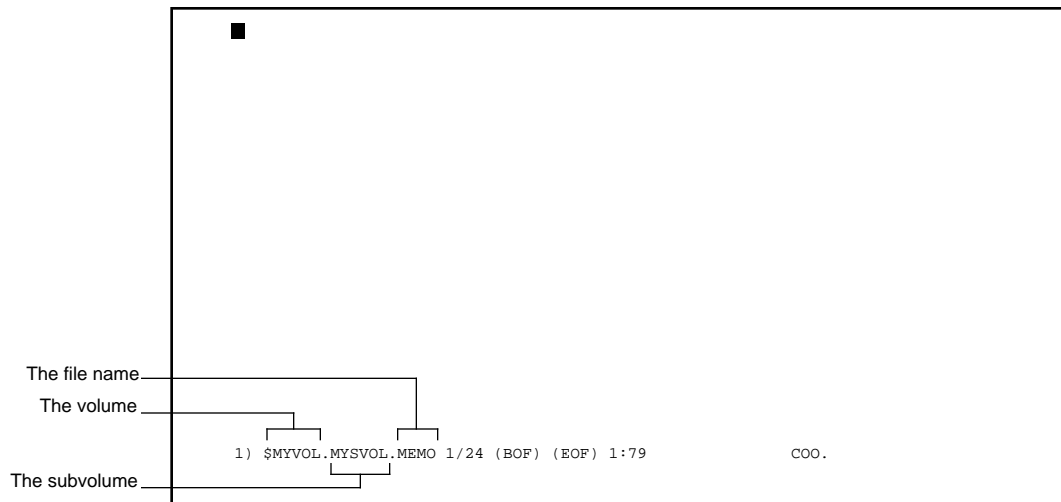
Figure 1-1. Storing Your Documents



## Creating a Simple Document

### Using the Status Line

**Using the Status Line** When you first create a new file, your screen looks like the following example. The status line at the bottom of the screen gives you information about your file. Most of the information on the status line is discussed in later sections. For now just notice that TEDIT displays the name of the file you are in.



**Putting Text in the New File**

Notice that the *cursor* is positioned at the top left corner of the screen. Now you can start to enter text. Type the lines in the following example exactly the way you see them. When you get to the end of each line, press `RETURN`. Don't worry about making mistakes. You'll learn easy ways to correct them as you go along.

```

TO: All dept heads
FROM: The Publisher

The quarterly budgeting meeting has been scheduled for March 15.
Please be prepared with the actual expenses for the last two quarters,
current budget tracking, and requests for next quarter. Meet in the
conference room at 8 A.M. Don't be late!

J.█
    
```

Now that you have entered some text, you can go back and make some changes to it. The first thing you do is move the cursor to the place where you want to make the change. Most of the editing you do is done at the cursor position. For example, you can replace one letter with another by simply typing the new letter over the old one. You can erase the letter the cursor is on by pressing the `SPACE` bar.

Many of the keys you use to move the cursor are *terminal function keys*. There are also *TEDIT function keys* (like the ones named on the template). As you read through the following paragraphs, try each one of these terminal keys as it is introduced to see how it works: `HOME`, `SPACE` bar, `BACKSPACE`, arrow keys (`→`, `←`, `↑`, and `↓`), `RETURN`, `TAB`, and `BACKTAB`.

**Note** For information about the terminal function keys not discussed here, refer to the instruction book that comes with your terminal.



**Going to the HOME Location** Use the `HOME` key to move the cursor to the top left corner of the screen.

**Moving Forward and Backward** You can move forward or backward with the `SPACE` bar and the `BACKSPACE` key or with the arrow keys. The `SPACE` bar moves the cursor to the right by inserting a blank character each time you press it. You continue to insert blanks as long as you press on the `SPACE` bar. Pressing the `SPACE` bar to move the cursor over the characters you have already typed erases the characters by replacing them with blanks.

The `BACKSPACE` key moves the same way as the `SPACE` bar, but in the opposite direction. Pressing the `BACKSPACE` key over the characters you have already typed does not erase them.

**Moving Right, Left, Up, and Down** Press an arrow key once to move the cursor one column (the space of one character) to the right or left, or one line up or down. If you press and hold the key, the cursor continues to move in the direction the arrow points for as long as you press on the key. When you reach the edge of the screen by holding down the key, the cursor *wraps* around to the opposite edge of the screen. The cursor moves over the letters you have typed; it does not change the text in any way.

**Returning to Start a New Line** Press `RETURN` to move the cursor to the far left column of the next line. If you have a Tandem 6530 terminal, you may notice that the cursor is a bit sluggish when you press the `RETURN` key. The sluggishness is due to a feature called the *RETURN function*, which is discussed in Sections 3 and 4.

There is another way to return the cursor to start a new line: Press the `TAB` key. Try it. Now press the `BACKTAB` key. `BACKTAB` is a shifted `TAB` key on Tandem terminals. The cursor moves to the far left column of the same line the cursor is already in. If you press `BACKTAB` again, the cursor moves up to the previous line. Note that the `TAB` and `BACKTAB` terminal keys do not move to any tab stops as you might expect; you use TEDIT functions, discussed later, to do that.

**Inserting Characters** Now you can go back to the heading of the memo and change it by inserting some text. The heading currently looks like this:

Put the cursor here and press **CHAR INS**.

```
TO: All dept heads
FROM: The Publisher
```

To insert an extra space after the colon in each line of the heading, follow these steps:

1. Position the cursor on the space after the colon.
2. Press the **CHAR INS** key.

Notice that the cursor does not move, but that the characters to the right of the cursor shift to the right by one column. Each time you press **CHAR INS**, TEDIT inserts a blank space where the cursor is.

Follow these steps to insert more than one character:

1. Put the cursor on the "t" of "dept."
2. Press **CHAR INS** six times.
3. Now type "artmen" in the space you made.

Your heading should now look like this:

Insert space for these letters with **CHAR INS**.

```
TO: All department heads
FROM: The Publisher
```

Another way to insert several characters in one place is to use the **SHIFT** key plus the **CHAR INS** key. Try this:

1. Put the cursor on the "T" of "The Publisher."
2. Hold down the **SHIFT** key and press **CHAR INS**. If you are using a Tandem 6530 terminal, notice that a light on the top right of the keyboard (labeled INSERT MODE) lights up.
3. Now type "Office of " and notice how the words to the right of the cursor move over as you type.
4. Press the **CHAR INS** key again to return it to an unshifted key. Type "t" on top of the "T" in "the."

The heading should now look like this:

Use **SHIFT**  
**CHAR INS** to insert  
longer phrases.

```
TO: All departmen heads
FROM: Office of the Publisher
```

**Deleting Characters** Now try deleting some characters. Follow these steps:

1. Put the cursor on the third line, the one that starts with the word "current." See the following example.

Put the cursor here and  
follow steps 2 and 3.

```
The quarterly budgeting meeting has been scheduled for March 15.
Please be prepared with the actual expenses for the last two quarters,
current budget tracking, and requests for next quarter. Meet in the
conference room at 8 A.M. Don't be late!
J.
```

Put the cursor here  
and follow step 4.

2. Press the **CHAR DEL** key. Notice that the cursor stays in the same position and all the characters to the right of the cursor move to the left by one character.
3. Now press the **CHAR DEL** key and hold it down. The terminal continues to delete characters as long as you hold down the key. Lift your finger from the key after you have deleted the word "current."

- Another way to delete characters is to press `SHIFT CHAR INS`, put the cursor at the end of the phrase you want to delete, and press the `BACKSPACE` key. Try it now. Position the cursor after the word “tracking”—on the “a” in the word “and.” Hold down the `SHIFT` key and press `CHAR INS`. Now press the `BACKSPACE` key and hold it down until you have deleted “budget tracking, ”; the text should now look like this:

```
The quarterly budgeting meeting has been scheduled for March 15. Please be
prepared with the actual expenses for the last two quarters,
and requests for next quarter. Meet in the
conference room at 8 A.M. Don't be late!

J.
```

#### Deleting and Inserting Lines

The terminal functions of delete line (`LINE DEL`) and insert line (`LINE INS`) are both on the same terminal key. `LINE DEL` is the *shifted function*; `LINE INS` is the unshifted function. Going back to the memo, try deleting and inserting some lines.

To delete one of the blank lines between the heading and the body of the memo, follow these steps:

- Put the cursor anywhere on the line.
- Press `SHIFT`, then press `LINE DEL`.

## Creating a Simple Document

### Deleting and Inserting Lines

See the next two examples. The first one shows where to place the cursor. The second one shows how the screen looks after TEDIT deletes the line. The cursor stays in the same position and all the text moves up by one line, deleting the line where the cursor was. You delete a blank line and a line containing characters in the same way.

To insert a line, put the cursor anywhere on a line and press `LINE INS`. The terminal inserts a blank line above the line that contains the cursor. The cursor stays in the same position it was in and all the text moves down by one line. Try it, following these steps:

1. Put the cursor on the line that starts with "Please."
2. Press `LINE INS`.

See the next two examples. The first one shows where to place the cursor. The second one shows how the screen looks after TEDIT inserts the blank line.

The image shows two terminal window screenshots. The top screenshot shows a terminal with the following text: "TO: All department heads", "FROM: Office of the Publisher", a blank line, "The quarterly budgeting meeting has been scheduled for March 15.", "Please be prepared with the actual expenses for the last two quarters, and requests for next quarter. Meet in the conference room at 8 A.M. Don't be late!", and "J.". A cursor is positioned at the start of the line "Please be prepared...". To the left of the terminal, text reads: "With the cursor here, press `LINE DEL`." and "With the cursor here, press `LINE INS`". The bottom screenshot shows the same terminal after the operation. The text is now: "TO: All department heads", "FROM: Office of the Publisher", a blank line, "The quarterly budgeting meeting has been scheduled for March 15.", a blank line, "Please be prepared with the actual expenses for the last two quarters, and requests for next quarter. Meet in the conference room at 8 A.M. Don't be late!", and "J.". To the left of the terminal, text reads: "Pressing `LINE DEL` deleted the other blank line." and "Pressing `LINE INS` inserted this blank line."

On the blank line you inserted, type the sentence you see in the next example.

Type this sentence on the blank line.

```

TO: All department heads
FROM: Office of the Publisher

The quarterly budgeting meeting has been scheduled for March 15.
Your attendance is required.
Please be prepared with the actual expenses for the last two quarters,
and requests for next quarter. Meet in the
conference room at 8 A.M. Don't be late!

J.
    
```

If you have typos to correct, you can correct them now, using the keys you just learned. Now you have finished all the editing you will do on the memo in this section. You will continue to edit this document in Section 2—the MEMO file will look exactly the same when you return to it. You are now ready to exit from the file. In order to exit, however, you need to know how to use TEDIT function keys. Using function keys and asking TEDIT for help are discussed next.

## Highlight Using TEDIT Function Keys

Look at the template that goes with your function keys. Locate the label for the EXIT key. Notice that the template has two labels that name the functions for each key. TEDIT performs the lower function when you press the key alone; TEDIT performs the upper function when you hold down the `[SHIFT]` key while pressing the function key. The keys and template are illustrated in Figure 1-2.

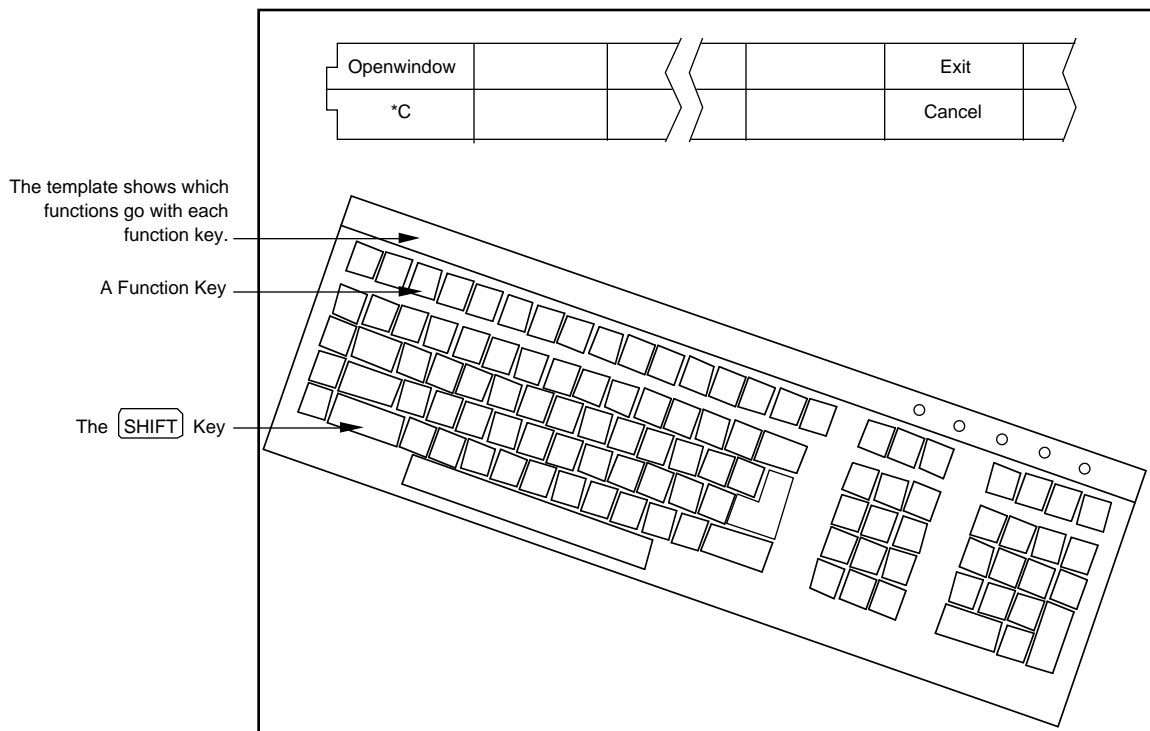
On the Tandem 6530 template, the EXIT key is the shifted `[F16]` key. In this manual the keys are referred to by name, however, not by number. Regarding shifted keys, therefore, if you are asked to press the EXIT key, you should look at the template and see that you must press both the terminal `[SHIFT]` key and the TEDIT EXIT function key in order to exit.

## Creating a Simple Document

### Using TEDIT Function Keys

In particular, notice the key labeled \*C (called star C). When you press this key, TEDIT displays a highlighted line at the top of the screen where you can type *commands* or ask for help. After you type the command, press \*C again to have TEDIT perform the command. Using \*C and typing a command works the same way as pressing a function key labeled with that command. The \*C key is discussed in more detail in Section 2.

Figure 1-2. Using Shifted and Unshifted Function Keys



---

**Highlight** Asking for Help

You can get help from TEDIT about any function that you see named on the function key template, such as \*C or CANCEL. You can also get help on functions and *topics* that are not shown on the template; for example, you can ask for help on COMMANDS or STATUSLINE. Follow these steps to get help from TEDIT:

1. Press the key labeled HELP. At the bottom of the screen, on the status line, TEDIT displays the message shown in the following example. The message means that TEDIT is getting the *online help* ready to display whatever help you request.

```
1) Wait. Initializing HELP                                C00
```

As soon as the help is ready (it usually takes just a few seconds), TEDIT displays the following highlighted line at the top of the screen:

```
Topic:█
```

2. Type the name of any function or topic on the highlighted line after the "Topic:" prompt.
3. Then press the HELP key again to get information on the function you specified.

If you do not want to type a topic now, simply press the CANCEL key. The online help is discussed in more detail in Section 3.

---

**Note** You can use the online help to learn how to use TEDIT. Simply type TEDIT at your command interpreter instead of typing TEDIT followed by a file name. When TEDIT starts up, it tells you what to do next.

---



---

**Exiting From the File** Any time you want to get out of the file you are editing, you can simply press the EXIT key. Do it now. If you do not have a key labeled EXIT, skip to the end of this topic and read the “Note.”

TEDIT responds with this question at the top of the screen:

```
Do you really want to exit? Respond Y or N: █
```

- If you don't want to exit, type N (meaning no) and press \*C. On the Tandem 6530 terminal, you can press `RETURN` instead of \*C. TEDIT erases the question at the top of the screen and puts your cursor back where it was before you gave the EXIT command.
- If you want to exit, type Y (meaning yes) and press \*C. Do it now. TEDIT exits from your file and returns you to the command interpreter.

---

**Note** If you do not have a key labeled EXIT, you can exit from TEDIT in another way. Follow these steps:

1. Press \*C. A highlighted line appears at the top of the screen with the word “Command:” on it.
2. Type EXIT on the highlighted line. See the following example.

```
Command: EXIT █
```

3. Press \*C again. From this point, the steps are the same as if you had pressed the EXIT key discussed above.
-

**Summary** In this section you created a TEDIT file. You learned about the status line, where TEDIT always displays the name of the file you are in. You should now know how to enter text into the file and use some of the function keys on your terminal to move around in the file and to edit it.

The editing functions you learned in this section were inserting characters with the `CHAR INS` key and `SHIFT CHAR INS` keys, deleting characters with the `CHAR DEL` key or the `SHIFT CHAR INS` and `BACKSPACE` keys, erasing characters with the `SPACE` bar, and replacing characters by typing over them. You also learned to insert and to delete lines. You used the terminal function keys to do these editing tasks.

You learned about TEDIT functions, using your template to identify shifted and unshifted function keys. You saw how to give commands by pressing a function key or by typing the command on a highlighted line at the top of the screen. You saw how to ask for help from TEDIT. Finally, you learned two ways to exit from a TEDIT file. In the next section, you will reopen the same file and edit it using TEDIT function keys.

---

## 2 Revising a Simple Document

---

**Overview** In Section 2 you revise the document you created in Section 1. The first thing you do is reopen the MEMO file where the document is stored. You learn to use TEDIT function keys to edit the memo and, at the same time, you practice using the terminal keys you already learned. The features covered in this section are:

- Inserting lines, characters, and sentences
- Moving text from one place to another
- Giving written commands
- Using written commands in conjunction with function keys
- Breaking lines
- Deleting words, lines, and sentences
- Moving forward by sentences
- Moving forward or backward to the end of a line
- Canceling commands
- Adjusting line widths
- Undoing mistakes you have made

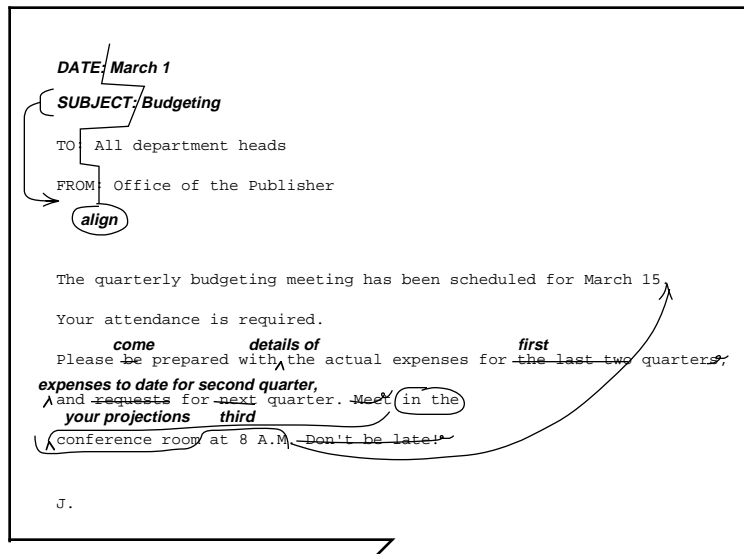
The way the material in Section 2 is presented assumes that you will work through the entire section in one editing *session*; that is, without exiting. It should take you about 30 minutes. If you do not intend to do that, look at “Undoing Mistakes You Have Made” in this section before you exit.

Look at the hand-edited memo in Figure 2-1. In this section you will use TEDIT to do the editing you see in the figure.

## Revising a Simple Document

### Overview

Figure 2-1. Editing a Simple Memo



**Reopening the File** Once you have created a file, it is easy to open it again. Make sure you are in the right volume and subvolume. Then, at the command interpreter prompt, type TEDIT followed by the name of the file and press `RETURN`. See the example.

```
2> TEDIT MEMO █
```

TEDIT gets the file and displays it for you. The cursor is in the top left corner of the screen at the beginning of the text. The file looks exactly the same as it did at the end of your last editing session. See the following example.

```
█O: All department heads
FROM: Office of the Publisher

The quarterly budgeting meeting has been scheduled for March 15.
Your attendance is required.
Please be prepared with the actual expenses for the last two quarters,
and requests for next quarter. Meet in the
conference room at 8 A.M. Don't be late!

J.
```

**Inserting Lines and Characters**

Start editing this memo by adding two more lines to the heading. You learned how to insert lines by using the `LINE INS` key in Section 1. In this section you will use TEDIT function keys to do the same thing. Follow these steps:

1. If it is not already there, put the cursor in the HOME position.
2. Look at the template and find the key labeled LINE. Press it. Notice that the word LINE appears on the status line at the bottom of the screen. Your screen should look like the following example.

TO: All department heads  
FROM: Office of the Publisher

The quarterly budgeting meeting has been scheduled for March 15. Your attendance is required. Please be prepared with the actual expenses for the last two quarters, and requests for next quarter. Meet in the conference room at 8 A.M. Don't be late!

J.

1) line

C00.

LINE appears on the status line.

3. Now find the key labeled INSERT on the template. Remember that to get the INSERT function, you must hold down the `SHIFT` key and then press INSERT. Do it now. TEDIT inserts a blank line where the cursor is positioned, and all the lines below it move down. Notice the change in the status line. The appearance of the word LINE on the status line has to do with the *persistence* feature, which is discussed later in this section.

- To insert another line, press INSERT again. You do not need to press LINE again. Your screen should now look like the following example.

Use LINE plus INSERT to insert two blank lines.

Notice that TEDIT numbers the inserted lines with decimal numbers.

```

■
TO: All department heads
FROM: Office of the Publisher

The quarterly budgeting meeting has been scheduled for March 15.
Your attendance is required.
Please be prepared with the actual expenses for the last two quarters,
and requests for next quarter. Meet in the
conference room at 8 A.M. Don't be late!

J.
1) $MYVOL.MYSVOL.MEMO .01/21 (BOF) (EOF) 1:79 Line C00.
```

- Type the date you see in the following example. Press RETURN or press TAB when you reach the end of the line. Then type the subject as shown in the following example.
- Now use the CHAR INS key, which you learned in Section 1, to move the heading lines so that all the colons are aligned with each other. You could use TEDIT function keys to insert characters as well, but it is easier to use the terminal key to do it. Your screen should look like the following example when you are finished.

Aligned colons using CHAR INS.

Typed date and subject lines.

```

DATE: March 1
SUBJECT: Budgeting
TO: All department heads
■ FROM: Office of the Publisher

The quarterly budgeting meeting has been scheduled for March 15.
Your attendance is required.
Please be prepared with the actual expenses for the last two quarters,
and requests for next quarter. Meet in the
conference room at 8 A.M. Don't be late!

J.
1) $MYVOL.MYSVOL.MEMO .01/21 (BOF) (EOF) 1:79 Line C00.
```

## Revising a Simple Document

### Moving a Line and a Sentence

#### Moving a Line and a Sentence

Some of the sentences in the memo seem out of order. Now that you have added two lines to the heading, it looks out of order, too. The next editing task you do is moving lines and sentences around.

#### Moving a Line

To move the subject line to the bottom of the heading, follow these steps:

1. Put the cursor anywhere on the subject line. Because LINE still appears, or persists, on the status line, you do not need to press LINE again to tell TEDIT that you want to do something else with this line.

The cursor is here.

LINE is on the status line.

```
DATE: March 1
SUBJECT: Budgeting
TO: All department heads
FROM: Office of the Publisher

The quarterly budgeting meeting has been scheduled for March 15.
Your attendance is required.
Please be prepared with the actual expenses for the last two quarters,
and requests for next quarter. Meet in the
conference room at 8 A.M. Don't be late!

J.

1) $MYVOL.MYSVOL.MEMO .01/21 (BOF) (EOF) 1:79 Line C00.
```

2. Find the key labeled MOVE. Press it. The subject line disappears. Notice that when TEDIT moves the line, it closes up the space the line was in. At this point, the heading should look like the following example.

TEDIT removes the line.

```
DATE: March 1
TO: All department heads
FROM: Office of the Publisher
```



3. Next you tell TEDIT where you want to move the line. Move the cursor to the blank line below "FROM."
4. Find the key labeled RETRIEVE. Press it. TEDIT retrieves the subject in the line in which the cursor is positioned.

TEDIT retrieves the subject in the line where the cursor is.

```
DATE: March 1
TO: All department heads
FROM: Office of the Publisher
SUBJECT: Budgeting

The quarterly budgeting meeting has been scheduled for March 15.
Your attendance is required.
Please be prepared with the actual expenses for the last two quarters,
and requests for next quarter. Meet in the
conference room at 8 A.M. Don't be late!

J.

1) $MYVOL.MYSVOL.MEMO .01/21 (BOF) (EOF) 1:79 Line C00.
```

---

**Highlight**   **How TEDIT Moves Text**

What happens to the text when TEDIT moves it from one place to another? When you use the MOVE or COPY command, TEDIT puts the text in a temporary holding area called the *scratchpad*. The text stays on the scratchpad until you use the RETRIEVE command. In fact, a copy of the text stays on the scratchpad until you retrieve it and then replace the contents of the scratchpad with some other text, using MOVE or COPY. The following written example describes, and Figure 2-2 illustrates, how the scratchpad works.

Here is an example: If you give a MOVE command on text A, TEDIT moves text A to the scratchpad. If the next command you give is a RETRIEVE command, a copy of text A is retrieved into the file and another copy stays on the scratchpad. You can retrieve text A as many times as you like. However, the next time you use the MOVE command, for example to move text B, this new text replaces (writes over) the text A that was previously on the scratchpad. The next time you give the RETRIEVE command, TEDIT retrieves only text B.

If you want to move several different items of text to the scratchpad, give sequential MOVE commands without retrieving any of the text in between the MOVE commands. Thus, if you give a MOVE command first on text A, next on text B, and then on text C, TEDIT saves all three text items on the scratchpad in the order you moved them. When you give a RETRIEVE command, TEDIT retrieves text A, text B, and text C all together into the file.

One advantage of the way the scratchpad is designed is that until you move something else, the text stays on the scratchpad so that you can retrieve it as many times as you want. Another advantage is that you can move several different items to the scratchpad and then retrieve them all together in one place.

If you press RETRIEVE again now, the subject line you moved is retrieved again. Try it. Then use the `LINE DEL` key to delete the extra line.

Figure 2-2. Using the Scratchpad to Move Text (Page 1 of 2)

	Step	Appearance of File	Appearance of Scratchpad
<b>Task 1A: Move line A.</b>	1. Position the cursor on line A.	█ This is A. This is B. This is C.	
	2. Press LINE. Press MOVE.	█ This is B. This is C.	This is A.
	3. Position the cursor where you want to place line A.	This is B. This is C. █	This is A.
	4. Press RETRIEVE.	This is B. This is C. █ This is A.	This is A.
<b>Task 1B: Move line B.</b>	5. Position the cursor on line B.	█ This is B. This is C. This is A.	This is A.
	6. Press LINE. Press MOVE.	█ This is C. This is A.	This is B.
	7. Position the cursor where you want to place line B.	█ This is C. This is A.	This is B.
	8. Press RETRIEVE.	This is C. █ This is B. This is A.	This is B.

## Revising a Simple Document

### Moving a Sentence

Figure 2-2. Using the Scratchpad to Move Text (Page 2 of 2)

	Step	Appearance of File	Appearance of Scratchpad
<b>Task 2: Move lines B and A.</b>	1. Position the cursor on line B.	This is A. █ This is B. This is C.	
	2. Press LINE. Press MOVE.	This is A. █ This is C.	This is B.
	3. Position the cursor on line A.	█ This is A. This is C.	This is B.
	4. Press MOVE.	█ This is C.	This is B. This is A.
	5. Position the cursor where you want to retrieve lines B and A.	This is C. █	This is B. This is A.
	6. Press RETRIEVE.	This is C. █ This is B. This is A.	This is B. This is A.

**Moving a Sentence** Now try moving a sentence. Refer to Figure 2-1 to recall the editing planned for this memo. Follow these steps to move the penultimate sentence:

1. Find the sentence that reads, "Meet in the conference room at 8 A.M." Put the cursor on the first letter of the sentence, the "M" in "Meet."

2. Press the key labeled SENTENCE. Notice the change to the status line.



Put the cursor here and press SENTENCE.


```

The quarterly budgeting meeting has been scheduled for March 15.
Your attendance is required.
Please be prepared with the actual expenses for the last two quarters,
and requests for next quarter. Meet in the
conference room at 8 A.M. Don't be late!
J.
1) sentence                                C00.
    
```


3. Press MOVE. TEDIT moves all the text from the position of the cursor to the end of the sentence. TEDIT looks for a period, an exclamation mark, or a question mark followed by a space so that it knows where the end of the sentence is. The sentence is now on the scratchpad. Notice that TEDIT moves the following sentence up into the space vacated by the sentence you moved.

Locating the New Position for the Sentence

You are going to tack this sentence onto the end of the first sentence. You could use the  key to move the cursor up four lines and then the  key to move to the end of the line. But there is a faster way to move the cursor to the end of the line. Follow these steps:

1. Use  to move the cursor up to the top line.
2. Find the key labeled EOL and press it. (EOL stands for end-of-line.)
3. Find the key labeled FORWARD. Press it. TEDIT moves the cursor forward to the end of the line. See the example.

1 Start with the cursor here.

2 Use  to move the cursor here.

3 Press EOL and FORWARD to move the cursor here.

```

The quarterly budgeting meeting has been scheduled for March 15.
Your attendance is required.
Please be prepared with the actual expenses for the last two quarters
and requests for next quarter. Don't be late!
J.
1) $MYVOL.MYSVOL.MEMO .01/21 (BOF) (EOF) 1:79 End-Of-Line      C0
    
```

## Revising a Simple Document

---

### Retrieving the Sentence

**Note** On the Tandem 6530 terminal you can move quickly to the end of the line by holding down the `CTRL` key and pressing `RETURN`. It works the same way as the TEDIT command, FORWARD EOL.

---

**Retrieving the Sentence** Now you are ready to retrieve the sentence you have on the scratchpad at the cursor location. First, backspace once and space over the period to erase it (you do not want a period here anymore).

Notice that there isn't room for a sentence at the end of this line. Press `RETURN` to move to the next line. Now press RETRIEVE. Look at the following example. The sentence is retrieved above the line the cursor was in.

TEDIT retrieves the sentence here.

```
The quarterly budgeting meeting has been scheduled for March 15
met in the
conference room at 8 A.M.
Your attendance is required.
Please be prepared with the actual expenses for the last two quarters,
and requests for next quarter. Don't be late!

J.

1) $MYVOL.MYSVOL.MEMO .01/21 (BOF) (EOF) 1:79 End-Of-Line C00.
```

**Moving Words** The next step to editing the memo is moving words. You probably do not have a key labeled WORD on your template. To move a word if the text item WORD is not on a function key, you use a written command. You already saw how to EXIT by giving a written command in Section 1. Here you learn how to give a written command in combination with pressing a function key. Here's how to do it:

1. Look on your template and find the key labeled \*C. Press it. The word "Command:" appears on a line at the top of the screen and the cursor is positioned just after it. The line is called a *response line*. See the following example.

2 The cursor is here.

1 Press \*C to get a response line.

```

Command: █
      TO: All department heads
      FROM: Office of the Publisher
      SUBJECT: Budgeting

The quarterly budgeting meeting has been scheduled for March 15
  
```

2. Notice that the cursor is on the line, ready for you to type something. Type WORD on the line. Look at the example.

Type WORD here.

```

Command: word █
      TO: All department heads
      FROM: Office of the Publisher
      SUBJECT: Budgeting

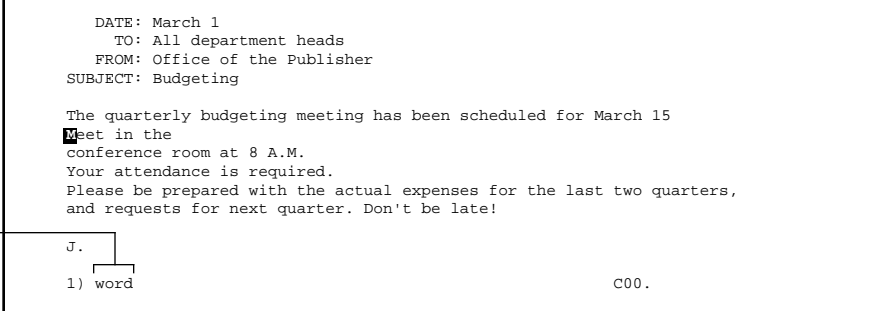
The quarterly budgeting meeting has been scheduled for March 15
  
```

## Revising a Simple Document

---

### Moving Words

3. Press \*C again. The response line disappears and WORD appears on the status line at the bottom of the screen exactly as if you had used a function key to give the command. See the following example.



DATE: March 1  
TO: All department heads  
FROM: Office of the Publisher  
SUBJECT: Budgeting

The quarterly budgeting meeting has been scheduled for March 15  
Meet in the  
conference room at 8 A.M.  
Your attendance is required.  
Please be prepared with the actual expenses for the last two quarters,  
and requests for next quarter. Don't be late!

J.  
1) word

c00.

WORD is now on the status line.

---

**Note** If you decide you do not want to type anything on the response line, simply press \*C again. The response line disappears and the screen looks exactly as it did before you pressed \*C the first time. TEDIT did not make any changes to the file.

---



---

**Highlight Giving Written Commands**

WORD is just one example of a TEDIT function that is not assigned to a function key. TEDIT has many more functions than there are function keys on your terminal. You can use these additional functions to extend your editing capabilities beyond the physical limit of the number of function keys on a terminal. You use these additional functions by giving TEDIT *written commands*.

So far when you wanted TEDIT to do something, you pressed a function key or two function keys in combination. The types of actions the keys discussed so far in this section perform are divided into two categories: text commands and text items. A *text command* tells TEDIT to do something, to take an action—such as move or delete—on a text item. A *text item* is the name of the object on which the action is being taken; it tells TEDIT what part of your text you want to manipulate, such as a word or a sentence.

Look on your template to see if you can distinguish the text commands from the text items. The text commands you have used so far in this section are INSERT, MOVE, and FORWARD. The text items you have used are LINE, SENTENCE, and EOL.

**Using Function Keys Along  
With a Written Command**

Now that you have WORD on the status line you can use the function keys to take actions on specific words. Start by rearranging the words in the second sentence so that it reads properly after it is connected to the first sentence. Look at the editing in Figure 2-1 to see what you are going to do.

## Revising a Simple Document

### Using Function Keys Along With a Written Command

Now, follow these steps:

1. Make sure that WORD is still on the status line. If it is not, reread the previous steps under the heading, "Moving Words." Put the cursor at the beginning of the word "Meet." Putting the cursor here tells TEDIT that you want to take action on this word. Look at the following example.

```
The quarterly budgeting meeting has been scheduled for March 15
M Meet in the
conference room at 8 A.M.
Your attendance is required.
Please be prepared with the actual expenses for the last two quarters,
and requests for next quarter. Don't be late!

J.

1) word                                     C00.
```

2. Now find the key labeled DELETE on the template. (DELETE is a text command.) Press it. TEDIT deletes the word "Meet." See the example.

```
The quarterly budgeting meeting has been scheduled for March 15
M the
conference room at 8 A.M.
Your attendance is required.
Please be prepared with the actual expenses for the last two quarters,
and requests for next quarter. Don't be late!

J.

1) $MYVOL.MYSVOL.MEMO .01/21 (BOF) (EOF) 1:79 Word      C00.
```

3. Now move the next four words to the end of the sentence. With the cursor starting on the word "in," press MOVE. Press MOVE once for each word you want to move. Each time you press MOVE, the cursor advances to the next word as it removes the previous one. All four words are now on the scratchpad.
4. Use the `→` key (or use `CTRL` and `RETURN`) to move to the end of the line. Press the `SPACE` bar once.

5. Press RETRIEVE. All four words are retrieved together at the cursor position.
6. Now press EOL and then press FORWARD to move the cursor to the end of the line and put a period on the sentence. By the way, you can use EOL and FORWARD (or `CTRL` and `RETURN`) or EOL and BACKWARD to move forward or backward along the ends of the lines. Your file should now look like this:

1 Retrieved the four words here.

2 Added a period here.

```

The quarterly budgeting meeting has been scheduled for March 15
at 8 A.M. in the conference room.
Your attendance is required.
Please be prepared with the actual expenses for the last two quarters,
and requests for next quarter. Don't be late!

J.

1) $MYVOL.MYSVOL.MEMO .01/21 (BOF) (EOF) 1:79 End-Of-Line C00.
    
```

**Breaking a Line** Now that you have finished editing the first sentence, go on to the third sentence. Refer to the hand-edited memo in Figure 2-1 to recall the changes needed in the sentence.

First, you want to insert the phrase “details of” before “the actual expenses,” but there is not enough room on the line. So, to make space to insert text, you break the line at the place where you want to insert; *breaking the line* moves the words on the last part of the line down to the next line. Follow these steps:

1. Using the arrow keys, move to the first “the” in the third sentence. With the cursor on the “t,” press SENTENCE. (SENTENCE is a text item.) Notice the change to the status line. Look at the next example.

Put the cursor here to break the line.

```
The quarterly budgeting meeting has been scheduled for March 15
at 8 A.M. in the conference room.
Your attendance is required.
Please be prepared with the actual expenses for the last two quarters,
and requests for next quarter. Don't be late!
J.
1) sentence                                C00.
```

2. Now press INSERT. The text command INSERT works with the text item SENTENCE. TEDIT breaks the line at the cursor and moves all the text that is to the right of the cursor down to the next line.
3. Now type “details of” at the end of the line where the cursor is still sitting. Your screen should look like the following example.

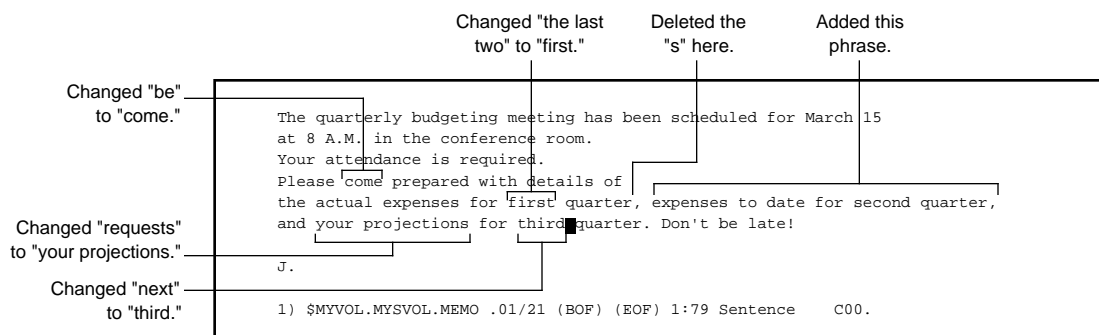
Type the new phrase here.

```
The quarterly budgeting meeting has been scheduled for March 15
at 8 A.M. in the conference room.
Your attendance is required.
Please be prepared with details of the actual expenses for the last two quarters,
and requests for next quarter. Don't be late!
J.
1) $MYVOL.MYSVOL.MEMO .01/21 (BOF) (EOF) 1:79 Sentence C00.
```

**Replacing Words and Inserting Text**

Follow these steps and look at the example to continue editing the third sentence in the memo.

1. Using the arrow keys, position the cursor on the word "be" in the fourth line.
2. Press `CHAR INS` twice. Type "come" in the space, typing over "be" and the two blank spaces you inserted.
3. Using the arrow keys, move to the last "the" in the next line. Type "first" and use the `CHAR DEL` key to delete the remaining characters between "first" and "quarters."
4. Using the arrow keys, move the cursor to the "s" in "quarters" and delete the "s" with the `CHAR DEL` key. Type "expenses to date for second quarter," at the end of the line.
5. Press `RETURN` to go to the next line and use the arrow keys to move the cursor to the word "requests." Use the `CHAR INS` key to make enough space to type "your projections" to replace "requests."
6. Using the arrow key, move over to the word "next"; change it to the word "third." Press `CHAR INS` once to make room for it. Your screen should look like the following example.



---

**Deleting a Sentence** Finish editing the body of the memo by deleting the last sentence. You have learned several ways to delete. However, as long as you have the item SENTENCE on the status line, it is quite efficient to simply use the DELETE command in conjunction with SENTENCE.

1. If you do not have the item SENTENCE on the status line, press the SENTENCE key.
2. Press the FORWARD key to move to the beginning of the sentence.
3. Press the key labeled DELETE on the template. TEDIT deletes the last sentence.

---

**Highlight Using and Canceling Persistence**

So far in this section you pressed the function key for a text item first, then you pressed the key for the text command. (Remember that a text command—such as DELETE or MOVE—tells TEDIT to take action on a text item; the item is the object—such as a WORD or SENTENCE—on which the action is taken.) The function of the text item appeared on the status line. Then you used the item that persisted on the status line with keys that give commands. Thus, throughout this section, you have been using the TEDIT feature called *persistence*.

This is how persistence works: Once you have used a text command or text item, it “persists,” or remains effective, until you replace it with another. The persistent command or item always appears on the status line. However, you can replace a persistent command only with another command; you can replace a persistent item only with another item.

For example, the last two keys you used were SENTENCE and DELETE. If you had started out by pressing DELETE and then SENTENCE instead of SENTENCE and then DELETE, the command DELETE would show up on the status line instead of the item SENTENCE. In addition, the command DELETE would persist instead of the item SENTENCE.

If you want to change from a persistent item to a persistent command, you must cancel the item that appears on the status line. You cancel the persistent item or command by pressing the function key labeled CANCEL; pressing CANCEL also clears the persistent word from the status line. Now if you want to give another command, simply press the function keys for that command. The key you press next (if it is a text item or text command) is the one that persists.

**Note** You can use CANCEL to cancel any command you give before TEDIT performs it; CANCEL is not used solely for the persistence feature.

### Adjusting the Line Width

Now that you have finished editing the memo, the lines are of quite varied widths. Your memo looks like the following example.

```

DATE: March 1
SUBJECT: Budgeting
      TO: All department heads
      FROM: Office of the Publisher

The quarterly budgeting meeting has been scheduled for March 15
at 8 A.M. in the conference room.
Your attendance is required.
Please come prepared with details of
the actual expenses for first quarter, expenses to date for second quarter,
and your projections for third quarter.

❑
1) $MYVOL.MYSVOL.MEMO .01/21 (BOF) (EOF) 1:79 Sentence C00.
```

The memo would look better if all the lines were of approximately the same width. There is a TEDIT command called ALIGN that makes a line or a sentence the width you want by breaking the line if it exceeds that width. You can also use the ALIGN command to align all the lines in a paragraph at the same time. Here is how to do it:

1. Put the cursor at the beginning of the body of the memo.
2. Press PARAGRAPH.

3. Press **ALIGN**. TEDIT aligns all the lines in the paragraph to a width of 70 characters or less. See the following example.

```
The quarterly budgeting meeting has been scheduled for March 15 at 8
A.M. in the conference room. Your attendance is required. Please
come prepared with details of the actual expenses for first quarter,
expenses to date for second quarter, and your projections for third
quarter.
```

```
▣.
```

```
1) $MYVOL.MYSVOL.MEMO .01/21 (BOF) (EOF) 1:79 Paragraph C00.
```

What if you would rather align the lines at a different width? You can change the line width if you want. In fact, you can change any of the functions the TEDIT function keys perform. You can also change many optional settings, such as tab stops and line width.

---

**Note** Your TEDIT contact assigns the functions the keys perform (such as **ALIGN**) and the optional settings associated with them (such as align width). The assignments are saved in your personal TEDIT profile. This manual mainly discusses these original assignments made by your TEDIT contact, which are called the default functions. The defaults are the functions the keys perform unless you change their assignments in your profile. Changing your profile is discussed in Section 3.

---





---

**Exiting** Now that you have finished editing this memo, you can exit from the file. (Press EXIT, answer Y, and press \*C.) You will use this same memo again in Section 4.

You may want to print the memo you created. If you have TFORM, you can use it to format and print the memo. TFORM is discussed in Part Two of this manual.

---

**Summary** In this section you learned how to use TEDIT function keys to perform such text commands as INSERT, DELETE, and MOVE on text items, such as LINE, SENTENCE, and WORD. You used TEDIT functions, such as EOL with FORWARD or the terminal keys `CTRL` and `RETURN`, to move the cursor to the end of a line quickly.

You learned how to break a line with SENTENCE INSERT so you could insert a large portion of text in the middle of the line. And you learned how to adjust the line widths in a paragraph with PARAGRAPH ALIGN to even up the lines in the file after editing.

You used the persistence feature to do editing faster with fewer keystrokes, and you learned how to CANCEL persistence. Through the process of moving items of text you learned the difference between a text command and a text item. You saw how items are stored in a temporary location called the scratchpad when you move them. Finally, you learned how to UNDO and REDO changes you make to a file before you EXIT from the file.

---

## 3 Customizing TEDIT for a Columned List

---

**Overview** In this section you create a list in a three-column format. While doing that, you learn and use the following TEDIT features:

- Displaying the ruler to determine where to set tab stops
- Setting tab stops in the profile
- Tabbing to the stops you have set
- Getting an online list of all the TEDIT features
- Moving and copying text
- Making words uppercase or lowercase without retyping
- Creating and saving your own profile

---

**Creating a New File** To begin, create a new file for the list you are going to make. You create this file (naming it AGENDA) in the same way you created the MEMO file in Section 1. Briefly, here is what you do:

1. At the command interpreter prompt, type the command you see in the following example and press `RETURN`:

```
3> TEDIT AGENDA █
```

2. TEDIT shows you the name of the file you asked for and asks if it is OK to create it. Type Y and press `RETURN`. TEDIT creates the file and displays a blank screen for you to start typing.

## Customizing TEDIT for a Columned List

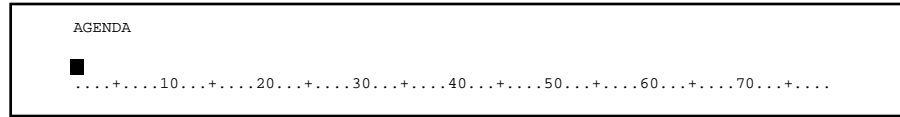
### Typing in Columns

#### Typing in Columns

Start with the cursor at the top of the screen. Type the word "AGENDA" and press **RETURN** twice. Now you are ready to type the body of the list, which is set up in three columns. TEDIT has two features that make typing in columns easy: 1) a *ruler* to determine where you want to position the columns and 2) a way to set tab stops to make your typing quicker.

#### Using the Ruler to Determine Column Widths

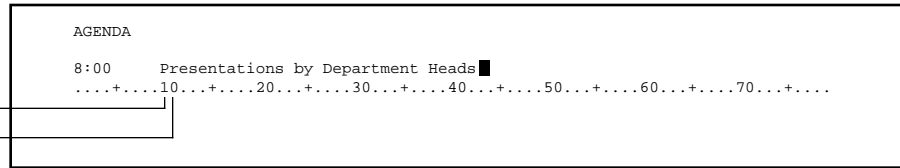
Before you start typing the next line, look on the template and locate the key labeled **RULER**. Press it. Your screen should look like the following example. A bright line appears on the line below the cursor with numbers and dots that mark every column across the screen. This is the ruler. Notice that the cursor is still on the line on which you want to type.



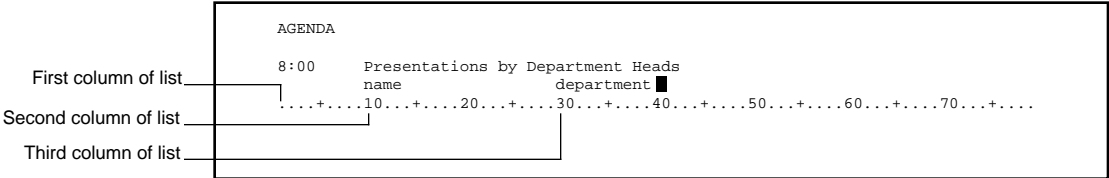
Use the ruler to determine where you want to start the next column. In this example, you put the first column at the far left (at column 1) and the second at column 10. To start, type "8:00" beginning in column 1. Then move the cursor to the right until it is directly above the 10 on the ruler. Type the heading shown in the next example. Do not press **RETURN**. When you finish, the screen should look like this:

The 1 is in column 10  
on the screen.

The 0 is in column 11  
on the screen.



Press the RULER key again to make the ruler disappear. (Actually, you could press any function key to make the ruler disappear.) Press RETURN to move to the next line. Press RULER again to determine where to place the two column headings shown in the next example. Type the headings at 10 and 30 as shown.



**Setting the Tab Stops** Before you type any more lines, you can set the tab stops in order to tab to the columns as you type them. Find the key labeled TAB on the template and press it a few times. You see that the tab stops are set at every eight characters. To reset the tab stops, follow these steps:

1. Look on the template and locate the key labeled RECONFIGURE OPTIONS. Press it. TEDIT displays the information shown in Figure 3-1. These are the options TEDIT uses when you edit; you can customize these options by creating and saving your own profile.

**Highlight The Options Part of the Profile**

A profile consists of all the settings for options and function keys that appear on the RECONFIGURE OPTIONS screen and the RECONFIGURE FUNCTION-KEYS screen. The original settings, called default settings, are shown in the default profile illustrated in Figure 3-1. Your TEDIT contact assigns the settings you start out with; yours could be different than the defaults shown in the figure. You can change the default settings by changing the assignments on the RECONFIGURE OPTIONS screen, discussed here. The RECONFIGURE FUNCTION-KEYS screen is discussed in Section 4.

## Customizing TEDIT for a Columned List

### The Options Part of the Profile

After you change the assignments, you save the new settings in another profile; you can create and save as many profiles as you want. The default profile is always available if you want to use it instead of one of your own. Changing the profile is discussed here under “Setting the Tab Stops.” Saving and reusing the profile is discussed at the end of this section under “Saving and Using the Profile.”

Figure 3-1. The Options Part of the Profile (Page 1)

```
Profile * from $SYSTEM.SYSTEM.TEDPROFL, created 18JUN87.

      Remarks: Standard Defaults
Align Width: 70           Auto Word Wrap: ON
Overlap (0-12): 2         Exit Confirmation: ON
Paragraph Lines: 6       Auto Renumber: ON
Printer Page Size: 0     RETURN Function: ON
Blank Fill on 3270: ON
Tab Stops (2-238): 9 17 25 33 41 49 57 65 73 81
Word Stops: " !"#$$%&()*+,-./:;<=>?@\_{}|~"
Sentence Stops: " !.?"
Format Stops: " !#%&'()*+,-./:;<=>?@\_{}|~"
SEARCH Options:
REPLACE Options:
OPENWINDOW Options:

Page 1 of 2
```

2. Use the arrow keys or the terminal keys for **TAB** and **BACKTAB** to move around on the RECONFIGURE OPTIONS screen. If you use the **RETURN** key, it might cause you to exit from the screen before you are ready. When you press **TAB**, TEDIT moves the cursor to the next place where you can type something. Each place where you can enter information is called a *field*. On the Tandem 6530 terminal, these fields are highlighted. Move the cursor to the field marked “Tab Stops.”

3. To change the tab stops—which are set at 9, 17, 25, and so on—you simply type the new numbers on top of the old ones. The numbers you type here correspond to the numbers you saw on the ruler; there is a number for each column visible on the screen. Type “10 30” on the tab stops line. Look at the following example.

```
Tab Stops (2-239): 10 3025 33 41 49 57 65 73 8
```

4. There are two things wrong with this line, which you should clean up before you leave the profile. It appears that 3025 is set as a tab stop, and TEDIT only accepts tabs in columns 2 through 239. Also, you must put the stops in order, from smallest to largest. So, you should do one of the following to make the tab stops work right: either erase the rest of the numbers on the line by spacing over them or delete the number 25 by pressing the `CHAR DEL` key. If you do the former, the tab stops line now looks like the next example. (If you have a Tandem 6530 terminal, you can erase the rest of the line by holding down the `CTRL` key and pressing the terminal `ERASE LINE` key.)

```
Tab Stops (2-239): 10 30
```

5. Now you can exit from the RECONFIGURE OPTIONS screen. As soon as you exit, you can use the tabs you have set. You exit by pressing `*C` or the `RETURN` key. Do it now. TEDIT returns you to the same place you were before you pressed RECONFIGURE OPTIONS the first time.

---

**Note** If you want to use these new settings again after this editing session, you must save the profile before you exit from TEDIT. See “Saving and Using the Profile,” discussed later in this section.

---

## Customizing TEDIT for a Columned List

### Tabbing to the Stops and Typing the Agenda

#### Tabbing to the Stops and Typing the Agenda

Now you are ready to type the rest of the agenda. If you have the RETURN function, press `RETURN` to put the cursor under the first column heading. If not, press the terminal `TAB` key followed by the TEDIT TAB key. Note that the terminal `TAB` key does not work with the tab stops set with TEDIT; the terminal `TAB` key merely moves you to the first column of the next line. Pressing the TEDIT TAB key moves the cursor to the tab stops.

Type the information you see in the following example.

```
AGENDA

8:00   Presentations by Department Heads
      name                department
      Monreau             publisher
      Riveira             finance
      Yuen                administration
      Medici              design and layout
      Strauss             production
      McAdams             editing
      Kitahara            printing and distribution
      Goncherov           marketing

12:00  Lunch

1:30   Ideas for next quarter

2:45   Break

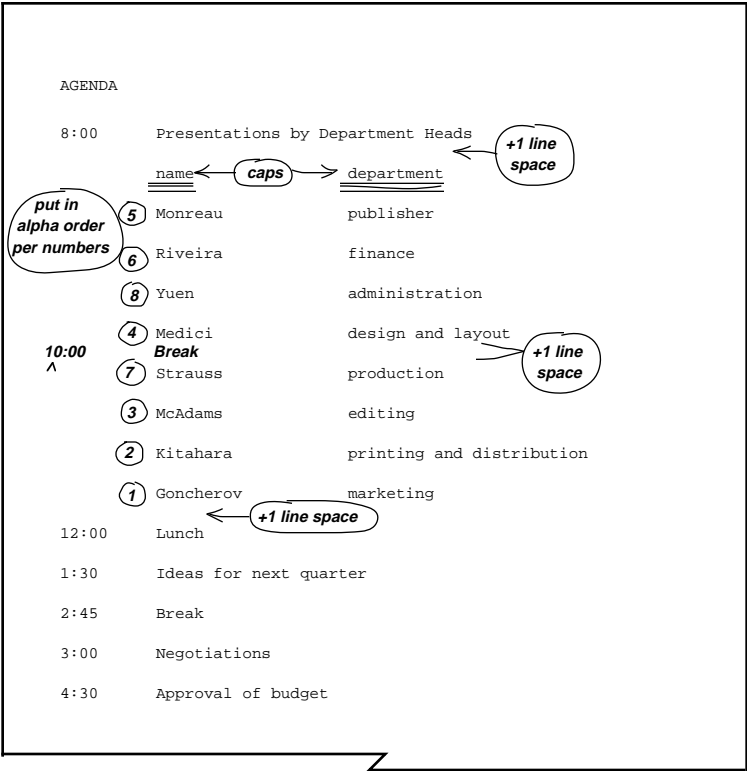
3:00   Negotiations

4:30   Approval of budget █
```



Look at the agenda in Figure 3-2. In the rest of this section, you will edit the agenda, using TEDIT to do the editing you see done by hand in the figure.

Figure 3-2. Editing the Agenda



---

**Moving and Copying Text**

The functions of moving and copying text are very much alike. You use MOVE to move the text from one place to another, removing the text from its original position. You use COPY to move text as well, but you still leave a copy of it in the original position.

You learned how to move words, lines, and sentences one at a time in the last section. In this section you use one of the features of the scratchpad to move seven different lines and retrieve them all together in alphabetical order. You learn what to do to empty the scratchpad if you make a mistake.

**Moving Names Into Alphabetical Order**

Start your editing of the agenda by rearranging the names into alphabetical order. You can use MOVE plus LINE to do the rearranging quickly and easily. Follow these steps:

1. Put the cursor anywhere on the line that contains the name you want first on the list. As shown in Figure 3-2, the name that goes first is Goncherov.
2. Press LINE, then press MOVE. TEDIT moves the line from the screen to the scratchpad.
3. Put the cursor on the line that contains the name you want next on the list, Kitahara. Press MOVE again. Referring to Figure 3-2 for the correct order, place the cursor on each line that is next in the order and press MOVE until you have moved all the names except Yuen to the scratchpad. Because Yuen is the last name in the alphabetical order, you do not need to move that name.

- Put the cursor in column 1 of the line that contains Yuen and press RETRIEVE. TEDIT retrieves all the names in the order you put them on the scratchpad. The agenda should now look like the following example:

```

AGENDA

8:00   Presentations by Department Heads
      name      department
      █ Gonchero  marketing
        Kitahara printing and distribution
        McAdams   editing
        Medici    design and layout
        Monreau   publisher
        Riveira   finance
        Strauss   production
        Yuen      administration

12:00  Lunch
1:30   Ideas for next quarter
2:45   Break
3:00   Negotiations
4:30   Approval of budget

```

### Emptying the Scratchpad

If you made a mistake in the process of putting the names on the scratchpad in the right order, you can start over. Use UNDO to put all the names back onto the screen. Because you have not used RETRIEVE yet, copies of the names you put on the scratchpad are still there. UNDO does not affect the information that is on the scratchpad. So, you must empty the scratchpad before you begin to move the names there again; otherwise, you will retrieve both sets of names when you use the RETRIEVE key. To empty the scratchpad, follow these steps:

- Press \*C.
- When you see the response line, type EMPTYSRATCHPAD (or the abbreviation, EM).

**Note** You can abbreviate all the TEDIT commands and items when you type them on the response line. Use any abbreviation you like as long as it uniquely identifies the function. You can find the shortest unique abbreviations in several places: on the reference card, in the reference manual, and in the online help.

## Customizing TEDIT for a Columned List

### Copying

3. Press \*C again. The response line goes away and everything looks the same as it did before. You do not see anything happen, but the scratchpad has been completely erased. You can use the EMPTYSCRATCHPAD command before you use MOVE or COPY if you do not remember whether there is anything on the scratchpad.

**Copying** Assume that you decide one break during a day of meeting is not enough. You want to put a break in the morning as well as in the afternoon. You can simply copy the line in the agenda that has the break scheduled and insert it where you want it in the morning schedule, leaving the original line intact. Copying the line saves you the work of typing a whole new line. Follow these steps:

1. Put the cursor anywhere on the line you want to copy; in this example, the line that contains "2:45 Break."
2. Press LINE, then press COPY. Notice that the line remains on the screen; it is also copied to the scratchpad.
3. Move the cursor to column 1 of the line just below where you want to insert the text you copied. See the following example.

The screenshot shows a terminal window with the following text:

```
AGENDA
8:00 Presentations by Department Heads
      name           department
      Goncherov      marketing
      Kitahara        printing and distribution
      McAdams         editing
      Medici          design and layout
      Monreau         publisher
      Riveira         finance
      Strauss         production
      Yuen            administration
12:00 Lunch
1:30 Ideas for next quarter
2:45 Break
3:00 Negotiations
4:30 Approval of budget
```

Annotations on the left side of the screenshot:

- Annotation 1: "1 Put the cursor here and press LINE. Then press COPY." with a line pointing to the "2:45 Break" line.
- Annotation 2: "2 Move the cursor to here. Then press RETRIEVE." with a line pointing to the "1:30 Ideas for next quarter" line.

- Press RETRIEVE. The line is retrieved above the line in which the cursor is positioned. The cursor moves up to the new line. Look at the following example. Change the time for the morning break as shown.

The screenshot shows a TEDIT window titled "AGENDA" containing the following text:

```

8:00 Presentations by Department Heads
      name department
      Goncherov marketing
      Kitahara printing and distribution
      McAdams editing
      Medici design and layout
10:00 Break
      Monreau publisher
      Riveira finance
      Strauss production
      Yuen administration
12:00 Lunch
1:30 Ideas for next quarter
2:45 Break
3:00 Negotiations
4:30 Approval of budget
    
```

Annotations on the left side of the screenshot:

- 1 TEDIT retrieves a copy of the line. (Points to the 10:00 Break line)
- 2 Change the time from 2:45 to 10:00. (Points to the 10:00 Break line)

### Highlight Knowing All the Available Functions

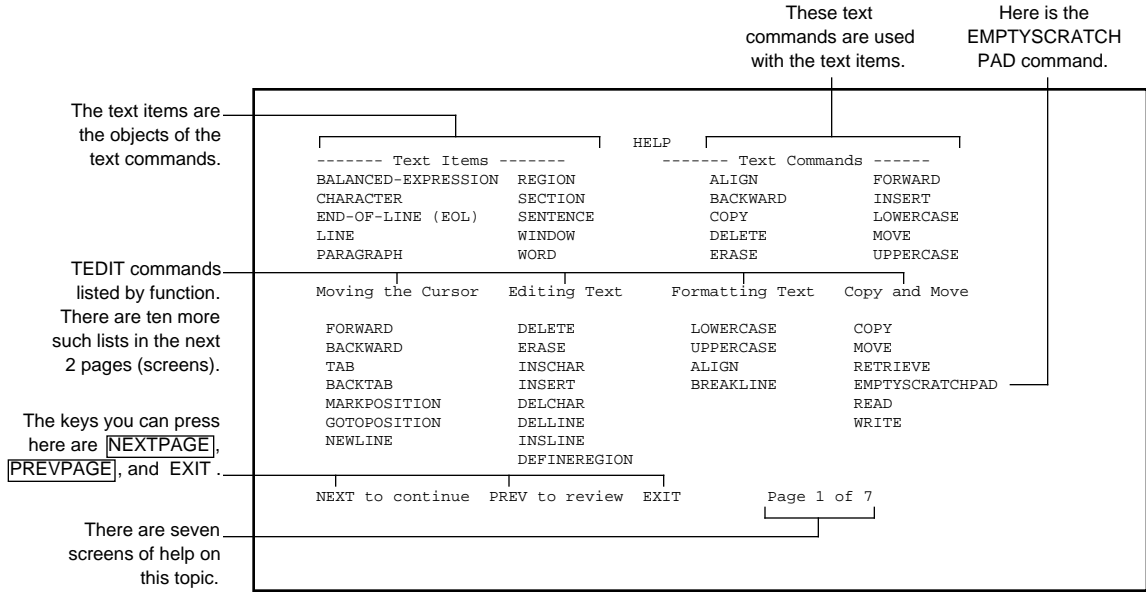
How do you find out about TEDIT functions, such as EMPTYSCRATCHPAD and WORD (used in Section 2), if they are not shown on the function key template? There are three sources for this information: the reference manual, the reference card, and the online help.

You can use the online help to get information on every TEDIT feature. The best place to start is to get a list of all the features of TEDIT. Press the HELP key. Remember that TEDIT takes a few seconds to get the help ready and then prompts you with the word "Topic:" and a response line. Press HELP twice more to get a list of all the available topics.

## Customizing TEDIT for a Columned List

Knowing All the Available Functions

This is what you see:



To see the next page of information about online help, press the **NEXT PAGE** key. Try it; notice that the status line now reads "Page 2 of 7." To go back to a previous page, press the **PREV PAGE** key. If you read through all seven pages of this **HELP**, TEDIT tells you what you need to know to get online help on any TEDIT feature.

**Note** If you want to go back to the Topic response line, press **EXIT** once. When the response line appears, you can type the name of any topic, command, text command, or text item. Type the topic names exactly as they are listed on the help screen. Then press the **HELP** key again to get information on the topic you specified. (The help includes the shortest abbreviation for each command.)

If you don't want to see any more information, press EXIT twice. EXIT returns you to your file. Nothing has changed; the file looks the same as it did before you pressed the HELP key. Do it now. If you did not type a topic name on the response line, the following message appears on the status line.

```
1) You didn't supply required information for the command C00.
```

Don't worry about the message. The next time you press a function key, the message goes away.

## Customizing TEDIT for a Columned List

### Making Words Uppercase Without Retyping

#### Making Words Uppercase Without Retyping

Now that you have the content of the agenda the way you want it, it is time to think about its appearance. The agenda would look better if you capitalized the column headings and added space above them.

Move the cursor to the line with the column headings. If you do not still have the text item LINE persisting on the status line, press the LINE key. To make this entire line uppercase, simply press the key that is labeled UPPERCASE on the template. Your column headings are now in uppercase letters as shown in the following example. You can make entire words, lines, paragraphs, or other text items uppercase or lowercase simply by using the text item in conjunction with the UPPERCASE or LOWERCASE command.

Move the cursor back up to the line with the column headings. With LINE persisting, press INSERT to add a blank line above the column headings. Add some more blank lines as shown in the following example, using the persisting text item LINE and the text command INSERT.

Made this line uppercase.

AGENDA	
8:00	Presentations by Department Heads
	NAME                      DEPARTMENT
	Goncherov                marketing
	Kitahara                 printing and distribution
	McAdams                 editing
	Medici                    design and layout
10:00	Break
	Monreau                 publisher
	Riveira                  finance
	Strauss                  production
	Yuen                      administration
12:00	Lunch
1:30	Ideas for next quarter
2:45	Break
3:00	Negotiations
4:30	Approval of budget

Added a line of space in each of these places.



**Saving and Using the Profile**

Now that you have finished editing the agenda, the next step would usually be to exit from the file. But, you made some changes to the tab stops in the profile that you might want to save. If you save this profile, you can use it again when you edit the agenda or if you create another agenda in the same format. If you do not save the settings, they are permanently erased when you exit from the file. You can save these settings by using the SAVEPROFILE command.

**Identifying the Profile**

Before you save the profile, you should identify it by writing comments in the Remarks field on the RECONFIGURE OPTIONS screen. Follow these steps:

1. Press the RECONFIGURE OPTIONS key.
2. The cursor is on the line labeled "Remarks." Starting at the cursor position, type something on the Remarks line that you can use to identify the profile later. Just type over the remarks that are there now. Here is an example of what to do for the agenda profile:

Write remarks about what is in the profile here.

```

Profile * from $SYSTEM.SYSTEM.TEDPROFL, created 18JUN87.
Remarks: Tabs at 10 & 30
Align Width: 70           Auto Word Wrap : ON
Overlap (0-12): 2       Exit Confirmation: ON
Paragraph Lines: 6      Auto Renumber: ON
Printer Page Size: 0     RETURN Function: ON
Blank Fill on 3270: ON
Tab Stops (2-239): 10 30
Word Stops: " !"#%&()*+.,/;<=>?@[\\_{}~"

```

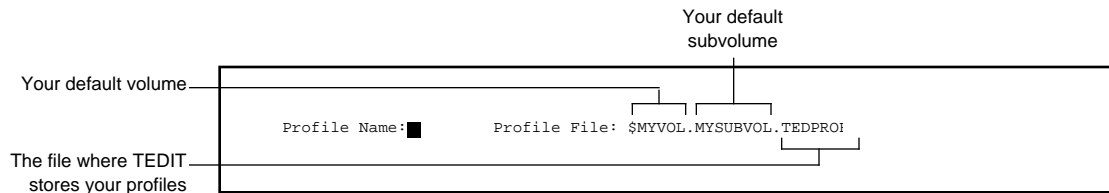
3. Press RECONFIGURE OPTIONS again to exit from the RECONFIGURE OPTIONS screen.

## Customizing TEDIT for a Columned List

### Saving the Profile

**Saving the Profile** Now you are ready to save the profile. You save it in a special file reserved for profiles in your default volume and subvolume. Follow these steps:

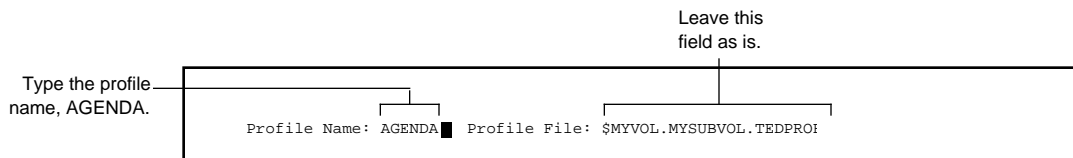
1. Press \*C.
2. On the response line type SAVEPROFILE (or simply SA) and press \*C again. You see the following display at the top of the screen.



TEDIT displays the name of the file where all the profiles you create are automatically saved. The *profile file* is called *TEDPROFL*, and it is on your default volume and subvolume.

TEDIT might also display a message on the status line telling you that you do not yet have any *local profiles*. Local profiles are ones that are stored on your default volume and subvolume. You do not need to respond to this message in any way.

3. Type the name of the profile you want to save; call this profile AGENDA. You can save the profile file in another location if you wish, but it is much better to have your profiles in your default volume.



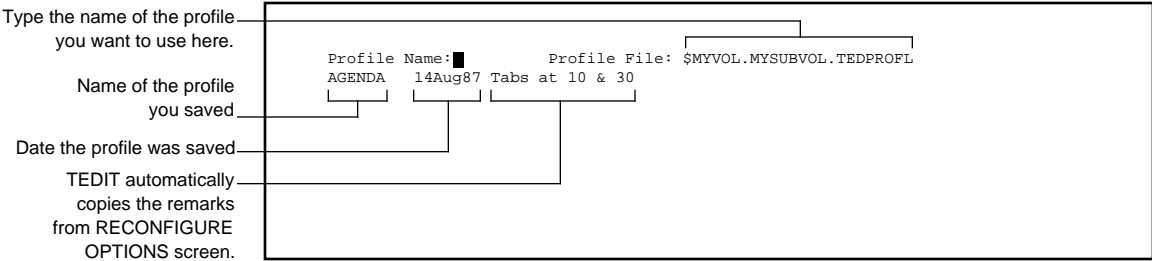
4. Press \*C to save the profile.

**Note** You can name the profile anything you wish as long as the name contains eight characters or fewer, the characters are either letters or numbers, and the first character is a letter. The exception is that if you save the default profile, you must name it \* (star).

To check that the profile was saved, give the SAVEPROFILE command again or give the USEPROFILE command discussed next. Both commands show you a list of the profiles you have saved. After you have seen the list, you can cancel the SAVEPROFILE or USEPROFILE command, or you can reissue it—TEDIT does not save the profile twice under the same name.

**Using the Profile** When you start a TEDIT session, you automatically use the default profile. The name of the default profile is \* (star). If you want to use another profile you have saved, such as AGENDA, give the USEPROFILE command. You can give this command at any time during an editing session when you want to use another profile. Follow these steps to instruct TEDIT to use the AGENDA profile:

1. Press \*C.
2. On the response line, type USEPROFILE (or simply US). Press \*C again.
3. When the second set of response lines appears, type the profile name, AGENDA. See the following example. Notice that the response lines look just the same as when you give the SAVEPROFILE command except the profiles you have saved, in this case just the one profile called AGENDA, appear in a list below the response lines.



---

**Highlight Using the Default Profile**

To use the default profile, give the USEPROFILE command and fill in the response lines as shown in the following example.

Type an asterisk (\*) for the profile name.

Type this file name here.

Profile Name: \*      Profile File: \$SYSTEM.SYSTEM.TEDPROFL

To avoid having to type the profile file name every time you want to use the default profile, save the default profile on your own default volume and subvolume. You do this by giving the SAVEPROFILE command and typing \* (star) in the Profile Name field. The next time you start a TEDIT session, use the above procedure to save the default profile before you make any changes to the RECONFIGURE OPTIONS screen.

**Deleting the Profile** When you no longer want to save the profile, use the following procedure to delete the profile. Follow these steps:

1. Press \*C.
2. On the response line, type PURGEPROFILE (or simply PU), then press \*C.
3. When the second set of response lines appears, type the name of the profile you want to delete.

---

**Note** There is a shortcut you can use to give the profile commands: You give the entire command on the \*C response line. This shortcut is especially useful if you change profiles often when you are editing. There is another shortcut—the SETPROFILE command—that allows you to change the settings in the profile quickly, but not necessarily save them for later use. Both shortcuts are discussed in Section 5.

---

---

**Summary** In this section you created a three-column agenda. You learned about the ruler for determining where to set tab stops. You set the tab stops on the RECONFIGURE OPTIONS screen of your profile. Then you used the TEDIT TAB key to move the cursor to the tab stops you set.

You rearranged text with the MOVE command, putting several text items on the scratchpad and then retrieving them in a new order all at the same place. You learned to copy text with the COPY command. You also learned how to empty the scratchpad with EMPTYSCRATCHPAD if you made a mistake when moving or copying text.

You used the UPPERCASE command with the text item LINE to put an entire line into uppercase letters.

You saw how to use online help to get a list of TEDIT features. The list includes the features that are assigned to the function keys and those that are not. You use these unassigned features by giving them as written commands on the \*C response line.

Finally, you learned how to save the profile you created in the first part of the section; how to use the profile again; and how to delete it when you are through with it. You also saw how to use the default profile again.

---

# 4 Advanced Editing Features

---

**Overview** In this section you create a document using the memo you wrote in Section 1 as the rough draft. You use this document (a form letter) to learn some new, more advanced TEDIT features. You also get more practice using features you learned in the previous sections. Features discussed in Section 4 include:

- Learning to use windows to have two files open simultaneously and to move text from one file to another
- Creating and saving a special profile
- Inserting paragraphs and sentences within existing text
- Breaking lines to reformat a paragraph into a list
- Changing names and personalizing information in the form letter with SEARCH and REPLACE

---

## Transferring Text to the New File

The document you create in this section is a rewrite of the memo you created in Section 2. To save retyping the rough-draft text for the document, you can bring the text from the MEMO file into the new file. You do this by opening a window into the MEMO file and using the COPY command.

A *window* is a view of a file. When you have one file open, the window through which you look at the file takes up the full screen. If you have two files open, or two windows into the same file, the windows each take up only a portion of the screen. Once you have two windows open, you can copy and move text from one window into the other. In the following pages, you open a new file called LETTER and your previously created MEMO file simultaneously. Then you copy text from MEMO to LETTER.

## Advanced Editing Features

### Creating the File

**Creating the File** First, create a new file named LETTER. At the command interpreter prompt, type the following and press `[RETURN]`:

```
4> TEDIT LETTER █
```

Answer Y when TEDIT asks you if it is OK to create the file and press `[RETURN]`. TEDIT creates the new LETTER file and presents you with a blank screen.

**Opening the Second Window** Check the status line to see that you are in the LETTER file. To open a window into the MEMO file, follow these steps:

1. Look on the function key template and find the key labeled OPENWINDOW. Press it. TEDIT displays some response lines that look like the following example.

Type MEMO.

Type 2 and press the terminal `[TAB]` key.

Leave blank.

```
OpenWindow: 2
File: MEMO █
Options:
Size:
```

**Note** Use the arrow keys or the terminal `[TAB]` key to move between the fields on the OPENWINDOW response lines and on all the TEDIT response line fields. If you press `[RETURN]`, it might cause you to exit from the response lines before you are ready.

2. Fill in the response lines (or fields) as explained below and as shown in the example.

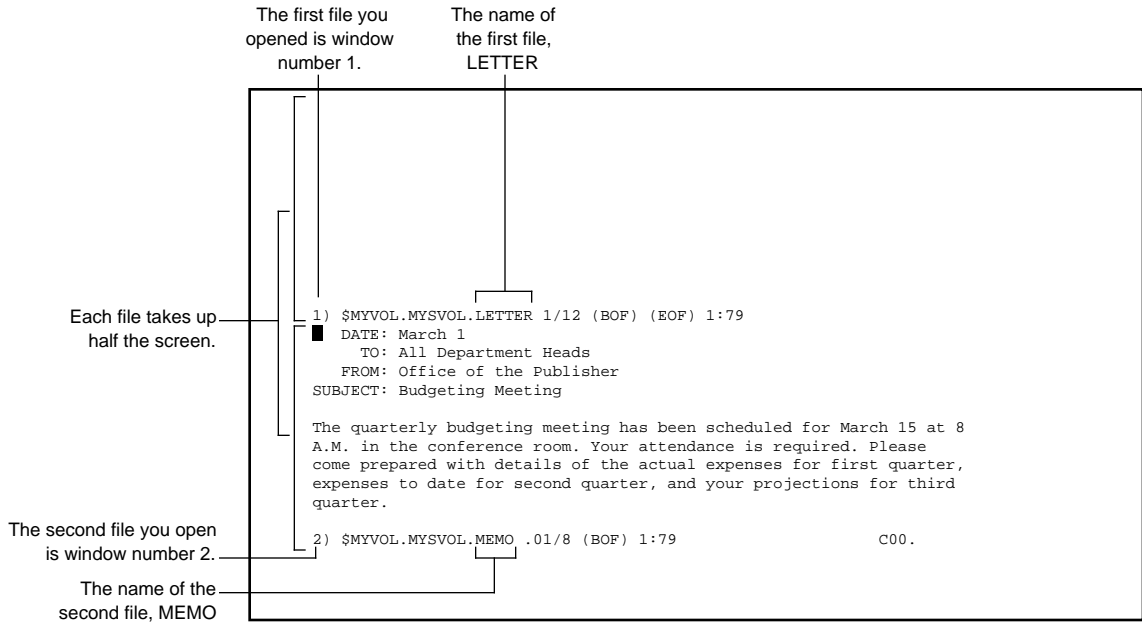
*OpenWindow:* Type either the number 1 or 2. If you type 2, TEDIT opens a second window. If you type 1, TEDIT closes the file you are in and opens the new file in window 1 (the window you currently have open). Instead of typing a 1 or a 2, you could type \* to mean close the current file and open another file in the window you currently have open or \*\* to open a second window.

**File:** Next, type the name of the file you want to open; in this case, you want to open the file named MEMO. If you want to open a second window into the same file, simply type the same file name again here.

**Options:** Some of the options you can specify here are related to *file security*; because you created MEMO, you don't need to worry about this. You can leave the field blank. For more information about the options, refer to the online help or the *PS TEXT EDIT Reference Manual*.

**Size:** You can specify what height you want the window, from 0 (zero) lines to the number of lines that fill the screen. (On a Tandem 6530 terminal a screen is 24 lines.) If you leave this field blank, the window will be half the size of the screen.

- After you fill in the information shown in the previous example, press \*C or **RETURN** to tell TEDIT to open the window. The screen should look like this:





## Advanced Editing Features

---

### Copying a Paragraph From One Window to Another

Now you have two files open simultaneously. You can have both files open at the same time; you can edit in a file when the cursor is in that window. Right now, the cursor should be in window 2. To move back to window 1, press the  key and hold it down until you have moved the cursor into the window. You can move the cursor back and forth between the two windows as often as you want, and you can copy and move text from one file to another as easily as if you were moving text around in only one file.

### Copying a Paragraph From One Window to Another

To get the rough-draft text into the LETTER file, copy the text from MEMO into LETTER. Follow these steps:

1. Using the arrow keys, position the cursor in the MEMO file at the beginning of the body of the memo on "T," the first letter of the first line.
2. Press the PARAGRAPH key. Then press the COPY key. You have just made a copy of the paragraph and put it on the scratchpad.
3. Press  to move the cursor to the first line of the LETTER file in window 1. You can also use the  arrow key to move the cursor if you want.

4. Press RETRIEVE. TEDIT retrieves the paragraph from the MEMO file into window 1, the LETTER file. The process of copying text between files is just as easy as copying within the same file. The screen should now look like this:

2 Move the cursor here. Press RETRIEVE to retrieve the paragraph.

1 Put the cursor here. Press PARAGRAPH, then press COPY

```

The quarterly budgeting meeting has been scheduled for March 15 at 8
A.M. in the conference room. Your attendance is required. Please
come prepared with details of the actual expenses for first quarter,
expenses to date for second quarter, and your projections for third
quarter.

1) $MYVOL.MYSVOL.LETTER 1/11 (BOF) (EOF) 1:79
DATE: March 1
TO: All Department Heads
FROM: Office of the Publisher
SUBJECT: Budgeting Meeting

The quarterly budgeting meeting has been scheduled for March 15 at 8
A.M. in the conference room. Your attendance is required. Please
come prepared with details of the actual expenses for first quarter,
expenses to date for second quarter, and your projections for third
quarter.

2) $MYVOL.MYSVOL.MEMO .01/8 (BOF) 1:79                                C00.
    
```

**Closing the Second Window**

Because you do not need to work with the MEMO file anymore, you can close the second window you opened. Follow these steps:

1. Press \*C. (It doesn't matter which file the cursor is in.)
2. When you get the response line, type CLOSEWINDOW 2 (or CL 2).
3. Press \*C again. TEDIT asks you if you are sure you want to close the window. Type Y for yes and press \*C. TEDIT exits from the MEMO file and removes the second window from the screen. Now you are looking at only the single file you opened first, LETTER. The window into the LETTER file now takes up the whole screen.

Now you are ready to start editing the memo to make it into a form letter.

---

**Setting Up the Profile** The first thing to do is to set up the options in the profile to determine how the letter will look. You can also set up some features that make it easier or faster for you to create the letter. The features you change in the profile for the letter are:

- Line width
- Tab settings
- A function key setting that allows TEDIT to adjust the line width automatically as you type

The first two features are set in the options part of the profile. (You changed the default profile already in Section 3, using RECONFIGURE OPTIONS, and saved the new settings in a profile named AGENDA.) The third feature in the previous list is set by changing the assignment made to a function key, in the function-keys part of the profile. After changing the settings in the default profile for this letter, you save them in a new profile named LETTER so you can use them again.

**Setting Line Width and Tabs**

To format the letter so that it looks nice and is easy to read, set the line width to be shorter than the default 70 characters. A 60-character line width would probably look best for this letter. Also, remember that the default tab stops are set at every eight characters; an eight-character indent is probably too deep for paragraphs in the letter. Change the indent to five. To change the settings for line width and tabs, follow these steps:

1. Press the RECONFIGURE OPTIONS key.
2. When the RECONFIGURE OPTIONS screen appears (showing your default profile), make the changes shown in the following example.

Type remarks that identify this profile.

These two options should be set to ON.

```

Profile * from $SYSTEM.SYSTEM.TEDPROFL, created 18JUN87.
Remarks: tabs x 5; line width 60
Align Width: 60
Overlap (0-12): 2
Paragraph Lines: 6
Printer Page Size: 0
Blank Fill on 3270: ON
Tab Stops (2-238): 6 11 16 21 26 31 36 41 46 51 56 61 66
Word Stops: " !"#$%()*+,-./:;<=> ?@[\\]^_`{|}~"
Sentence Stops: "!.? "
Format Stops: "!#%&'()*+,-./:;<=>?@[\\]^_`{|}~"
SEARCH Options:
REPLACE Options:
OPENWINDOW Options:

Auto Word Wrap : ON
Exit Confirmation: ON
Auto Renumber: ON
RETURN Function: ON

Page 1 of 2
    
```

Type 60 here.

Type numbers in increments of five. The left margin, column 1, is always an implicit tab stop.

3. Before you exit from the screen, check the right-hand column of the screen to see that the fields called auto word wrap and RETURN function are set to ON. TEDIT needs these settings to adjust line width automatically. If these fields do not read ON, type ON in the field following the option.
4. Press \*C or `RETURN` to exit from the screen, TEDIT keeps the changes you have made intact as long as you do not exit from the editor.

## Advanced Editing Features

### Adjusting Line Width as You Type

#### Adjusting Line Width as You Type

You can set TEDIT to adjust the line width as you type. That is, when you press **RETURN**, TEDIT automatically moves any words that are beyond the line width you specified down to the next line. The cursor moves along with the words. This feature is called *auto word wrap*.

The following examples illustrate how TEDIT adjusts line width when auto word wrap is in effect. Assume you are typing the sentence shown in the next example and you have the align width set to 50. If you have auto word wrap set to ON, when you press **RETURN** at the end of the line, all the text that falls to the right of 50 columns (and the cursor) move down to the next line as shown. (If you have auto word wrap set to OFF, only the cursor moves to the next line; the text stays exactly where you typed it.)

1 You have just typed this line.

2 The cursor is here; press **RETURN**.

1 After you press **RETURN**, this text moves to the next line.

2 The cursor is still at the end of the line.

Setting up auto word wrap involves making changes on both the RECONFIGURE OPTIONS screen and the RECONFIGURE FUNCTION-KEYS screen. You already made the change to the RECONFIGURE OPTIONS screen. Here are the steps to set up the RECONFIGURE FUNCTION-KEYS screen for auto word wrap.

1. Make sure that you have set both the auto word wrap and RETURN function settings on the RECONFIGURE OPTIONS screen to ON. See the previous discussion.
2. Press \*C.

3. When the response line appears, type RECONFIGURE FUNCTION-KEYS (you can abbreviate it to REC F).
4. Press \*C again. A display like Figure 4-1 appears. It is called the RECONFIGURE FUNCTION-KEYS screen. You can make changes to this screen just as you can to the RECONFIGURE OPTIONS screen.

---

**Highlight**    **The Function-Keys Part of the Profile**

A profile consists of all the settings for options and function keys that appear on the RECONFIGURE OPTIONS screen and the RECONFIGURE FUNCTION-KEYS screen. The original (default) settings are assigned by your TEDIT contact.

You can change the default settings for the function keys by changing the assignments on the RECONFIGURE FUNCTION-KEYS screen, discussed here. Changing the settings on the RECONFIGURE OPTIONS screen is discussed in Section 3.

The function key assignments shown in the profile are the same as the ones you see on the template. The ones shown in Figure 4-1 are for the unshifted keys. Page two of the RECONFIGURE FUNCTION-KEYS screen shows the shifted key assignments. To see the second page of the functions part of the profile, press the `[NEXT PAGE]` key on your terminal.

## Advanced Editing Features

### The Function-Keys Part of the Profile

Figure 4-1. The Function-Keys Part of the Profile

```
F1: *c
F2: eol
F3: line
F4: sentence
F5: paragraph
F6: window
F7: region
F8: defineregion
F9: ruler
F10: retrieve
F11: findnext
F12: search
F13: tab
F14: undo
F15: help
F16: cancel
ROLL UP: rollup
ROLL DOWN: rolldown
NEXT PAGE: nextpage
PREV PAGE: prevpage
LINEINS: insline
RETURN: newline

1 of 2
```

5. To set the auto word wrap feature, you must change the function that the `RETURN` key performs. Right now, when you press `RETURN`, TEDIT moves the cursor to a new line. To make use of auto word wrap, you set the function of the key to break the line when you press `RETURN`.

Use the arrow key to move the cursor to the last line on the screen. Type `BREAKLINE` over the word `NEWLINE`. The field should look like the following example when you finish.

```
PREV PAGE: prevpage
LINEINS: insline
RETURN: BREAKLINE
Page 1 of 2
```

Type BREAKLINE over  
NEWLINE here.

**Note** If TEDIT does not break the line when you press `[RETURN]`, check with your TEDIT contact. You may have a terminal on which the RETURN function is not available. If the RETURN function is not available on your terminal, you can assign the BREAKLINE function to another function key instead of to the `[RETURN]` key. Then you simply press the BREAKLINE function key when you reach the end of the line instead of the `[RETURN]` key. The following example shows how to type BREAKLINE over the word WINDOW.

Type BREAKLINE over  
WINDOW here.

```
F5: paragraph
F6: BREAKLINE █
F7: region
```

6. Press `*C` to exit from the RECONFIGURE FUNCTION-KEYS screen.

**Saving the Profile** You should now save your LETTER profile so you can use it again. Saving profiles was discussed in Section 3. Here is just a reminder. To save the profile you just set up, follow these steps:

1. Press `*C`.
2. On the response line, type SAVEPROFILE.
3. When the response lines appear requesting a profile name, type LETTER.
4. Press `*C` again. You have now saved the letter profile.



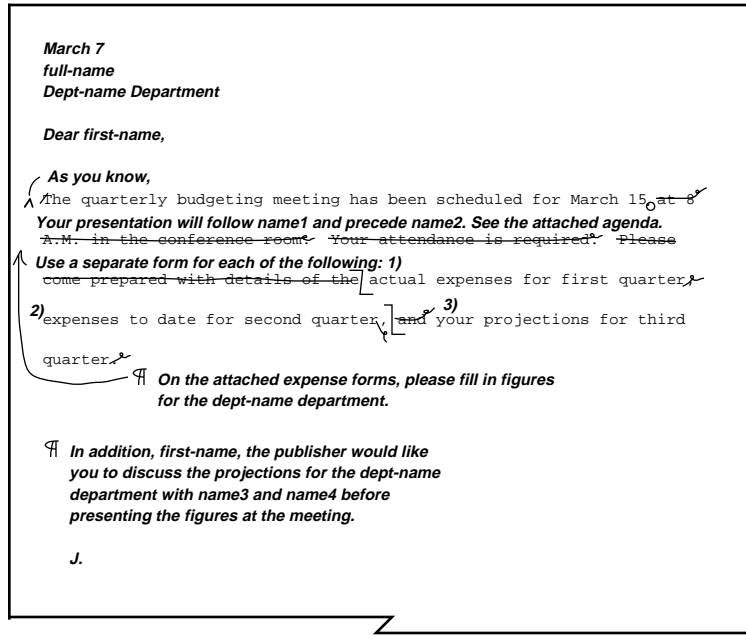
---

**Editing the Document** In the next part of this section, you use TEDIT to do the editing you see done by hand in Figure 4-2. Remember, the document you are creating in this section is a form letter. Notice the words, such as “full-name” and “dept-name,” that indicate which words change each time you personalize the form letter for the next recipient.

The editing that needs doing on the memo involves:

- Adding a heading
- Inserting text within the paragraph
- Taking the series of items out of sentence form and putting them in a numbered list
- Changing end punctuation on the items in a list
- Adding a paragraph
- Aligning the paragraphs

Figure 4-2. Editing the Memo to Create a Form Letter



**Inserting Six Blank Lines  
With One Command**

The first thing to do with this letter is to add the heading. To do that you insert seven lines (including the blank) at the beginning of the letter. The easiest way to do that is to insert a paragraph.

Press INSERT, then press PARAGRAPH.

TEDIT inserts six blank lines because TEDIT, by default, defines a paragraph as six lines. While you still have the command INSERT on the status line, press LINE to insert one more blank line.

## Advanced Editing Features

### Changing the Size of a Paragraph

In the seven blank lines you inserted above the text, type the lines you see in the following example. Remember to use the terminal `TAB` key to move to the next line, rather than the `RETURN` key. You have the `RETURN` key set to BREAKLINE; so if you press `RETURN` at the ends of these short lines, TEDIT inserts a blank line below the line the cursor is in. Also, remember to use the TEDIT TAB key to start the paragraphs in the right place.

Type these lines.

```
March 7
full-name
Dept-name Department

Dear first-name,

    As you know,
The quarterly budgeting meeting has been scheduled for March 15 at 8
A.M. in the conference room. Your attendance is required. Please
come prepared with details of the actual expenses for first quarter,
expenses to date for second quarter, and your projections for third
quarter.
```

### Highlight Changing the Size of a Paragraph

If you find that you generally want to insert more or fewer lines than six at a time, you can change the number of blank lines TEDIT inserts as a paragraph.

To change the size of an inserted paragraph, press RECONFIGURE OPTIONS.

On the RECONFIGURE OPTIONS screen, find the Paragraph Lines field. See the following example; the default is set to six lines. You can change the number to anything you want. If you want to save the profile so that it always has the new number of lines you set for the paragraph, remember to give the SAVEPROFILE command after you change the setting.

The number of blank lines TEDIT inserts when  
you press INSERT plus PARAGRAPH

```
Align Width: 60
Overlap (0-12): 2
Paragraph Lines: 6
Printer Page Size: 0

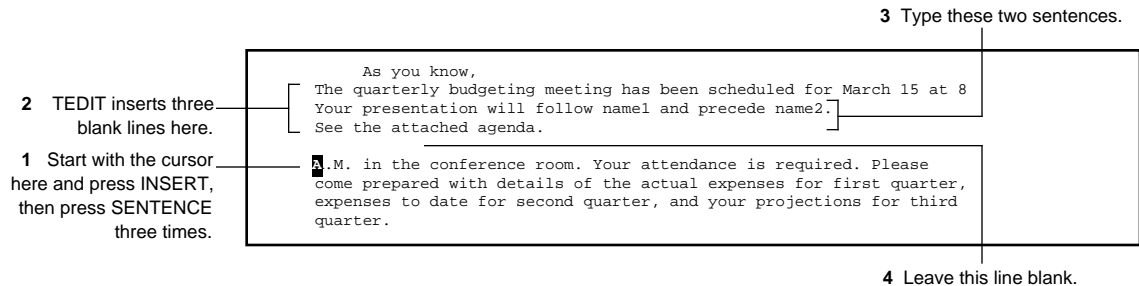
Auto Word Wrap : ON
Exit Confirmation: ON
Auto Renumber: ON
RETURN Function: ON
```

**Breaking Lines and Inserting Sentences**

In the first paragraph you replace a smaller amount of text with a larger amount. To do this replacement you must break the lines differently and insert some blank space within the paragraph in which to put the extra text. The SENTENCE INSERT command is useful for both procedures: breaking a line at the cursor position and inserting a blank line of space.

Refer to Figure 4-2 to recall the editing planned for the letter. Follow these steps and look at the following examples.

1. Put the cursor on the third line in the paragraph, the one that starts with "A.M."
2. You should still have the INSERT command on the status line. Press SENTENCE three times. (You could use the RETURN key with the BREAKLINE function instead of SENTENCE INSERT if you want. Or, in this example you could use INSERT plus LINE to do the same thing.)
3. Type the two sentences shown in the next example. Remember to use the terminal TAB key to return to the next line; do not use the RETURN key.



4. Move the cursor to what is now the start of the second paragraph, the line that starts with "A.M." Space over the first five characters to erase them.

## Advanced Editing Features

### Breaking Lines and Inserting Sentences

5. Type the first line of text in the next example right over the text that is already there. Space over the word "Please" at the end of the line to erase it. Press the terminal **TAB** key, not **RETURN** to start the next line. Type the rest of the sentence as shown in the following example. Then space over the word "the" to erase it.

1 Spaced over these characters to erase them.

2 Typed this line right over the old text.

3 Spaced over "Please" to erase it.

4 Typed the rest of the sentence here.

5 Spaced over "the" to erase it.

```
As you know,
The quarterly budgeting meeting has been scheduled for March 15 at 8
Your presentation will follow name1 and precede name2.
See the attached agenda.

On the attached expense forms, please fill in figures
for the dept-name department, actual expenses for first quarter,
expenses to date for second quarter, and your projections for third
quarter.
```

6. Put the cursor on the "a" of "actual" and press the **SENTENCE** key twice, once to break the line and once to insert a blank line. (**INSERT** should still be on the status line.) See the following example.

With the cursor here, press **INSERT**; then press **SENTENCE** once to break the line. Press **SENTENCE** again to add this blank line.

```
On the attached expense forms, please fill in figures
for the dept-name department.

actual expenses for first quarter,
expenses to date for second quarter, and your projections for third
quarter.
```

7. Move the cursor back to two spaces after the period and type the sentence shown in the next example. Type as many characters as you can fit on one line, then press `RETURN`. If you have auto word wrap working, words that exceed the 60-character line width automatically wrap to the next line when you press the `RETURN` key. If you do not have auto word wrap, use `INSERT SENTENCE` to insert another line in order to type the last phrase.

Type this sentence  
in the space you  
inserted.

```
On the attached expense forms, please fill in figures
for the dept-name department. Use a separate form for
each of the following:
actual expenses for first quarter,
expenses to date for second quarter, and your projections for third
quarter.
```

### Making a List From a Series in a Sentence

TEDIT makes it easy to make a numbered, line-by-line list out of a series in a sentence or paragraph; that is, to reformat without retyping. To make the list, follow these steps:

1. Move the cursor down to the second line of the next paragraph. Using the arrow keys, put the cursor on the "a" of "and." See the next example.

Put the cursor here.

```
actual expenses for first quarter,
expenses to date for second quarter, and your projections for third
quarter.
```

2. Press `SENTENCE` to break the line at the cursor position. (You should still have `INSERT` on the status line.) Your screen should look like the following example.

```
actual expenses for first quarter,
expenses to date for second quarter,
and your projections for third
quarter.
```

## Advanced Editing Features

### Changing End Punctuation and Finishing Up

- Next, to adjust the third item so that it takes up only one line, put the cursor anywhere on the line that starts with "and." Press ALIGN; then press LINE. ALIGN followed by the text item LINE rejoins a broken line. See the following example.

```
actual expenses for first quarter,  
expenses to date for second quarter,  
and your projections for third quarter.
```

- At the beginning of each of these three lines, press the CHAR INS key three times to make room for the numbers for each item. See the following example. After you type the "3)" use CHAR DEL to remove the word "and." Your screen should look like the following example.

Type these numbers.

Removed the word  
"and" here.

```
1) actual expenses for first quarter,  
2) expenses to date for second quarter,  
3) your projections for third quarter.
```

### Changing End Punctuation and Finishing Up

There is only a little more editing to do on the letter: remove the end punctuation from the items in the list, clean up the first sentence in the letter, and add the final paragraph. Follow these steps to complete the first draft of the form letter:

- To remove the end punctuation from the items in the list, put the cursor on the line that contains item 3 and press FORWARD. Then press EOL. The cursor moves to the end of the line. Press BACKSPACE once, then erase the period with the SPACE bar.
- To move up to the previous lines, press BACKWARD. Then press EOL twice. The cursor moves backward to the end of the line it is on, then again to the end of the previous line. Erase the comma with the SPACE bar. Press EOL twice again to move to the first line of the list. Erase the comma.

3. Press EOL twice more. If you continue to press EOL, you continue to move backward along the ends of the lines. Press EOL until you move backward to the second line of the letter. Backspace until the cursor is in the space after "March 15." Type a period. Press `CHAR DEL` to delete the rest of the characters on the line.
4. Press the terminal key labeled `BACKTAB` (`SHIFT` plus `TAB`) to move to the beginning of the line. Change the first letter on the line from "t" to "T."
5. Aligning the first two paragraphs at 60 characters per line is the next step. First, move the cursor up to the first line of the first paragraph. Press `CANCEL` to remove the text command `BACKWARD` from the status line. Then press `PARAGRAPH` to make it the persistent item. Press `ALIGN` to align the paragraph at the right width. Notice that each time you press `ALIGN`, `TEDIT` aligns the text and moves forward to the next paragraph.
6. Do not align the list; instead, press `FORWARD` (with `PARAGRAPH` still on the status line) to move past it.
7. Now all you have to do is add the final paragraph. Move the cursor down one line and press the `TEDIT TAB` key. Type the first line you see in the next example. When you get to the end of the word "you" in the first line, press the terminal `TAB` key to go back to the left margin. (If you press `RETURN`, the cursor aligns underneath the six-character paragraph indent.) For the rest of the lines in the paragraph, type as many words as will fit on a line, then press `RETURN`. `TEDIT` adjusts the line width as you type.
8. Initial the letter as shown.



## Advanced Editing Features

### Personalizing a Form Letter

The letter now looks like the following example.

The diagram shows a form letter with several annotations on the left side and two annotations at the top. The letter text is as follows:

March 7  
full-name  
Dept-name Department

Dear first-name,

As you know, the quarterly budgeting meeting has been scheduled for March 15. Your presentation will follow name1 and precede name2. See the attached agenda.

On the attached expense forms, please fill in figures for the dept-name department. Use a separate form for each of the following:

- 1) actual expenses for first quarter
- 2) expenses to date for second quarter
- 3) your projections for third quarter

In addition, first-name, the publisher would like you to discuss the projections for the dept-name department with name3 and name4 before presenting the figures at the meeting. Thank you.

J. ■

Annotations:

- Made this letter lowercase. (points to "Dear first-name,")
- Added a period here. (points to the period after "See the attached agenda.")
- Removed end punctuation. (points to the end of the list items)
- Added this paragraph. (points to the paragraph starting with "In addition...")
- Initialed the letter. (points to "J. ■")
- Deleted "at 8". (points to the space between "See" and "the attached agenda.")
- Aligned these paragraphs. (points to the two paragraphs starting with "On the attached..." and "In addition...")

### Personalizing a Form Letter

Now that the letter is written, you can replace the generic names with the actual names that make each letter personal. Use the names and departments in the agenda created in Section 3. You can look at the agenda without leaving the LETTER file by opening a window into the AGENDA file. When you have read the information you want, simply close the window. Refer to "Opening a Second Window" in the beginning of this section if you have forgotten how to do it.

To practice the TEDIT features you are learning, you may want to create eight form letters, one for each department head. If you want to create eight different letters, there are several methods you can use: You can print them as you create them, copy them into a separate file, or use the TFORM form-letter feature (see Part Two, Section 11). None of these methods is discussed here.

Even if you decide to use the TFORM letter feature, you should still read about the TEDIT features that make searching for and replacing specific text easy. Those features are discussed next; to learn them you only need to create one personalized form letter.

**Searching for Words** To personalize the form letter, replace the generic names with actual names. Use the SEARCH or the REPLACE command to find and replace the names. When you use SEARCH, TEDIT locates the name by placing the cursor on the first letter of the word you specify in the command. Then you can type over the old text to replace it with new text. Follow these steps:

1. Look at the letter you just finished and find one of the generic names you want to replace. In this example, use “dept-name.” It occurs three times; one of the times you replace it, the replacement needs an uppercase letter.
2. Look on the template and locate the key labeled SEARCH. Press it.
3. TEDIT displays the response lines shown in the next example. Fill them in as shown.

*Search for:* Type “dept-name” in this field. Whatever you type in this field is called the *search-text*.

*In lines:* Leave this field as is, F/L. TEDIT automatically fills this field in for you with the designation F/L, which means that TEDIT looks for the search-text in all the lines from the first line in the file to the last one. You can change this field to put in specific line numbers or a designation such as \*/B, which tells TEDIT to search between the cursor position (\*) and the bottom of the screen (B).

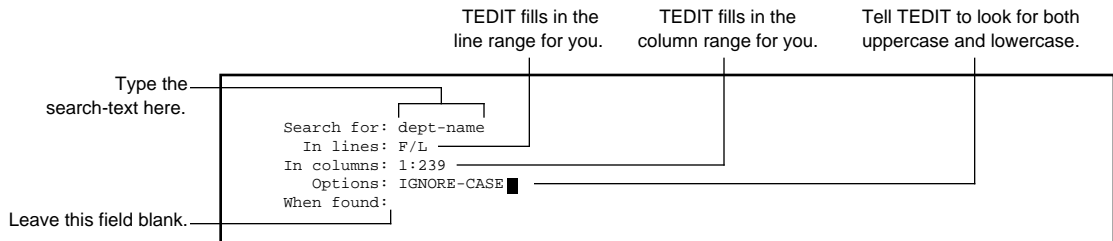
*In columns:* Leave this field as is, 1:239. TEDIT automatically fills this field in for you with the designation 1:239. Column 1 is the leftmost column on the screen. At any given time, you can see 79 columns on the screen; but with TEDIT you can put text in 239 columns by scrolling the screen horizontally. The point here is that TEDIT automatically looks in all possible columns for the search-text unless you limit the search to a specific place—columns 1 through 5, for example.

*Options:* Put IGNORE-CASE in this field. You can extend the search to include both uppercase and lowercase versions of the search-text by typing IGNORE-CASE (or the abbreviation I) in the Options field. If you do not tell TEDIT to find all occurrences of “dept-name” regardless of whether they are typed in uppercase or lowercase letters, TEDIT only locates the text that looks exactly like what you typed in the Search-for field.

You can limit the search to single words; for example, if you want to search for only “you,” put WORD-DELIMITED or W in the Options field. Otherwise, TEDIT searches for all the words with “you” in them, including “your,” “yours,” and “bayou.”

*When found:* Tell TEDIT what to do after finding the search-text. You can put any commands here that you would put on the \*C response line. For example, you could insert a line or change a word to uppercase letters.

The When-found option is best used with compound commands, such as two or more commands and their options strung together. If you have a single change to make to the search-text, it is easier to use the REPLACE command from the beginning rather than using the When-found option on the SEARCH command. REPLACE is discussed a little later in this section.



4. Press \*C to begin the search. TEDIT finds the first occurrence of “dept-name,” positions the cursor on the word, and displays a message on the status line confirming it. Type “Production” here right over “dept-name”; press `CHAR INS` once to make room for the new word.

5. Press the FINDNEXT key to find the next occurrence of dept-name. Type "production" over "dept-name," inserting space as necessary to fit. Continue to press FINDNEXT until TEDIT has found all the occurrences of your search-text. (Press FINDPREV to go back to previous occurrences of the search-text.) When TEDIT has found the last occurrence of the search-text, it displays the following message on the status line:

```
1) SEARCH didn't find the search-text          C00.
```

---

## Highlight Using Ranges for Searching

When you search a file, especially a large one, it is often useful to specify a precise area in which you want TEDIT to make the search. You specify the area by giving TEDIT *ranges* of lines and columns.

As shown in the previous example, you can specify lines with certain words indicating place, such as FIRST, LAST, TOP, BOTTOM, REGION (the text item) and \* (cursor position). You can also specify lines with line numbers. In addition, you can specify lines plus or minus a line number; for example, you can search from five lines below the top of the screen (T + 5) through ten lines above the bottom of the screen (B - 10). See the next example.

You can specify columns with words, numbers, and combinations of words and numbers. You can use + or - numbers with columns just as you can with lines. The words you can use to indicate place in columns are FIRST, LAST, LEFT, RIGHT, END, and \* (cursor position). Refer to the *PS TEXT EDIT Reference Manual* for more information on ranges.

## Advanced Editing Features

### Using Patterns for Searching

Here is an example of how to use the range feature during a search.

Search five lines from the top and ten from the bottom of the screen.

Search from one column past the cursor position to nine lines from the right of the screen.

```
Search for: name
In lines: T+5/B-10
In columns: *+1:R-9
Options:
When found:
```

### Highlight Using Patterns for Searching

In your letter, you could search once for all the *patterns* of “name” followed by a number instead of using SEARCH four times, once for each of the numbered names. To tell TEDIT to search for variations of an item, you put the text that varies in brackets in the Search-for field and use the EXTENDED-SEARCH option. See the following example.

Search for any occurrence of “name” followed by the number 1, 2, 3, or 4.

Always use EXTENDED-SEARCH (or simply E) when searching for patterns.

```
Search for: name[1234]
In lines: F/L
In columns: 1:239
Options: EXTENDED-SEARCH
When found:
```

Another way to use patterns when searching is to use *wild cards* to match any character or certain characters in certain positions. You can search for very specific things (for example, all the words that end with “ing”) by using wild cards in patterns. Wild cards for patterns are discussed in detail in the *PS TEXT EDIT Reference Manual*.

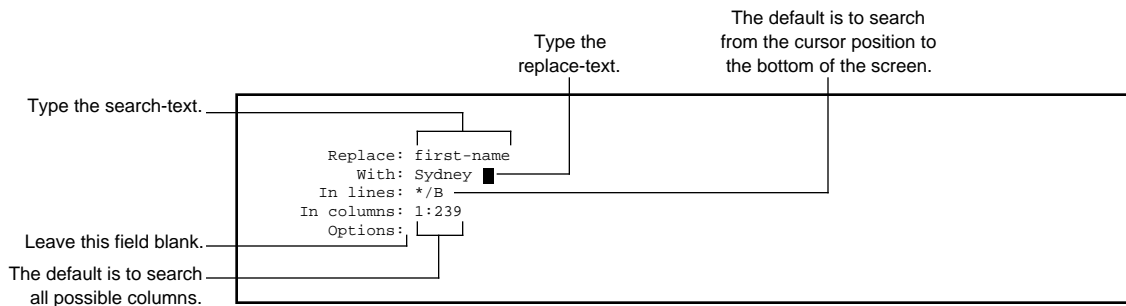
**Note** If you forget what the options for the response lines are for SEARCH (or for any of the other response line fields), you can get help by pressing the HELP key while you have the SEARCH response lines displayed on the screen. If you press HELP once, you get a brief reminder about what to do; if you press HELP again, you get a more detailed discussion.

**Replacing Words** The REPLACE command has the SEARCH function built into it. The way you usually use REPLACE is to have TEDIT search the entire file and replace all the search-text with *replace-text* in one step. During this process, TEDIT leaves the cursor where it is, makes all the replacements, and gives you a message on the status line when the task is finished.

You can, however, have TEDIT stop at each occurrence of the search-text and wait for you to confirm that you want the search-text replaced. You ask for this feature by typing CONFIRM (or C) in the Options field of the REPLACE response lines.

REPLACE is the best command to use when you want to make an exact substitute of a specific word or phrase with another throughout the entire file. A good example in the form letter is "first-name." It must be replaced twice with exactly the same name. To perform the replacement, follow these steps:

1. Press the REPLACE key.
2. When the response lines appear, fill them in as shown in the following example.



3. Press \*C to issue the command. TEDIT replaces all occurrences of the search-text with the replace-text within the lines and columns you specified. When TEDIT has finished the replacements, it displays a message on the status line confirming that.

## Advanced Editing Features

---

### Replacing Words

**Note** You may find that in most cases (especially if your file is longer than one screen), it is preferable to have TEDIT do the replacements throughout the entire file rather than from the cursor position to the bottom of the screen (\* /B). To make the replacement everywhere in the file with one command, specify F/L (first through last line) or A (for ALL lines in the file) on the In-lines field of the REPLACE response lines.

---

If you did not replace any of the numbered names in the letter, use SEARCH or REPLACE to do so now. To complete the letter, put an actual full name in the heading. See the following example for appropriate names with which to replace the remaining search-text. When you finish all the replacements, realign the paragraphs. The letter looks like the following example.

```
March 7
Sydney Strauss
Production Department
```

```
Dear Sydney,
```

```
As you know, the quarterly budgeting meeting has been
scheduled for March 15. Your presentation will follow
Riveira and precede Yuen. See the attached agenda.
```

```
On the attached expense forms, please fill in figures
for the production department. Use a separate form for each
of the following:
```

- 1) actual expenses for first quarter
- 2) expenses to date for second quarter
- 3) your projections for third quarter

```
In addition, Sydney, the publisher would like you to
discuss the projections for the production department with
Medici and Kitahara before presenting the figures at the
meeting. Thank you.
```

```
J.
```

**Summary** In this section you created a letter, using a preexisting memo for the rough-draft text. You used the WINDOW feature to copy the text from one file to another. You set up a customized profile for the letter (changing the line width and tabs and using the auto word wrap feature), and you saved the profile for later use.

Then you edited the memo to make it into a letter. You inserted a blank paragraph and added a heading. You added sentences and created a numbered list from a series of items in a paragraph, using INSERT plus SENTENCE to break lines. You used ALIGN plus LINE to rejoin broken lines. Then you used FORWARD plus EOL to clean up the end punctuation. Finally you aligned all the lines to the width you set in the profile.

After editing the letter to set it up as a form letter, you learned how to use the SEARCH and REPLACE features to personalize the letter. For more information about personalizing and printing form letters, refer to Section 11 in Part Two, "Using PS TEXT FORMAT."



---

# 5 Creating an Oversized Table

---

**Overview** In this section you create a table that is both longer and wider than what you can see on the screen. This table is to be used as an expense form for the department heads listed on the agenda in Section 3. You use the agenda in Section 3 as the rough-draft copy for the table. In this way, you avoid as much retyping as possible. The features discussed in Section 5 include:

- Transferring text from one file to another using windows
- Defining and manipulating regions of text
- Scrolling the text vertically and horizontally
- Assigning different commands to function keys
- Giving several commands at one time using the \*C response line
- Reissuing a command without retyping
- Moving around in and editing a file that is longer and wider than what you can view on the screen
- Setting the profile from the \*C response line
- Learning quick ways to insert horizontal and vertical rules
- Creating and saving a special profile just for the table

## Creating an Oversized Table

### Transferring Text to the New File

---

#### Transferring Text to the New File

You learned how to copy text from one file to another in Section 4. You can use that same method here to bring text from the AGENDA file you created in Section 3 to a new file called TABLE, which you work on in this section. In this section you learn some shortcuts to create files and to open windows.

#### Creating the File

At the command interpreter prompt, type TEDIT followed by the file name, TABLE, and an exclamation mark (!). The ! mark causes TEDIT to create the file without asking you whether you want to create it. Here is an example:

```
5> TEDIT TABLE! █
```

#### Note

Make sure you know what volume and subvolume you are in before you create a file using the exclamation mark (!). You can check your volume and subvolume by giving the WHO command at the TACL or command interpreter prompt.

---

#### Three Ways to Open a Second Window

As soon as TEDIT creates the TABLE file, you can open a window into the AGENDA file, which you created in Section 3. You can use the OPENWINDOW command in several ways. First, the way you learned it in Section 4:

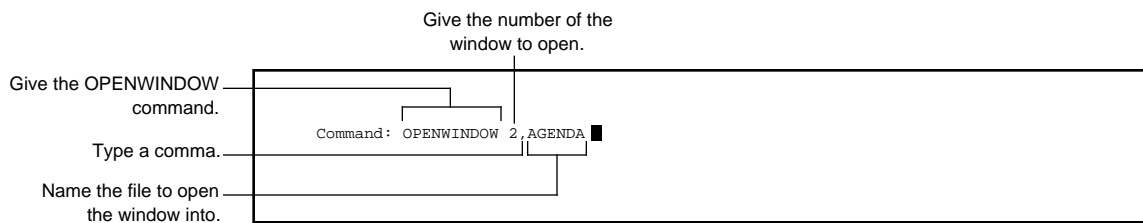
1. Press \*C. Type OPENWINDOW on the response line. Press \*C again.

```
Command: OPENWINDOW █
```

2. When the OPENWINDOW response lines appear, fill them in as shown.

```
OpenWindow: 2
File: AGENDA █
Options:
Size:
```

Second, you can open a window by typing all the information on the \*C response line and bypassing the OPENWINDOW response lines. You do that in one step:



**Note** The particular way you give commands on the \*C response line (for example, where you use spaces and commas) is determined by command syntax. Each TEDIT command has specific syntax; it must be typed in a specific way in order to work. You can get the syntax for each command from the reference card, the *PS TEXT EDIT Reference Manual*, or the online help.

Third, you can accomplish the entire process from the command interpreter. You create the new file and open a window all in one line, separating the two commands with a semicolon. See the following example:

```
5> TEDIT TABLE!; OPENWINDOW 2,AGENDA █
```

## Creating an Oversized Table

---

### Copying a Region From One Window to Another

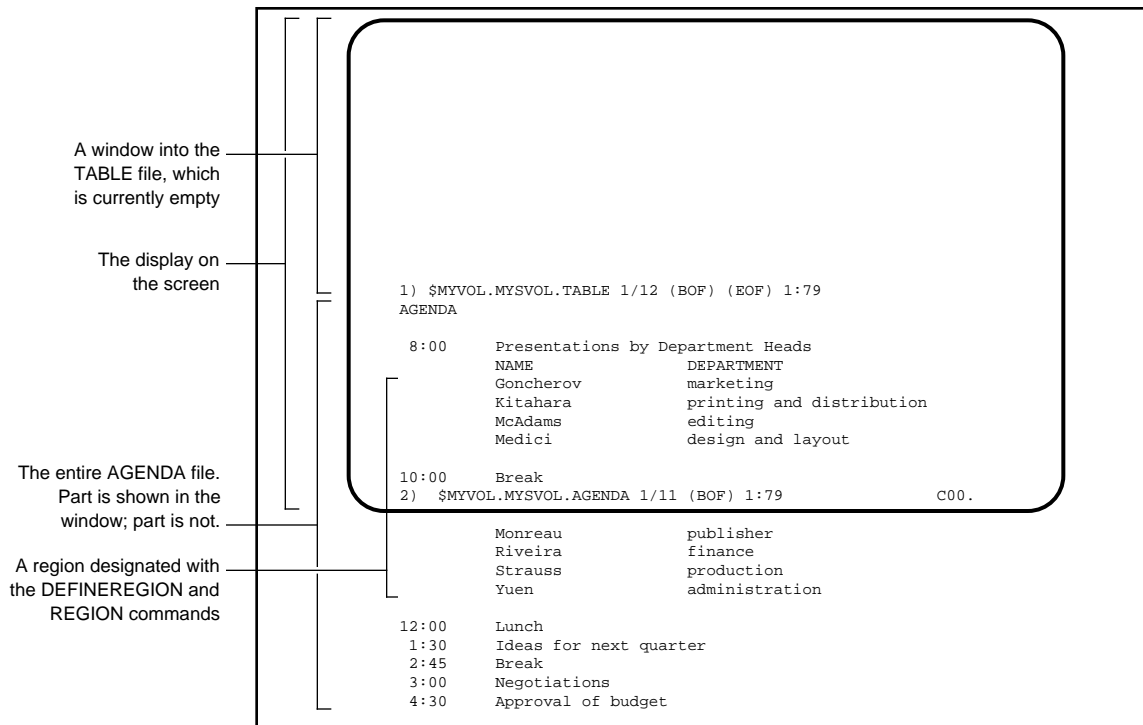
#### Copying a Region From One Window to Another

Now that you have two windows open, you are ready to copy some text. Copying text from this file requires more steps than copying text from the MEMO file for two reasons:

1. The text you want to copy is not in the form of a text item that TEDIT recognizes, such as SENTENCE or PARAGRAPH. Therefore, you must define the area of text yourself.
2. You cannot see all of the text you want to copy in the window you opened.

You can solve both these problems in one step: define a text item called REGION by marking the boundaries—the beginning and end—of the area with the cursor, then copy the REGION. Look at Figure 5-1.

Figure 5-1. Defining a Region



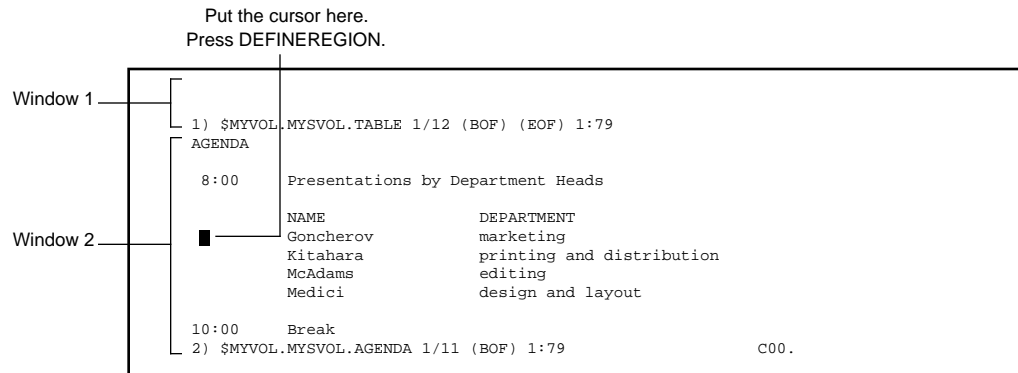
To define and copy a REGION of text from the AGENDA file to the new TABLE file, follow these steps:

1. With a window open into each file (see the previous discussion), move the cursor so that it is in column 1, line 1 of the region you want to copy, the line containing "Goncherov." The cursor position defines the beginning of the region.

## Creating an Oversized Table

### Copying a Region From One Window to Another

2. Press the DEFINEREGION key. The cursor blinks once to acknowledge the DEFINEREGION instruction. See the following example.



3. Move the cursor to the end of the region you want to copy. Because this window is only 11 lines deep, you cannot see the end of the region. To position the cursor at the end of the region, you must *scroll* the text up on the screen until you see the end of the region. To scroll the text up, press the terminal key labeled `NEXT PAGE`.

When you press the `NEXT PAGE` key, TEDIT scrolls the text up by one page. TEDIT considers a *page* to be the number of lines that fit in a window. If you only have one window open, the page covers the entire screen.

---

**Note** You can change the size of a window with the SIZEWINDOW command. On the \*C response line, type SIZEWINDOW (or SI) followed by number of lines deep you want the window to be.

---

Using the arrow keys, position the cursor at the farthest right and bottommost edge of the region. See the following example.

4. With the cursor marking the end of the region, press the REGION key. Then press COPY. See the next example.

Put the cursor here.  
Press REGION, then  
press COPY.

```

1) $MYVOL.MYSVOL.TABLE 1/12 (BOF) (EOF) 1:79
10:00      Break
           Monreau      publisher
           Riveira      finance
           Strauss      production
           Yuen          administration
12:00      Lunch
1:30      Ideas for next quarter
2:45      Break
2) $MYVOL.MYSVOL.AGENDA 10/20 (BOF) 1:79
C00.
```

5. Press `HOME` to move the cursor into the top window (in the empty TABLE file).
6. Press RETRIEVE. TEDIT retrieves the entire region you defined.

**Note** You can define a region from its beginning to its end or from its end to its beginning. In either order, mark the topmost, leftmost edge and the bottommost, rightmost edge with the cursor, pressing the DEFINEREGION key first and then the REGION key.

**Closing the Second Window**

Now that you have copied all the text you want from the AGENDA file, you can close the window. Here are three ways you can close a window:

1. Press \*C. When you see the response line, type CLOSEWINDOW 2. Press \*C again.
2. Put the cursor in the window that you want to keep open. Press \*C. Type CLOSEWINDOW \*\*. Press \*C again.
3. Put the cursor in the window that you want to close. Press \*C. Type CLOSEWINDOW \*. Press \*C again.

## Creating an Oversized Table

---

### Closing the Second Window

After you close the window into the AGENDA file, only the TABLE file remains on your screen. The TABLE file should look like the next example.

■	Goncherov	marketing
	Kitahara	printing and distribution
	McAdams	editing
	Medici	design and layout
10:00	Break	
	Monreau	publisher
	Riveira	finance
	Strauss	production
	Yuen	administration

---

**Note** You can open two windows into the same file. Perhaps you want to copy text from the beginning of a file to the end of a file that is longer than one screen. Rather than scrolling the text back and forth, simply use the OPENWINDOW 2 command and give the name of the file you currently have open in window 1. One use for this feature would be to check a table of contents against the headings in a report; with two windows open, you can change the headings in both areas easily and quickly when they are both in front of you on the screen.

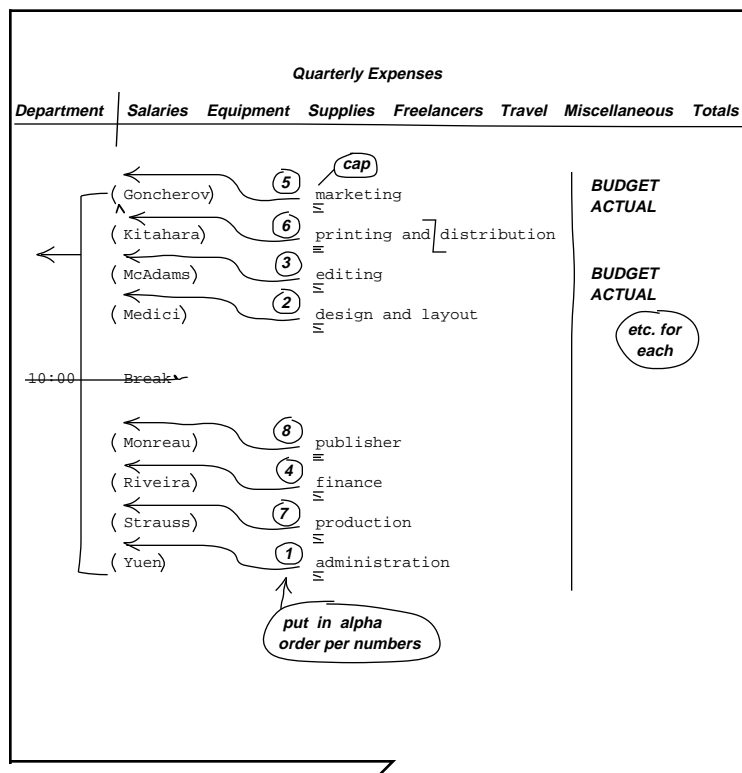
---



**Changing the Agenda Into a Table**

Before you start to do any editing, think about the overall restructuring that is necessary to change the agenda into a table. The table you create in this section is the expense form referred to in the letter in Section 4. Refer to the hand-edited text in Figure 5-2 to see the editing you will do with TEDIT.

Figure 5-2. Editing the Agenda to Create a Table



## Creating an Oversized Table

### Establishing Row Headings

**Establishing Row Headings** To begin creating the table, you should first establish a row heading for each of the departments. In this table the department name comes first; the department head's name is in parentheses below it. You can change the order—without retyping—in just a few steps, as discussed in the following paragraphs.

#### Putting Two Columns Into One

Move the department names over to the left so that they are in the same column as the people's names. Follow these steps:

1. Put the cursor on the first department name (department names are in the second column). See the following example.

Put the cursor here;  
press SENTENCE, then  
press INSERT.

Goncherov	marketing
Kitahara	printing and distribution
McAdams	editing

2. Press SENTENCE, then press INSERT. TEDIT breaks the line at the cursor and moves the text to align underneath the line above it. See the next example.
3. Move the cursor down two lines to the next department name and press INSERT again. Continue this process until you have put both columns into one column.

Put the cursor here;  
press INSERT again.

SENTENCE plus  
INSERT breaks the line  
and moves it here.

Goncherov	marketing
Kitahara	printing and distribution
McAdams	editing

4. Use LINE plus DELETE or the terminal `DEL LINE` key to delete the line containing the 10:00 break and the blank lines on either side of it. Your screen now looks like the following example.

```
Goncherov
marketing
Kitahara
printing and distribution
McAdams
editing
Medici
design and layout
Monreau
publisher
Riveira
finance
Strauss
production
Yuen
administration
```

### Moving a Column

The text is now in one column, but it is indented ten characters from the left edge. To move the text to the far left of the screen, you must delete nine columns of blanks. You could delete the blanks one at a time, using the `CHAR DEL` key for each of the lines, but that is tedious. It would be faster to delete the blanks from all the lines at once.

There is no built-in function in TEDIT that moves an entire column at one time. You can, however, use the REPLACE command to delete a column. (You can also use REPLACE to insert columns.) You can give the REPLACE command in three ways:

- Use the REPLACE response lines.
- Assign the command to a function key.
- Give the command on the \*C response line.

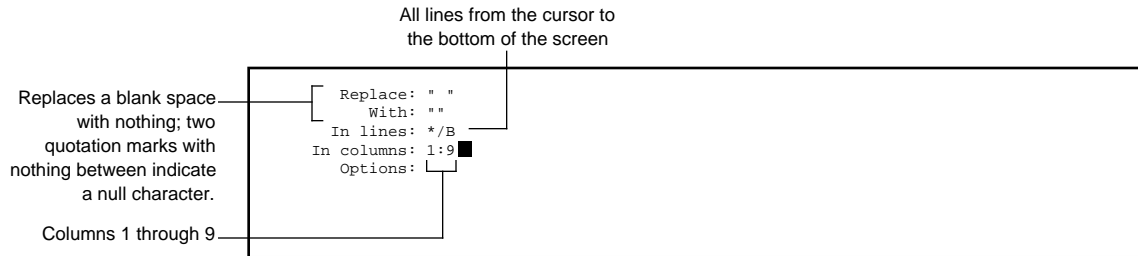
The first two methods are discussed next. Using \*C is discussed under “Giving a String of Commands in One Step” later in this section.

## Creating an Oversized Table

### Setting Up Function Keys to Move Columns

Using the REPLACE response lines is the easiest if you want to give the REPLACE command only once or you want to change it a little each time you give it. TEDIT saves the information you type on the response lines, keeping it in effect until you change it or exit from the file. Because TEDIT saves the information, you save time by not having to retype everything when you want to reissue the command or when you want to change only one field in the response lines. Here is one way to give the command to move the text to the left by nine columns:

1. Place the cursor in column 1 of the first line.
2. Press REPLACE. When the response lines appear, type the information shown in the following example.



3. Press \*C. TEDIT performs the replacement, moving all the text to the left by nine columns.

---

### Highlight Setting Up Function Keys to Move Columns

The following examples show you how to assign the commands that insert and delete columns to two function keys. Some advantages of having these commands on function keys are:

- You can use the DELETE COLUMN and INSERT COLUMN functions consecutively without having to type all the information on the response lines each time.
- You can save the functions of inserting and deleting columns in your TABLE profile.

After assigning the commands, all you have to do is press a function key each time you want to insert or delete a column. These functions are very handy when working with tabular material.

The discussion here is brief. To understand more fully how the commands work, refer to the REPLACE command and the PATTERNS topic in the *PS TEXT EDIT Reference Manual*.

To assign to function keys the command to delete any column the cursor is in and the command to insert a blank column, follow these steps:

1. Pick two function keys that you do not think you will use when creating tables. (You can always change them later if you need to.) For this example, use WINDOW and REGION.
2. Press \*C. When the response line appears, type RECONFIGURE FUNCTION-KEYS (or simply REC F). Press \*C again. TEDIT displays the first page of the function key settings.
3. Using the terminal `[TAB]` key, tab down to the first key you are changing. Type the command to delete a column exactly as shown in the following example.

The syntax for the command tells TEDIT to replace any two characters with just the second character of the pair (which essentially deletes the first character). TEDIT makes this replacement in all the lines from the cursor position to the bottom of the screen (\* /B). TEDIT replaces the character in the column the cursor is in (\*) with the character one column to the right of the cursor (\* + 1). The E stands for EXTENDED-SEARCH, which you must always use with a wild-card character such as “?”.

Assign the DELETE COLUMN command to `[F6]`.

You might prefer F/L here to do the entire file at once.

Type this line exactly as shown.

```
F5: paragraph
F6: REPLACE ??, @2, */B, **+1, E
F7: region
F8: defineregion
```

## Creating an Oversized Table

### Setting Up Function Keys to Move Columns

4. With the terminal `TAB` key, tab down to the second key you are changing. Type the command to insert a blank column exactly as shown in the following example. The syntax for the command tells TEDIT to replace a null character (“”) with a blank space (“ ”) in all lines from the cursor to the bottom of the screen in the column where the cursor is currently positioned.

Assign the  
INSERT COLUMN  
command to `F7`.  
Type this line  
exactly as shown.

```
F5: paragraph
F6: REPLACE ??, @2, */B, *:*+1, E
F7: REPLACE " ", " ", */B, *:*
F8: defineregion
```

5. Press `*C` to leave the RECONFIGURE FUNCTION-KEYS screen with the new key assignments intact.
6. Now you can move the entire column of text in the TABLE file to the left or to the right. Put the cursor in the HOME position. Press the function key that contains the INSERT COLUMN function (it should be the key labeled WINDOW on the template). Each time you press it, the text in every line below the cursor position moves to the right by one column. Then try pressing the function key that deletes a column; the text moves to the left by one column each time you press the key.

Remember, if you have the cursor in the wrong position and accidentally delete a column of text instead of a column of blanks, you can always use UNDO.

---

**Note** TEDIT saves the REPLACE commands you give with the function keys on the REPLACE response lines. If you press the REPLACE key now, you see the information you typed on the RECONFIGURE FUNCTION-KEYS screen. It is important to remember this because some REPLACE commands do not work with the EXTENDED-SEARCH option used in the delete column function in step 6 above. You only use EXTENDED-SEARCH when you have a wild-card pattern (such as ? or @) in the search-text or replace-text. If you want to give a REPLACE command that does not use a wild-card pattern, make sure the Option field does not contain EXTENDED-SEARCH.

If you pressed the REPLACE key to look at the response lines and now do not want TEDIT to perform the replacement, simply press CANCEL.

---

### Reversing Lines and Realphabetizing

To set up the horizontal headings so that the department names come first, you need to reverse the current order of the names. You used this method in Section 3 to put the names on the agenda in alphabetical order. Use the MOVE and RETRIEVE functions to put the names and departments into the correct order. Refer to Figure 5-2 if you want and follow the steps shown in the next example.

1. Put the cursor on the line containing "administration"; this is the department you want listed first in the series of horizontal headings. Press LINE. Press MOVE. TEDIT moves the line to the scratchpad.
2. Move the cursor to the line containing "Yuen," the administration department head's name. Press MOVE.

Continue to position the cursor on each line you want to move, in the order given in the next example, then press MOVE to move the lines. In this way, you reorder the list as you move each name-and-department pair from the screen to the scratchpad. When you have moved all the names, your screen is blank; everything is on the scratchpad.

## Creating an Oversized Table

### Setting Up Function Keys to Move Columns

10	—	Goncherov
9	—	marketing
12	—	Kitahara
11	—	printing and distribution
6	—	McAdams
		editing
		Medici
		design and layout
		Monreau
		publisher
		Riveira
		finance
		Strauss
		production
		Yuen
		Administration

3. Press RETRIEVE. All the lines are retrieved in the order you moved them. They should now be in alphabetical order by department name. (Remember to use EMPTYSCRATCHPAD as well as UNDO if you make a mistake.) Your screen should look like the following example.

Administration
Yuen
design and layout
Medici
editing
McAdams
finance
Riveira
marketing
Goncherov
printing and distribution
Kitahara
production
Strauss
publisher
Monreau



### Giving a Series of Commands in One Step

Look at the editing in Figure 5-2. The next step in editing the TABLE file is to make the first letter of each of the department names uppercase and to put parentheses around each of the department head's names. You could type the changes on each of these lines separately, using features of TEDIT you have already learned.

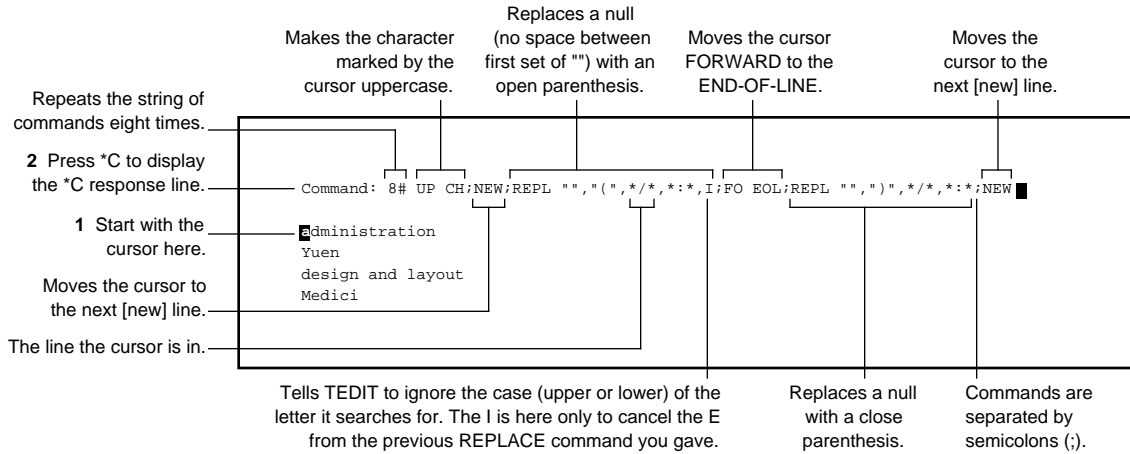
There is, however, a way to let TEDIT do repetitive tasks such as these for you. In the same way that you assigned the REPLACE command to the function keys earlier in this section, you can write a string of commands to do the following: 1) make a single character uppercase, 2) move the cursor to the next line, and 3) put parentheses around a word.

Because your need for a series of commands such as this is a fairly unusual occurrence, you probably will not want to put it on a function key and save it in the TABLE profile. Instead you can write the commands on the \*C response line. You write the commands in the same way as you would on a function key. Then you press \*C to give the command. Follow these steps:

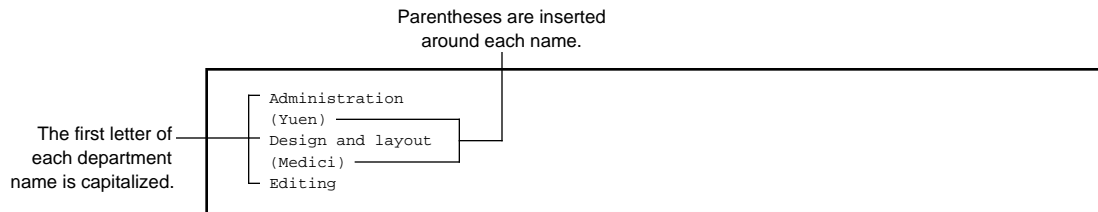
1. Put the cursor on the first letter of the first department name you want to change. This is the HOME position in this file.
2. Press \*C. When the response line appears, type the string of commands you see in the following example exactly as shown. The commands are abbreviated so that they all fit on one line. The correct abbreviations for each command are given in the reference manual and in the online help. The margin notes that accompany the example explain how the command works.

## Creating an Oversized Table

### Setting Up Function Keys to Move Columns



3. Press \*C again. TEDIT performs the string of commands you typed on the response line eight times. The file now looks like the following example. (Only part of the file is shown.)




---

**Highlight Three Ways to Repeat Commands**

**Repeat Count.** You can use a repeat count (8# in the previous example) with any command or string of commands. A repeat count is not the same as the REPEAT command, discussed in the following paragraphs. If you use a repeat count with a single command, you do not use the # sign. Type the command to copy five words on the \*C response line this way:

Repeats the command  
five times.



```
Command: 5 COPY WORD █
```

Without the # sign, the repeat count works only on the command immediately following the count. If you want to repeat several commands, as in the previous example with the long string, use the # sign after the repeat count number.

**FC Command.** If you do not want to use a repeat count, you can use the FC command to give the same commands several times in a row without retyping. FC is an acronym for fix command. Use FC to recall and reissue commands you gave with \*C. Follow these steps:

1. Press \*C.
2. When the response line appears, type FC.
3. Press \*C again. TEDIT redisplay the series of commands you just typed.
4. Press \*C again and TEDIT performs the commands again. You could use FC in this way to reissue the long string of commands in the previous example seven times instead of using the repeat count.

You can also change the command before you reissue it. Say you typed a command incorrectly and TEDIT could not perform it. You can use FC to redisplay the command line, fix the part of the command that was incorrect (without having to retype the entire line), and then press \*C to reissue the command.

***REPEAT Command.*** The third way to repeat a command is to use the REPEAT command. The REPEAT command simply reissues the last command, or string of commands separated with semicolons, that you gave. If you intend to reissue a string of commands several times, you could assign the REPEAT command to a function key (see “Using the SETPROFILE Command” later in this section). Putting REPEAT on a function key is faster than using FC if you do not want to alter the command before you reissue it.

**Highlight Giving a Series of Commands With an OBEY File**

There is another way, besides using a function key or the \*C response line, to give a string of commands: You can use an OBEY file. You put the commands you want TEDIT to perform in a separate file. You can put one command on each line or several commands separated by semicolons. Then you open the file on which you want the commands performed and give the OBEY command.

The following example shows an OBEY file that performs almost the same series of commands as the string of commands you gave on the \*C response line shown in the previous example. If you want to try out the OBEY file now, undo the previous series of commands that capitalized and added parentheses to your file. Simply press UNDO until all the editing is undone. Then, open a second window and follow these steps to create and use the OBEY file.

1. Create a new file called LAZY. Type the commands you see in the next example. You can comment on the commands within the file by prefacing the comment with the designation ?COMMENT. TEDIT ignores the comment lines when it performs the commands in the file. Put the comments in the file to remind you what the series of commands is designed to accomplish.

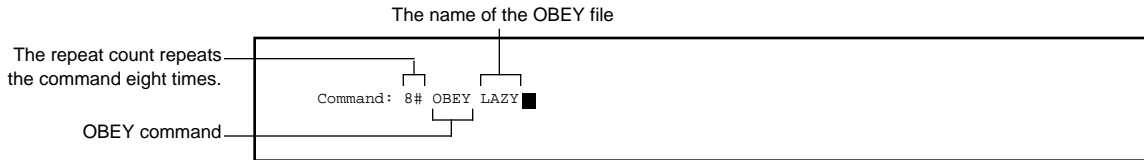
Do not put any blank space between this first set of quotation marks.

```
?COMMENT This OBEY file makes the first character of the line the
?COMMENT cursor is in uppercase. Then it goes to the next line and
?COMMENT puts parentheses around the words on the line. Then it
?COMMENT goes to the next line and stops. You can give a command !
?COMMENT repeat the sequence at this point if you want. IGNORE-CA
?COMMENT is specified in the REPLACE command only to cancel any
?COMMENT previous EXTENDED-SEARCH specification.
?COMMENT
UPPERCASE CHARACTER
NEWLINE
REPLACE " ", "( ", ** , ** , IGNORE-CASE
FORWARD END-OF-LINE
REPLACE " ", " )", ** , ** , IGNORE-CASE
NEWLINE
```

## Creating an Oversized Table

### Establishing Column Headings

2. To perform the commands in the OBEY file, exit from the OBEY file. Then open the file on which you want to perform the commands; in this case, it would be the TABLE file (which is already open).
3. Put the cursor on the first character of the first line where you want TEDIT to begin performing the commands.
4. Press \*C. When the response line appears, type the following command:



5. Press \*C again. TEDIT performs the commands in LAZY on the TABLE file.

### Establishing Column Headings

To establish where to put column headings, first decide how wide you want the horizontal headings (the department names) to be. One of the headings, "printing and distribution," is much wider than the rest. Use SENTENCE plus INSERT to move "distribution" down to the next line.

After you determine the width of the horizontal headings (these actually function as the first column of the table), you can use RULER to decide where to place the rest of the columns. You can probably tell by looking at the column headings that they will not fit in the 79-character view of the screen you are looking at now. With TEDIT, however, you can create a file that is up to 239 characters wide. To work with a file that is wider than 79 characters, you scroll the screen horizontally from left to right.

### Using RULER to Determine Placement

Using the next example as a guide, follow these steps to determine placement for the headings in the table.

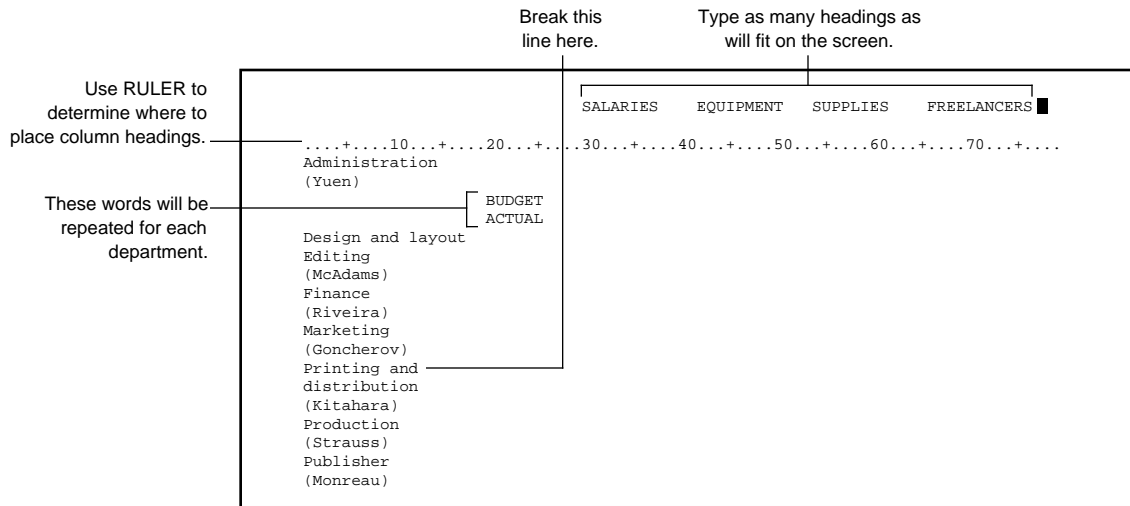
First, put in one set of row headings for each department; follow these steps:

1. Put the cursor on the line containing the widest entry. Press RULER. Use the RULER to judge where to start the column heads. Start at column 20; use the arrow key to move the cursor there.
2. Make sure the cursor is in column 20 of the line containing the words "Design and layout." Press RULER to remove the ruler. Next, press LINE. Then press INSERT.
3. Type the word "ACTUAL." Press INSERT again. Then, move the cursor back to column 20 and type "BUDGET" just above "ACTUAL."
4. Next, to put in the column headings, move the cursor up to the first line in the file. With LINE still on the status line, press INSERT four times. This creates a space for two levels of headings.
5. Space over to column 30 (you may have to press RULER again to show you where it is). Type the first column heading, "SALARIES."

## Creating an Oversized Table

### Establishing Column Headings

- To establish placement for the rest of the column headings, make each column at least twelve characters wide or put at least two spaces between it and the next heading. (You can refer to the tab stops shown in the profile at the end of this section if you want to use the same columns used in these examples.) Using the column headings listed in Figure 5-2, type the headings on the same line as "SALARIES." After you type "FREELANCERS," the screen should look like the following example.



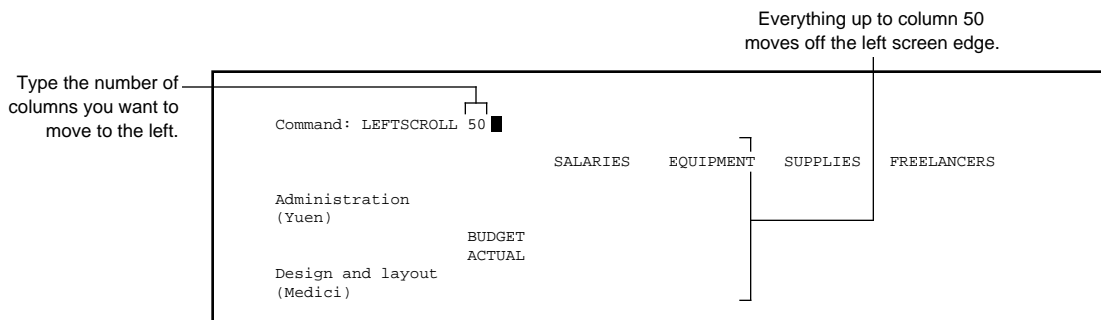
### Horizontal Scrolling

You have typed to the edge of the screen and you still have four more column headings to put in. You need to move the view of the file over so that you can put in the rest of the headings. The commands that shift the view from left to right do what is called horizontal *scrolling*. Those commands are LEFTSCROLL and RIGHTSCROLL.

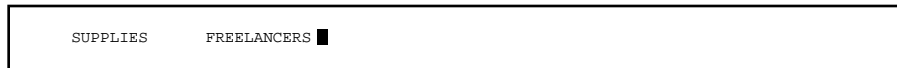
To see the part of the file that is to the right of the screen, you scroll the file to the left. Use the command LEFTSCROLL followed by the number of columns you want to scroll. To scroll the TABLE file to the left, follow these steps:



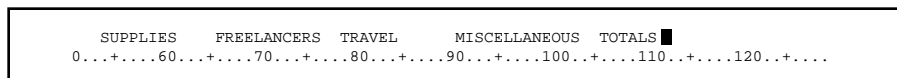
1. Press \*C. On the response line, type LEFTSCROLL 50 (or abbreviate it to LE 50) to scroll the view of the file 50 columns to the left. See the next example.



2. Press \*C again. When TEDIT performs the command, your view of the screen should look like the following example. Columns 1 through 50 are no longer visible. Notice the change to the status line: it shows columns 51:129 are currently visible on the screen.



3. With the cursor on the top line, press RULER. Notice that the numbers on the RULER start at 51 and go up, just as you would expect. Now type the rest of the column headings as shown in the following example.



4. To return to the far left, to column 1, press \*C. Type RIGHTSCROLL 50 (or abbreviate it to RI 50) and press \*C again. The view of the screen scrolls back to its previous position. You can also press  to go back to the leftmost character on the next line.

## Creating an Oversized Table

---

### Using the SETPROFILE Command to Set Tab Stops

#### Setting the Tab Stops in the Profile

You can use the TEDIT TAB and BACKTAB keys, as well as the `RETURN` key, to scroll the file to the left and right if you have set tab stops at column numbers that are both higher and lower than the 80 columns that fit on the screen. Then, for example, if you tab to column 150, the text scrolls to the left so that TEDIT can place the cursor on column 150. It usually saves time to use `RETURN`, TAB, and BACKTAB rather than using the LEFTSCROLL and RIGHTSCROLL commands.

Using the column headings you just typed to show you where to set tab stops, put the tab stops in the profile now so you can see how the scrolling feature works. Use RULER again if you do not remember the column numbers where each heading starts (or refer to the example profile at the end of this section and use the numbers shown there). You can either type the tab stops on the RECONFIGURE OPTIONS screen or you can use the SETPROFILE command discussed next.

To set the tab stops on the RECONFIGURE OPTIONS screen, press RECONFIGURE OPTIONS and type the column numbers in the Tab Stops field. Press \*C to return to the file.

---

#### **Highlight** Using the SETPROFILE Command to Set Tab Stops

You can save time by using the SETPROFILE command to change the settings in the profile from the \*C response line instead of making the changes on the RECONFIGURE OPTIONS or the RECONFIGURE FUNCTION-KEYS screen.

For example, to set the tab stops for the column headings needed in the table, follow these steps:

1. Press \*C.
2. When the response line appears, type the information shown in the following example.

Separate the option and the settings with a comma.

Command: SETPROFILE TAB STOPS, 30 42 54 66 79 91 106 118 █

You can abbreviate the command to SET.

Type the option here.

Type the settings here.

Next is an example of using SETPROFILE to change the assignment for a function key. For this example, let TEDIT prompt you for the information. Follow these steps:

1. Press \*C, type SETPROFILE on the \*C response line, and press \*C again.

Command: SETPROFILE █

2. TEDIT displays the SETPROFILE response lines. Type the information shown in the following example to change the setting for the **F2** function key from its previous setting (which was EOL) to the new setting of REPEAT. Then press \*C to issue the SETPROFILE command.

Item: <F2>  
Change: REPEAT █

Put the function key number inside angle brackets here.

Type the new setting (the change) here.

You can use the REPEAT command, for example, to repeat a string of commands you gave on the \*C response line. Using the SETPROFILE command is an easy way to change the function key setting to REPEAT and then change it back when you do not need it anymore.

## Creating an Oversized Table

Using the SETPROFILE Command to Set Tab Stops

### Using TAB and BACKTAB for Horizontal Scrolling

You now have only the two main headings left to put on the table. Use the RULER to determine the midpoint in each of the areas where you want the headings for "DEPARTMENT" and "QUARTERLY EXPENSES." Use the TAB and BACKTAB commands to move horizontally in the file and to scroll the text. Center the headings in their areas as shown in the following example.

The diagram shows a table with two main columns: DEPARTMENT and QUARTERLY EXPENSES. The DEPARTMENT column contains 'Administration (Yuen)' and 'Design and layout (Medici)'. The QUARTERLY EXPENSES column contains 'SALARIES', 'EQUIPMENT', 'SUPPLIES', and 'FREELANCERS'. Below these are the labels 'BUDGET' and 'ACTUAL'. Annotations indicate that 'DEPARTMENT' is centered in the space between columns 1 and 19, and 'QUARTERLY EXPENSES' is centered in the space between columns 20 and 118. A ruler at the top of the table shows column positions from 10 to 70.

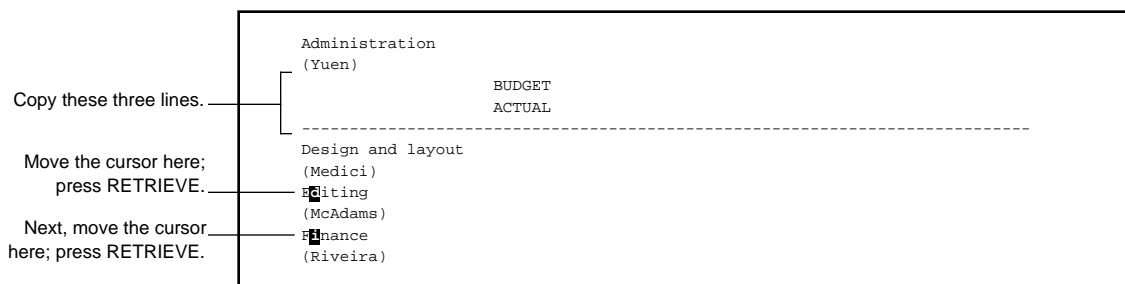
DEPARTMENT	QUARTERLY EXPENSES
Administration (Yuen)	SALARIES
Design and layout (Medici)	EQUIPMENT
	SUPPLIES
	FREELANCERS
	BUDGET
	ACTUAL

### Putting in Horizontal and Vertical Rules

You can add row headings and put horizontal rules in the file easily by using the COPY and RETRIEVE commands. You type the headings and the horizontal rule only once and then simply copy it to each new location. To easily put vertical rules in the table, you use the REPLACE command. Both of these procedures are discussed next.

**Horizontal Rules.** The first column to the right of the department names contains the row headings “BUDGET” and “ACTUAL.” Below the row headings you want a horizontal rule. So, to put in one rule and to repeat the headings and the rule once for each department, follow these steps:

1. Put the cursor on the line containing “Design and layout.” Press LINE. Then press INSERT. On the blank line you inserted, type a line of hyphens all the way across the file. Just press the - (hyphen) key and hold it down. Use the LEFTSCROLL command or the TEDIT TAB key to shift the text over when you get to the right edge of the screen. Then continue inserting hyphens until you get to column 118.
2. Using RIGHTSCROLL or the TEDIT BACKTAB key, move the cursor to the line containing “BUDGET.” Press the LINE key. With LINE persisting on the status line, press COPY. The cursor moves down one line. Press COPY again to copy the line containing the word “ACTUAL.” When the cursor moves down to the line containing the horizontal rule, press COPY once more. Now you have copied all three lines to the scratchpad.
3. The next step is to retrieve the lines. Move the cursor to the line following each grouping of department and department head, then press RETRIEVE.



## Creating an Oversized Table

Using the SETPROFILE Command to Set Tab Stops

4. Continue to work your way down the list until you have copied “BUDGET,” “ACTUAL,” and the horizontal rule into position for each department. The file should look like the next example (only part of the file is shown). When you get to the bottom of the screen, you can either press `NEXT PAGE` to scroll the text up, or you can press the `ROLL UP` key to scroll the text up one line at a time. Or, you can press `SHIFT` and `ROLL UP` to scroll the screen up eight lines at a time.

**Note** You can adjust the number of lines `SHIFT ROLL UP` and `SHIFT ROLL DOWN` scroll the text by changing the repeat count number on the RECONFIGURE FUNCTION-KEYS screen.

```
Administration
(Yuen)
                BUDGET
                ACTUAL
-----
Design and layout
(Medici)
                BUDGET
                ACTUAL
-----
Editing
(McAdams)
                BUDGET
                ACTUAL
-----
Finance
(Riveira)
```

**Vertical Rules.** It is easy to use the REPLACE command to insert vertical rules into the table you have set up. You can use the REPLACE response lines to type the command. Because you will probably use vertical rules regularly when you create tables, however, you may want to assign the command to a function key in your TABLE profile so you always have it available. Both methods are shown here.

Follow these steps to use the REPLACE response lines to insert some vertical rules into the table:

1. Before you start, make sure that auto word wrap is set to OFF. You can check by pressing the RECONFIGURE OPTIONS key.
2. Position the cursor in column 19 of the line containing the column headings. Column 19 is where you want to insert the first vertical rule.
3. Press the REPLACE key. When the response lines appear, type the information you see in the following example exactly as shown.

Replaces a null character (" ") with a vertical bar (|).

\*/L inserts rules throughout the rest of the file. To insert rules one screen at a time use \*/B.

```

Replace: ""
With: "|"
In lines: */L
In columns: *: *
Options: |
    
```

The column the cursor is in

4. Press REPLACE again. TEDIT inserts a vertical bar in the column where the cursor is (column 19) from the line the cursor is in through the last line in the file. The file should now look like the following example; only part of the file is shown.

With the cursor here, press REPLACE.

	SALARIES	EQUIPMENT	SUPPLIES
Administration (Yuen)	BUDGET		
	ACTUAL		
Design and layout (Medici)	BUDGET		
	ACTUAL		
Editing (McAdams)	BUDGET		
	ACTUAL		
Finance (Riveira)			

## Creating an Oversized Table

### Inserting Vertical Rules With a Function Key

5. To put vertical rules in the rest of the file, simply move the cursor to the next column where you want to insert a rule. Press REPLACE twice. Do this for each rule in the table. When you get to the right edge of the screen, use LEFTSCROLL or TAB to scroll the text so you can put in the rest of the rules.
6. Because the REPLACE command changed a null character to a "|", the horizontal lines of hyphens moved to the right each time you inserted a vertical rule. Go to the right edge of the table now and delete the extra hyphens so that the rules line up with the end of the TOTALS column. Your table should look like Figure 5-3 when you have finished.

---

### Highlight Inserting Vertical Rules With a Function Key

To use a function key to give the command to insert a vertical rule, follow these steps:

1. Press \*C. Type RECONFIGURE FUNCTION-KEYS (or REC F). Press \*C again.
2. Pick a key you use infrequently. Type the command shown in the next example.

Leave no blank spaces between  
these quotation marks.

This command inserts  
"|" in every line from  
the cursor to the  
bottom of the screen.

```
F8: REPLACE "", "|", */B, *:*
```



3. To insert the rules in the file, simply put the cursor in the column where you want the rule and press the function key.

---

**Note** Use TEDIT function keys to edit text that is wider than the screen; do not use the terminal function keys. The terminal function keys, such as `CHAR DEL` and `CHAR INS`, affect only the characters that are in view on the screen.

When you use `CHAR DEL`, for example, the characters to the right of the cursor position that are currently on the screen move to the left to close up the space taken by the deleted characters. But the characters not in view on the screen do not move. Thus, you create a gap in the line at the right edge of the screen.

---

When you finish, the table should look like Figure 5-3, printed broadside on 11-inch by 8-1/2-inch paper. For information about printing, refer to Part Two, "Using PS TEXT FORMAT."

## Creating an Oversized Table

Inserting Vertical Rules With a Function Key

Figure 5-3. The Completed Table—an Expense Form

DEPARTMENT	QUARTERLY EXPENSES						TOTALS
	SALARIES	EQUIPMENT	SUPPLIES	FREELANCERS	TRAVEL	MISCELLANEOUS	
Administration (Yuen)							
	BUDGET						
	ACTUAL						
Design and layout (Medici)							
	BUDGET						
	ACTUAL						
Editing (McKdams)							
	BUDGET						
	ACTUAL						
Finance (Riveira)							
	BUDGET						
	ACTUAL						
Marketing (Goncherov)							
	BUDGET						
	ACTUAL						
Printing and distribution (Kitahara)							
	BUDGET						
	ACTUAL						
Production (Strauss)							
	BUDGET						
	ACTUAL						
Publisher (Monreau)							
	BUDGET						
	ACTUAL						

---

**Creating and Saving  
the Profile**

The table you create in this section requires profile settings that take into account its more than average width and its tab settings. You have already made most of the changes you will want to save in the TABLE profile:

- You put in the tab stops on the RECONFIGURE OPTIONS screen.
- You assigned to two function keys the commands that insert and delete columns.
- You assigned to another function key a command to insert a vertical rule.

You have been using your changed profile without having saved it. Remember, though, the profile will be erased when you exit from TEDIT unless you save it with the SAVEPROFILE command. Before you save it, however, read the following Note.

---

**Note** One feature you frequently use in TEDIT, auto word wrap, is a disadvantage when you are working mainly with lines rather than with paragraphs, especially if the lines exceed your align width. When auto word wrap is ON, TEDIT automatically adjusts the width of any lines you move, copy, insert text items into, or replace to the align width you set in the profile. If you want the text on each line you copy or move to remain exactly the same—that is, to not wrap down to the next line if it is wider than the align width—make sure you set auto word wrap to OFF.

---

Check the options part of the profile to make sure you have the changes shown in the next example.

## Creating an Oversized Table

### Creating and Saving the Profile

Change the remarks to match the profile.

```
Remarks: TABS X 12; INS & DEL COLUMNS; INS "|"; AWW OFF
Align Width: 70
Overlap (0-12): 2
Paragraph Lines: 6
Printer Page Size: 0
Blank Fill on 3270: ON
Tab Stops (2-238): 30 42 54 66 79 91 106 118
Word Stops: " !""#$(*)*+,-./:;<=>?@[\\_{}|}~"
Sentence Stops: "!..?"
Format Stops: "!#%&'()*+,-./:;<=>?@\\^_`{|}~"
SEARCH Options:
REPLACE Options:
OPENWINDOW Options:
```

Set auto word wrap to OFF.

Set the tabs for the column headings.

Check the function-keys part of the profile to make sure you have the key assignments shown in the next example.

```
F6: REPLACE ??, @2, */B, *:*+1, E
F7: REPLACE "", " ", */B, *:*
F8: REPLACE "", "|", */B, *:*
```

Deletes a column.

Inserts a column.

Inserts a vertical rule.

Now you can save the TABLE profile. Use the shortcut shown in the next example. Follow these steps:

1. Press \*C. When the response line appears, type the following information:

```
Command: SAVEPROFILE TABLE
```

2. Press \*C again. TEDIT saves the TABLE profile in your profile file.

**Summary** In this section you learned how to edit a file that is both longer and wider than what you can see on the screen at any given time. To edit a file such as this, you scrolled the text vertically using `NEXT PAGE`, `PREV PAGE`, `ROLL UP`, and `ROLL DOWN`. You scrolled the text horizontally using `LEFTSCROLL`, `RIGHTSCROLL`, `TAB`, `BACKTAB`, and `RETURN`.

You transferred the rough-draft text from another file using the window feature. You defined the specific text you wanted to move using the `DEFINEREGION` and `REGION` commands. You learned two new ways to create files and open windows.

You made some repetitive editing tasks easy to do by reassigning some of your function keys, or by using `*C` or an `OBEY` file to give a series of commands in one step. You learned three ways to reissue a command without retyping: using a repeat count, the `FC` command, and the `REPEAT` command. You used `COPY` and `RETRIEVE` and the `REPLACE` command to do other repetitive tasks. The repetitive tasks you had `TEDIT` do for you are:

- Deleting a column from every line that appears on the screen
- Inserting a blank column in every line that appears on the screen
- Making the first letter of every other line uppercase
- Putting parentheses around a word in every other line
- Inserting repeating heads and horizontal rules
- Inserting vertical rules

Finally, you saved the special function-key assignments and tab stops you used for the table in a profile called `TABLE` so you can use them the next time you create a table. You used both the `RECONFIGURE OPTIONS` and the `SETPROFILE` commands to change your profile settings.

---

# 6 Working With Larger Files

---

**Overview** In Section 6 you put the form letter, the agenda, and the expense form (table) into a single file. You combine these separate elements into a package as a simulation of how you would prepare them for distribution. Section 6 gives you practice at moving around in and editing a larger file.

The features you learn in this section are:

- Copying text to and from other files using the READ and WRITE commands
- Finding specific text in a large file by marking sections and positions or by using line numbers
- Scrolling text by lines and pages
- Setting the profile from the \*C response line
- Moving large blocks of text using the SECTION, WINDOW, and REGION text items

**Combining Files** There are several ways you can combine files. You can use the window feature you learned in previous sections, or you can use the READ and WRITE commands. You use READ to transfer text from another file *into* the one you currently have open. You use WRITE to transfer text into another file *from* the one you currently have open. READ and WRITE transfer a copy of the text you specify, they do not remove it from its original file.

Start by using READ to combine the three files you created in Sections 3, 4, and 5 into one file. Follow these steps:

1. Create a file named DOCUMENT. Type the following command at the command interpreter.

```
6> TEDIT DOCUMENT! █
```

2. After TEDIT creates the file, press \*C. When you see the response line, type the command shown in the following example.

```
Command: READ █
```

3. TEDIT displays the following response lines. Type the information shown in the example.

Name the file you want to transfer. —

The default range of ALL copies all the lines in the file. —

```
Read from: TABLE
Range: ALL █
```

4. Press \*C again. TEDIT copies all the lines from the TABLE file into the scratchpad of DOCUMENT, the current file.
5. Position the cursor where you want the text from the TABLE file to begin—in this case, leave the cursor where it is, in the HOME position.
6. The next step is to retrieve the text. First, go to the RECONFIGURE OPTIONS screen and make sure auto word wrap is set to OFF. Or, you can disable auto word wrap and retrieve the text in one step; see the following example.

### Highlight Turning Off Auto Word Wrap When Using RETRIEVE

Before you press RETRIEVE, you must make sure that the auto word wrap function is OFF. Otherwise, when you retrieve the text, TEDIT adjusts the lines to the align width set on the RECONFIGURE OPTIONS screen; this makes the table quite unreadable.

To avoid this problem, assign to the RETRIEVE key three commands that set auto word wrap OFF, retrieve text, and then set auto word wrap back ON. Use the series of commands shown in the next example.

SET is the abbreviation  
for SETPROFILE.

You can abbreviate  
AUTO WORD WRAP  
to AUTO W.

```
F10: SET AUTO WORD WRAP,OFF; RETRIEVE; SET AUTO WORD WRAP,ON ■
```

7. Press RETRIEVE. TEDIT retrieves the text into the DOCUMENT file at the cursor position.



## Working With Larger Files

### Turning Off Auto Word Wrap When Using RETRIEVE

---

To transfer the next file, AGENDA, into the DOCUMENT file, follow the same steps as before:

1. Press \*C.
2. This time give all the information on the \*C response line that you gave on the READ response lines in the previous example. Using \*C you can bypass the secondary READ response lines. In fact, you can use \*C to bypass most of the secondary response lines in TEDIT. Type the information shown in the following example to read all the text from the AGENDA file into the DOCUMENT file.

```
Command: READ AGENDA, ALL █
```

3. Press \*C again. The AGENDA file is now in the DOCUMENT file's scratchpad.
4. Go to the last line of the table. To locate the last line, scroll the text up. You can scroll through the file one page at a time; every time you press `NEXT PAGE`, TEDIT scrolls the text up by one page. You can scroll ahead to the end of the file with `SHIFT NEXT PAGE`, which takes you to the last page of the file. (To return to the first page of the file, press `SHIFT PREV PAGE`.)
5. After you find the last line, press `RETURN` twice to leave a blank line between the table and the agenda.
6. Press RETRIEVE. TEDIT retrieves the AGENDA file from the scratchpad at the cursor position.

Use the same steps to transfer the LETTER file into the DOCUMENT file, putting the letter at the end of the file. Now you should have copies of all three files in the DOCUMENT file.

---

## Finding and Moving Large Blocks of Text

To see where the text is situated, page through the file using the `NEXT PAGE` and `PREV PAGE` keys. You want to rearrange the three elements in this file so that the letter comes first, the agenda next, and the table last.

The first step to rearranging text in a large file is to locate the text. There are several easy ways to locate text in a large file. Perhaps the easiest is to search for a unique word or phrase, using the `SEARCH` command. You have already learned how to use the `SEARCH` command. This section discusses three other methods of getting around in a large file:

- Marking certain sections and moving to the sections with the `FORWARD` and `BACKWARD` commands
- Searching for a specific line number and column number
- Marking specific positions with your own numbering system and moving to the positions with the `GOTOPOSITION` command

## Using Section Markers

You can divide the text into sections and label each section with a special marker. After you have the sections labeled, you can move around by using the `FORWARD` and `BACKWARD` commands with the text item `SECTION`. You can also use other text commands, such as `DELETE` or `MOVE`, with the text item `SECTION`; moving sections is one way to rearrange large blocks of text.

### Determining the Section Marker

Before you mark the sections, tell `TEDIT` what the label is that you intend to use as a marker. You do this on the `RECONFIGURE OPTIONS` screen.

1. Press `RECONFIGURE OPTIONS`.
2. When the screen appears, press `NEXT PAGE`. The first field on the second page of the `RECONFIGURE OPTIONS` screens is for the section marker.



- Again insert two blank lines. These lines are used to separate the sections visually; they are not necessary for the commands using SECTION to function. Type three asterisks on the topmost blank line. The file should look like the next example.

Type \*\*\* between the table and the agenda.

```
-----|-----|-----|-----|-----|-----|
***
AGENDA
```

- Use SEARCH to find the beginning of the letter. Press SEARCH and fill in the first field with the date that is on the letter, March 7. Press \*C. TEDIT displays the beginning of the letter.
- Again insert two blank lines. Type three asterisks on the topmost blank line. The file should look like the next example.

Type \*\*\* between the agenda and the letter.

```
4:30      Approval of budget
***
March 7
```

### Moving a Section

Now, use the SECTION text item with the FORWARD command or BACKWARD command to move the cursor to any section. First find the table section, then move it to follow the letter. Follow these steps:

- Press \*C. When the response line appears, type the word SECTION. You do this to make the text item SECTION persistent. Press \*C again. Notice that SECTION is now on the status line.
- Press BACKWARD. The cursor moves to the SECTION marker at the beginning of the previous section. Press BACKWARD twice more to move the cursor to the first section marker. The cursor should be at the beginning of the table now.

3. To move the table section, make sure that SECTION is still persisting on the status line. Press MOVE. TEDIT moves the text that is between the cursor position and the next section marker onto the scratchpad. Notice that the section marker at the end of the section was not moved along with the section.
4. Press FORWARD to move to the end of the letter section. Notice that there is no section marker at the end of the letter, but the end of the letter is also the end of the file. TEDIT sees the end of the file as the end of a section. In fact, TEDIT sees the end of the file as the end of any text item (SECTION as well as SENTENCE, PARAGRAPH, REGION, and so on).
5. Move the cursor down one line. Press RETRIEVE. TEDIT retrieves the table at the cursor position.

You can use this same procedure to move the agenda section. Move it to fall between the letter and the table. First, find the agenda. With SECTION still on the status line, press BACKWARD until you see the agenda. Then press MOVE, reposition the cursor by pressing FORWARD until you see the table, insert a blank line, and press RETRIEVE.

---

**Note** Because the entire agenda fits on one screen, you could move the agenda section using the text item SECTION or the text item WINDOW. The text item WINDOW is not the same thing as the windows you open and close with the OPENWINDOW and CLOSEWINDOW commands. The text item WINDOW is defined as all the text between the cursor position and the bottom of the window the cursor is in (in some cases the bottom of the window is the same as the bottom of the screen). Try using MOVE with both SECTION and WINDOW to see how it works; try one, UNDO it, and try the other.

---

**Using Line and Column Numbers**

You may not have a document that divides nicely into sections the way this one does. In that case, you might want to use line numbers (and with an extra-wide file, column numbers) to find specific text. You probably are aware that TEDIT refers to lines of text by number. There is a unique number associated with each line and each column. You can see the numbers of the first and last lines and columns of text on the screen just by looking at the status line. Notice that the column notation on the status line changes to 1:79+ when there is text beyond the 79 characters that are visible on the screen.

If the cursor is still at the place where you retrieved the agenda, the status line should look like the following example.

Lines 83 through 95.11 are currently in view.	<pre> 1) \$MYVOL.MYSVOL.DOCUMENT 83/95.11 1:79 </pre>
Columns 1 through 79 are currently in view.	<pre> C00. </pre>

If you press `ROLL UP` or `ROLL DOWN`, you can see the line numbers change by one at either end. Rather than scrolling the text so you can determine line numbers from the status line, you can use several TEDIT commands that work with line numbers. These commands are discussed next.

**Displaying All the Line Numbers**

To see the line numbers associated with each line, use the `SHOWNUMBER` command. The `SHOWNUMBER` command has three options: `ON`, `OFF`, and blank. If you give the `SHOWNUMBER` command alone, TEDIT displays on the status line the number of the line the cursor is in. To show all the numbers in the file, press `*C` and type the command on the response line as shown in the following example. You can abbreviate `SHOWNUMBER` to `SH`.

<pre> Command: SHOWNUMBER ON █ </pre>
---------------------------------------

## Working With Larger Files

### Using Line and Column Numbers

TEDIT displays the line numbers along the left edge of the screen. Your screen should look like the following example; only part of the file is shown.

```
90      with name3 and name4 before presenting the figures at
91      the meeting.  Thank you.
92
93      J.
94
95      ***
95.01
95.02  AGENDA
95.03
95.04  8:00  Presentations by Department Heads
95.05
95.06      NAME           DEPARTMENT
95.07      Goncherov      marketing
```

Page through the text to see the numbers change. Look at the place where the table ends and the letter begins. Notice that the places where you inserted lines to put in the section marker has numbers with one and two decimal places. Also notice that the line numbers start at 69.01. The table and the agenda sections of the file had the numbers 1 through 69 before you moved them.

TEDIT does not fully renumber when you insert or delete lines. With the original line numbers left intact (by using decimal numbers or leaving gaps in the numbering), TEDIT keeps track of where lines were inserted and deleted. You can, however, renumber the entire file; see “Renumbering Lines” later in this section.

### Displaying Individual Lines by Number

Try moving around in the file using the line numbers. First, go to the beginning of the file. You now know several methods of getting to the beginning of the file, but here is a method that finds text by line number. Follow these steps:

1. Press \*C. When the response line appears, type DISPLAYLINE 0 (or simply DI 0). See the example.

```
Command: DISPLAYLINE 0 █
```

2. Press \*C again. TEDIT scrolls the text to the beginning of the file and places the cursor on the first line. TEDIT moves to the beginning of the file because you asked for a line number that is smaller than the first line in the file. Notice the status line gives the first line number as 69.01 and the designation BOF. BOF stands for beginning of file. TEDIT designates the end of the file with EOF.

If you type just DISPLAYLINE without a number after it, TEDIT displays a secondary set of response lines. The fields ask you to type a line number and a column number. You could fill in only the Display-line field, but if you give the column number as well, TEDIT positions the cursor on the line and in the column you specify.

### Displaying Columns by Number

Moving directly to a particular column is useful if you want, for example, to change a number in a part of a table (such as the expense form) that is not currently in view on the screen. Using the DISPLAYLINE command to find the column could save you from having to use the LEFTSCROLL command and then move to the correct place with the arrow keys.



## Working With Larger Files

### Using Line and Column Numbers

Follow these steps to locate a specific line and column in the right-hand portion of the table, which is not currently in view:

1. Page forward in your file until the table is in view. Put the cursor in one of rows containing the BUDGET or ACTUAL heading. Knowing where you set the tabs for the table (in Section 5), pick the tab stop for the MISCELLANEOUS column—the number is 91.
2. Press \*C. When the response line appears, type DISPLAYLINE. Press \*C again.
3. When the DISPLAYLINE response lines appear, fill them in as shown in the following example.

Type \* to display the line the cursor is in (or you can type a line number).

```
Display line: *          column: 91█
```

4. TEDIT positions the cursor in column 91 of the line containing the cursor.

To go back to column 1, you can give the DISPLAYLINE command again, specifying column 1. To give this same command entirely from the \*C response line, press \*C. When the response line appears, type the following command:

Separate the line number and the column number with a comma.

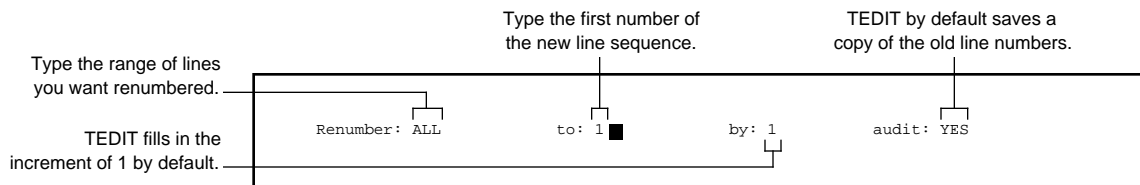
```
Command: DISPLAYLINE *,1█
```

### Renumbering Lines

If you use the line numbers a lot, you probably want to renumber them from time to time so that 1) you do not have too many decimal numbers and 2) the lines are all numbered sequentially with no gaps or decimals in the numbering. To renumber the lines, follow these steps:

1. Press \*C. When the response line appears, type RENUMBER (or REN). Press \*C again.

- When the RENUMBER response lines appear, type the information you see in the following example. Remember to use the terminal `TAB` key or the arrow keys to move the cursor.

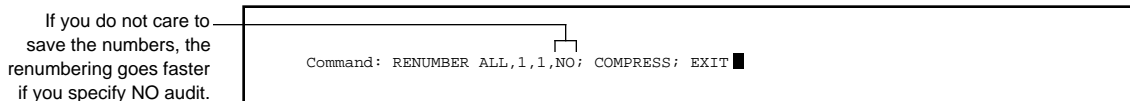


If you want, you can renumber only part of the file by giving a specific range of lines in the first field. You can start renumbering at any number, say at 25 or at 300. You can change the increment; for example, to renumber by fives or tens. If you audit the renumbering (see the Audit field in the example), you can use UNDO to put the numbers back the way they were before you gave the command.

- Press `*C` again. TEDIT renumbers the file so that all the lines are numbered sequentially by ones starting at 1.

### Highlight Renumbering and Compressing Files for Storage

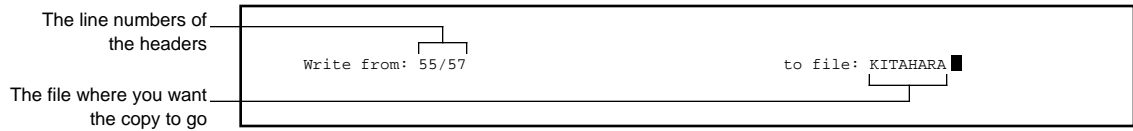
Renumbering is one way to keep larger files organized. You might want to make a habit of renumbering every time you exit from a file. Another good habit to form is to compress your files occasionally to reduce the storage (disk) space that the files occupy. The following example shows a series of commands that you can give on the `*C` response line or assign to the function key for the EXIT command; these commands renumber and compress the file every time you exit.



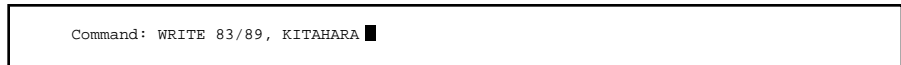
**Copying a Range of Lines to Another File**

You can copy part of the file you are in to another file without closing your current file or opening another window. For example, say you want to copy the figures for printing and distribution from the table to another file. You can copy the lines containing the table headers and the printing and distribution figures by line number. Follow these steps:

1. Make sure the headers for the table are displayed on the screen. The table should be in the range of lines 53 to 99 now that you have renumbered the file. You should still have the line numbers displayed along the left edge of the screen. If you do not, give the SHOWNUMBER ON command again. Note the line numbers of the header.
2. Press \*C. When the response line appears, type WRITE and press \*C. TEDIT displays the response lines shown in the following example. Using the line numbers displayed on the screen, fill in the fields as shown.



3. Press \*C again. TEDIT copies lines 55 through 57 to a new file called KITAHARA.
4. Scroll the text up until the lines that contain the information on printing and distribution are visible. Note the line numbers.
5. This time give the command in one step. Press \*C. When the response line appears, type the following command:



6. Press \*C again. TEDIT copies lines 83 through 89 to the end of the KITAHARA file.

If you want to see what you have copied, press \*C. When the response line appears, type the following command:

```
Command: OPENWINDOW 2, KITAHARA █
```

Press \*C again. TEDIT opens a window into the KITAHARA file; it should look like the following example.

DEPARTMENT	QUARTERLY EXPENSES				
	SALARIES	EQUIPMENT	SUPPLIES	FREELANCERS	
Printing and distribution (Kitahara)					
	BUDGET				
	ACTUAL				

To close the KITAHARA file, type the following on the \*C response line:

```
Command: CLOSEWINDOW 2 █
```

Now that you are finished with the section on line numbers you can remove the numbers from the screen by typing the following command on the \*C response line:

```
Command: SHOWNUMBER OFF █
```

**Using Marked Positions** You can mark specific positions in the file, numbering the positions in any order you like. As you personalize the form letters, you need to refer to the agenda several times in order to fill in the blanks in the letter. To scroll quickly from the letter to the agenda and back again, you mark positions in both the letter and the agenda. Then you can use the marks to scroll the text. Follow these steps to mark and display specific positions:

1. Use `SHIFT` `PREV PAGE` to scroll to the beginning of the file and display the letter. Position the cursor in the line that contains two names that change for each personalized letter.

Put the cursor on a line that has names that change.

```
As you know, the quarterly budgeting meeting has been
scheduled for March 15. Your presentation will follow
■veira and precede Yuen. See the attached agenda.
```

2. Press `*C`. When the response line appears, type `MARKPOSITION 1` (or `MAR 1`).

```
Command: MARKPOSITION 1 ■
```

3. Press `*C` again. The cursor position is now marked as position 1.
4. Use `NEXT PAGE` to scroll the text so that the agenda is visible. Position the cursor at the second position you want to mark; the 10:00 break in the middle of the agenda is a good place. When you use the command to go to the position, `TEDIT` positions the text you marked in the middle of the screen; thus, you can see all the names on the agenda when you go to this middle position.

Put the cursor on a line in the middle of the agenda.

```
Medici                design and layout
■0:00                Break
Monreau                publisher
```

5. Press \*C. When the response line appears, type MARKPOSITION 2.

```
Command: MARKPOSITION 2 █
```

6. Press \*C again. The position is now marked.

To return to the letter to mark the third position, use the GOTOPOSITION command to go back to the letter. Follow these steps:

1. Press \*C.
2. Type GOTOPOSITION (or you can abbreviate it to GO) followed by the position number you marked for the line. See the next example.

```
Command: GOTOPOSITION 1 █
```

3. Press \*C again. TEDIT places the cursor exactly on the position you marked.
4. Now scroll the text down until you see the line containing the other set of names that change with each personalized letter. Put the cursor on the line.

```
      In addition, Sydney, the publisher would like you to  
      discuss your projections for the production department with  
      Medici and Kitahara before presenting them at the meeting.  
      Thank you.
```

5. Mark the line as position 3 by pressing \*C and typing MARKPOSITION 3. Press \*C again. The position is now marked.

```
Command: MARKPOSITION 3 █
```

6. Now that you have the positions marked, you can move quickly between positions with the GOTOPOSITION command. Simply type GOTOPOSITION followed by the position number on the \*C response line each time you want to move.

If you forget the number you assigned to a position or what line the position you marked is in, you can type the command GOTOPOSITION by itself (not followed by a number) on the \*C response line. See the example.

```
Command: GOTOPOSITION █
```

TEDIT displays a list of all the positions you have marked. The list includes the line number and some text, which helps you recognize the place you marked. See the next example.

```
Position Number: █  
1 11 Riveira and precede Yuen. See the attached agenda.  
2 40 10:00 Break  
3 23 Medici and Kitahara before presenting them at the meeting.
```

Another way you can use MARKPOSITION and GOTOPOSITION is to locate either end of a large block of text that you want to delete, move, or manipulate in some way. You can define the text between the two markers as a REGION; then you can use text commands on that region.

You can remove any of the marked positions from the list with the command UNMARKPOSITION followed by a position number, such as UNMARKPOSITION 3 (or UNM 3).

```
Command: UNMARKPOSITION 3 █
```

Note that TEDIT erases all the position markers when you exit from the file. You can, however, save the positions by using the WRITE command to write them to another file. To save the marked positions in a file named DOCPOSNS, type:

```
Command: WRITE POSITIONS, DOCPOSNS █
```

To read the positions back into your file, use this command:

```
Command: READ DOCPOSNS, POSITIONS █
```



---

**Summary** In this section you combined three files, which you created in previous sections. You learned how to use the READ command to copy text from another file to your current file's scratchpad without having to open another window.

When you combined the files, you had to learn new ways to find your way around in a larger file. You moved around by scrolling the text, either by lines or by pages. You jumped from the beginning of the file to the end and back again using `SHIFT` `NEXT PAGE` and `SHIFT` `PREV PAGE`. You also moved around by locating specific text: You marked sections and positions, then used the FORWARD and BACKWARD commands to find the sections or the GOTOPosition command to find the positions.

Another way you located text was by asking for specific line numbers. You already knew how to display column numbers with RULER; in this section you learned to display line numbers with SHOWNUMBER. Then you positioned the cursor in a specific line and column by giving the DISPLAYLINE command followed by a line number and a column number.

You learned to copy or move large blocks of text as defined by the text items REGION, SECTION, and WINDOW. You also used WRITE to copy blocks of text to another file by specifying a range of line numbers.

Now you have all the elements of the package each department head needs to prepare for the quarterly budgeting meeting:

- The personalized letter that explains the meeting
- The agenda for the meeting
- The tabular expense form they must fill out before they come to the meeting

You can personalize each of the packages and deliver them by way of electronic mail or by printing them. Those procedures are not covered in the TEDIT part of this manual. For information about printing documents, refer to Part Two, "Using PS TEXT FORMAT." For information about sending documents through electronic mail, refer to the manuals for the PS MAIL products.

---

# 7 Introduction to TFORM

---

**Overview** This section introduces you to PS TEXT FORMAT (or TFORM). TFORM is a program you use to produce printed versions of *documents* stored on your computer. This section covers the following topics:

- What is a formatter?
- How you use TFORM capabilities
- Entering TFORM commands
- Getting help

---

**What Is a Formatter?** TFORM is a *text formatter*. A *formatter* controls the printed appearance of documents. Formatters work hand in hand with text editors.

You use an *editor* to enter and edit text. Editors display text on your screen exactly as you enter it from the keyboard. If you use a line editor, each line of text is identified by sequential line numbers. If you use a screen editor, text appears in sentences and paragraphs. The documents you create using an editor program are stored in *files*.

In your *edit file*, text often appears uneven or disjointed. For example, when you use the `RETURN` or `ENTER` key to move to the next line in an edit file, your lines are often unequal in length.

Or, after you've moved or copied text, your edit file can be left with gaps in lines and paragraphs. As long as you can read the text, it doesn't really matter how it appears when you're working in the edit file.

When you print out the edit file, you want a document with uniform line lengths and consistent paragraphs. You might also want to create a specific *page layout* by adjusting the spacing or by including other elements such as headers and footers. You use TFORM to determine how your printed document looks.

You enter formatting commands directly into your edit file. For example, to create uniform paragraphs, you enter TFORM commands that control line length and justification. To add headers or footers, you enter commands that place text in specific locations along the bottom or top of each page. TFORM formatting commands do not alter the appearance of your document on the screen or in the edit file; they take effect only when the document is formatted and sent to the printer or to a storage file.

Figure 7-1 shows text and TFORM commands in an edit file. Figure 7-2 shows how the text appears in the printed document.

---

**Note** Most of the figures in the TFORM part of this manual use two-step illustrations to explain how TFORM works. The edit file appears first, containing the text and formatting commands. This file representation is followed by an example of the printed document.

---

Figure 7-1. Text and TFORM Commands in an Edit File

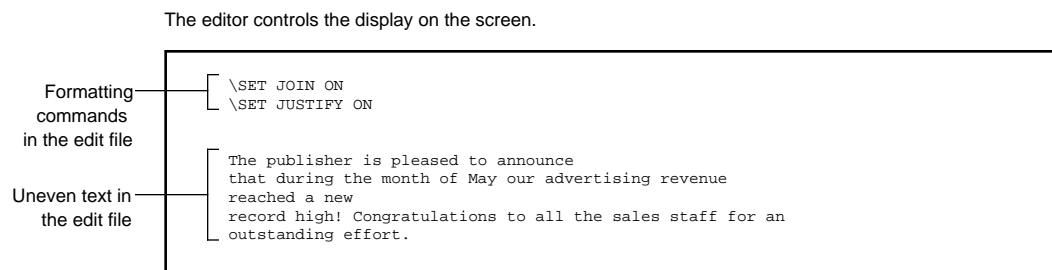
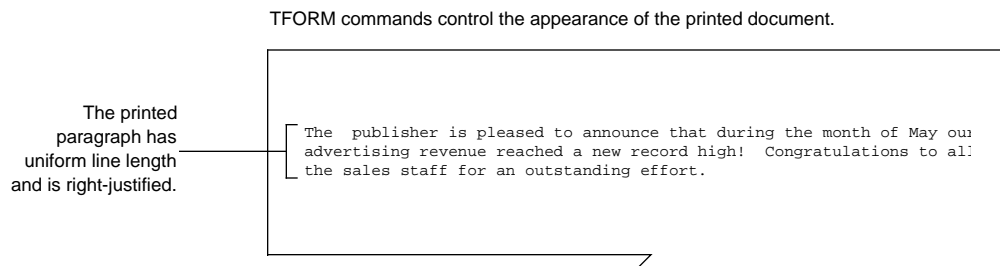


Figure 7-2. Formatted Text in a Printed Document



---

**What TFORM Can Do for You**

Using TFORM, you can format the contents of an edit file in the following ways:

- Determine the length and width of a printed page
- Set left, right, top, and bottom margins
- Indicate spacing between lines, sentences, and paragraphs
- Number pages
- Insert or override page breaks
- Align, justify, and indent text

TFORM also provides commands that let you style text and add other special effects. This manual describes only some of these features. In the following sections you'll find information on how to perform the following tasks:

- Add headers and footers to the document
- Produce boldface and underlined text
- Make bulleted or numbered lists
- Enclose text or figures in boxes
- Separate sections of text using level headings
- Generate a table of contents
- Print form letters

See the *PS TEXT FORMAT Reference Manual* for complete information on all of the special capabilities of TFORM.

**Entering TFORM Commands** You tell TFORM how to format a document by entering commands into an edit file. You can type in the commands as you enter text, or you can go back and insert TFORM commands just before you print out the document.

*Commands* are instructions to TFORM, presented in a language the program understands. Like any language, TFORM commands have some rules you must follow to be understood. These rules are sometimes called the syntax of a command.

**The Backslash** All TFORM commands begin with a backslash. In most cases, the backslash must appear in the first column of a command line. You can have spaces after the backslash if you want:

```
\SET JOIN ON
\  SET JUSTIFY ON
```

---

**Note** There are a few styling commands, such as BOLD and UNDERLINE, that can appear anywhere in the text. These commands are explained in Section 11, "Using Special TFORM Features."

---

**The Command Name** The command name follows the backslash. Command names are one or two words that describe the action you want to take or the printed effect you want to see. For example:

```
\STYLE MARGIN controls margin setting on a page.
\SET JUSTIFY controls paragraph justification.
\NEW begins a new page.
\CENTER centers the text on the line.
```

In the examples in this manual, commands always appear in uppercase. When you're entering command names, however, you can use either uppercase or lowercase.

**Qualifiers and Options** Many TFORM commands are followed by additional information that further qualifies the action of the command. For example, you might need to tell TFORM to turn the command on and off, or to indicate how many lines are affected by the command. This information is often expressed in *values*, or in *units of measurement*. For example:

- `\STYLE MARGIN TOP 1 INCH` creates a top margin of one inch.
- `\SET SPACING 2` produces double-spaced text.
- `\CENTER ON` begins centering text.
- `\CENTER OFF` ends centering of text.
- `\INDENT 3 LEFT 5` indents three lines five characters from the left margin.

The examples in this manual clearly indicate when you must enter a qualifier, value, or unit of measurement to complete a command. In most cases, TFORM assumes a unit of measurement to be in inches, lines, or characters (whichever is appropriate to the action you're taking). You can enter units of measurement in other forms. Table 7-1 lists the units you can use in TFORM and their conversion values.

---

**Table 7-1. TFORM Units of Measurement**

Unit	Conversion Values
DECIPOINT	1/720 inch
POINT	10 decipoints; 1/72 inch
PICA	12 points; 1/6 inch
MM (millimeter)	28 decipoints; approx. 1/25 inch
CM (centimeter)	283 decipoints; approx. 2/5 inch
INCH	720 decipoints; 72 points; 6 pics; 25.4 mm; 2.54 cm
CHAR	Width of a character at the current pitch setting
LINE	Height of a line at the current vertical pitch

---



**Command Shortcuts** TFORM offers two shortcuts to entering commands. You can:

- Use abbreviations for command names
- Enter more than one command on a single line

To abbreviate commands and qualifiers, enter only the first three letters. For example:

```
\STY MAR BOT 1
\SET SPA 2
\CEN ON
\CEN OFF
\IND 3 LEF 5
```

When you enter multiple TFORM commands on one line, only the first command on the line is preceded by the backslash. Separate all the commands with semicolons:

```
\STYLE MAR TOP 1; STYLE MAR BOT 1; SET SPACING 2;
SET JOIN ON
```

---

**Note** The COMMENT command is an exception to this rule. See "Adding Comments" in Section 11 for more information on this command.

---

**Default Settings** *Defaults* are preset values or conditions that are automatically in effect, unless you override them by turning them off or by entering different values. Some TFORM commands are in effect by default or have *default settings*. If you send a file to the printer that doesn't contain any TFORM commands, it's printed using these defaults. The defaults are summarized in Table 7-2.

---

**Table 7-2. TFORM Default Settings**

Feature	Default Setting
Form length	11 inches
Form width	8.5 inches
Top margin	0.5 inch
Bottom margin	0.5 inch
Left margin	0.75 inch
Right margin	0.75 inch
Horizontal pitch	10 characters per inch
Vertical pitch	6 lines per inch
Interline spacing	Single spacing
Line justification	Left-justified, ragged right
Line joining	Off
Headers	None
Footers	None
Page numbers	None

---

In Section 8 and Section 9 of this manual, default settings appear at the beginning of the topic they control. Before entering a TFORM command, check Table 7-2 or these settings. If the default setting meets your needs, you don't need to enter any TFORM commands for that formatting task.

**Getting Help** You can get *help* on specific TFORM commands (and some other more advanced features) by entering the HELP command. Unlike other TFORM commands, you use HELP interactively. This means you enter the HELP command from the TFORM prompt, not in an edit file. To get help, follow these steps:

1. At the command interpreter prompt, enter TFORM. For example:

```
1> TFORM
```

The TFORM identification and TFORM prompt, a question mark (?), appear on the screen:

```
TFORM - T9605C00 - (15JUL87)
?
```

2. At the TFORM prompt, enter the following:

```
? \HELP
```

The help topic screen appears on your terminal. This lists all the help items. Until you are familiar with TFORM, you'll probably use help only to remind yourself of command syntax.

To get help for a specific command, enter HELP and the name of the command. For example:

```
? \HELP STYLE MARGIN
```

TFORM displays the syntax of the STYLE command. If you're not sure of the command name or its spelling, you can enter the following:

```
? \HELP ALL COMMANDS
```

TFORM then displays a list of all the command names. After locating the command you want, enter the HELP command again with the correct command name.

3. To exit help, enter the following at the TFORM prompt:

```
? \EXIT
```

You can also press **CTRL** and **Y** to exit help; TFORM then returns you to the command interpreter prompt.

See the *PS TEXT FORMAT Reference Manual* for a complete description of the HELP command.

---

**Summary** This section provided an introduction to TFORM. It discussed the following topics:

- What is a formatter?
- TFORM capabilities
- Entering TFORM commands
- Using TFORM online help

---

# 8 Designing the Page Layout

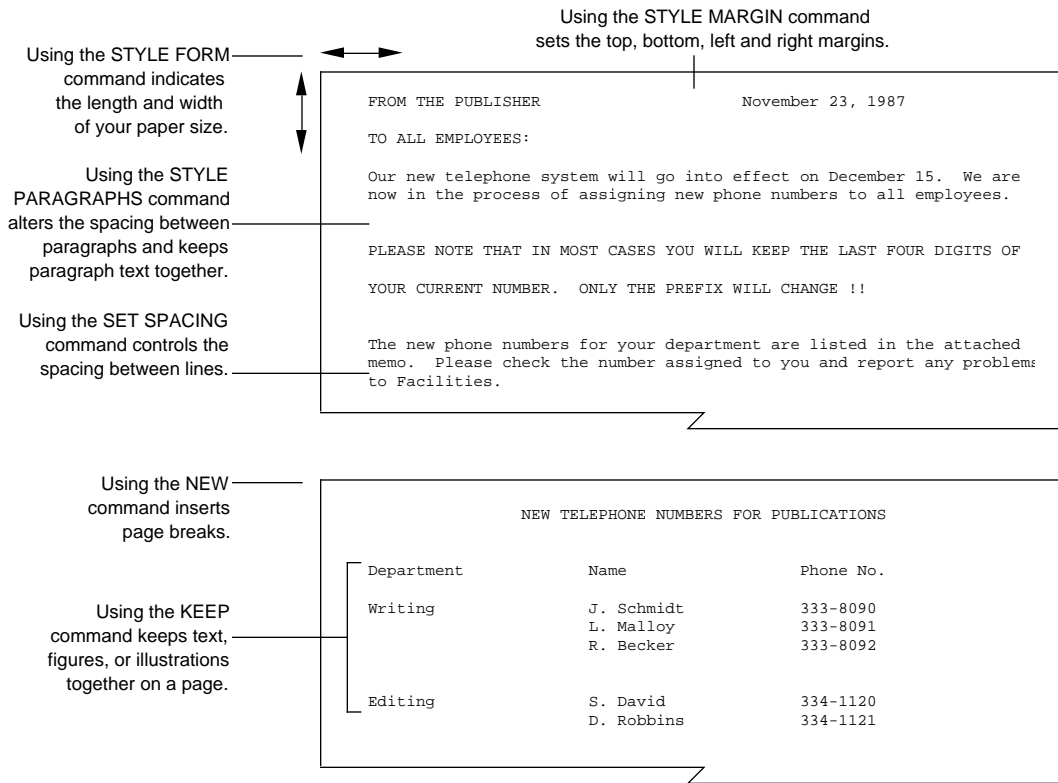
---

**Overview** This section discusses the TFORM commands that control the page layout of a printed document. To design a page layout, you give TFORM the following information:

- The paper size on which you're printing
- The margins to set around the text
- The spacing to put between lines, sentences, and paragraphs
- Where to insert page breaks
- Instructions for keeping lines together on a page

Figure 8-1 shows two pages of a formatted document and lists the TFORM commands used for determining page layout.

Figure 8-1. Using TFORM Commands for Page Layout



---

**Using Page Layout  
Commands**

In many cases, such as printing out memos and simple letters, you might choose to use the page layout default settings. For your convenience, default settings are displayed at the beginning of each topic in this section. If you use the default settings, you don't have to enter any page layout commands. You can move directly to printing your document, as described in Section 12.

---

**Note** If you're planning on using TFORM default settings, keep the line length at 70 when creating your edit file. If your line exceeds 70, TFORM issues an error message for the line and can cut off some of your text.

---

In general, you enter one set of page layout commands to control the format of all the pages in a document. You can also change the page layout for a single page or section of your text.

---

**Note** If your page layout requires STYLE commands (STYLE FORM, STYLE MARGIN, STYLE SENTENCES, or STYLE PARAGRAPHS), these commands must immediately follow a NEW command, even at the beginning of your file. There cannot be any intervening blank lines or text between the NEW and the STYLE commands. This means that any STYLE command setting is in effect for a minimum of one page.

---





Figure 8-2 shows the TFORM page size commands in two separate edit files. The first tells TFORM the form is 3 by 5; the second tells TFORM to change to a 5 by 7 form. Figure 8-3 illustrates the resulting printed documents.

Figure 8-2. Setting the Form Size

A NEW command must precede all STYLE commands.

Sets the form for a 3 by 5 page

```

\NEW
\STYLE FORM WIDTH 3
\STYLE FORM LENGTH 5

MUSIC MONTHLY
12223 Labrea Rd.
Santa Maria
Calif. 88769

CONTACT: Holly
    
```

Sets the form for a 5 by 7 page

```

\NEW
\STYLE FORM WIDTH 5
\STYLE FORM LENGTH 7

ART SUMMARY:

SMALL SYSTEMS SOFTWARE REVIEW 12/88

TYPE            NUMBER    STATUS
PHOTOGRAPHS    12        SCREENED
FIGURES         34        GRAPHICS
ILLUSTRATIONS  18        COMPLETED
    
```

---

Figure 8-3. Printed Documents in Different Page Sizes

MUSIC MONTHLY  
122 Labrea Rd.  
Santa Maria  
Calif. 88769  
  
CONTACT: Holly

ART SUMMARY:  
SMALL SYSTEMS SOFTWARE REVIEW 12/88

TYPE	NUMBER	STATUS
PHOTOGRAPHS	12	SCREENED
FIGURES	34	GRAPHICS
ILLUSTRATIONS	18	COMPLETED



## Designing the Page Layout

---

### Entering Values for Margins

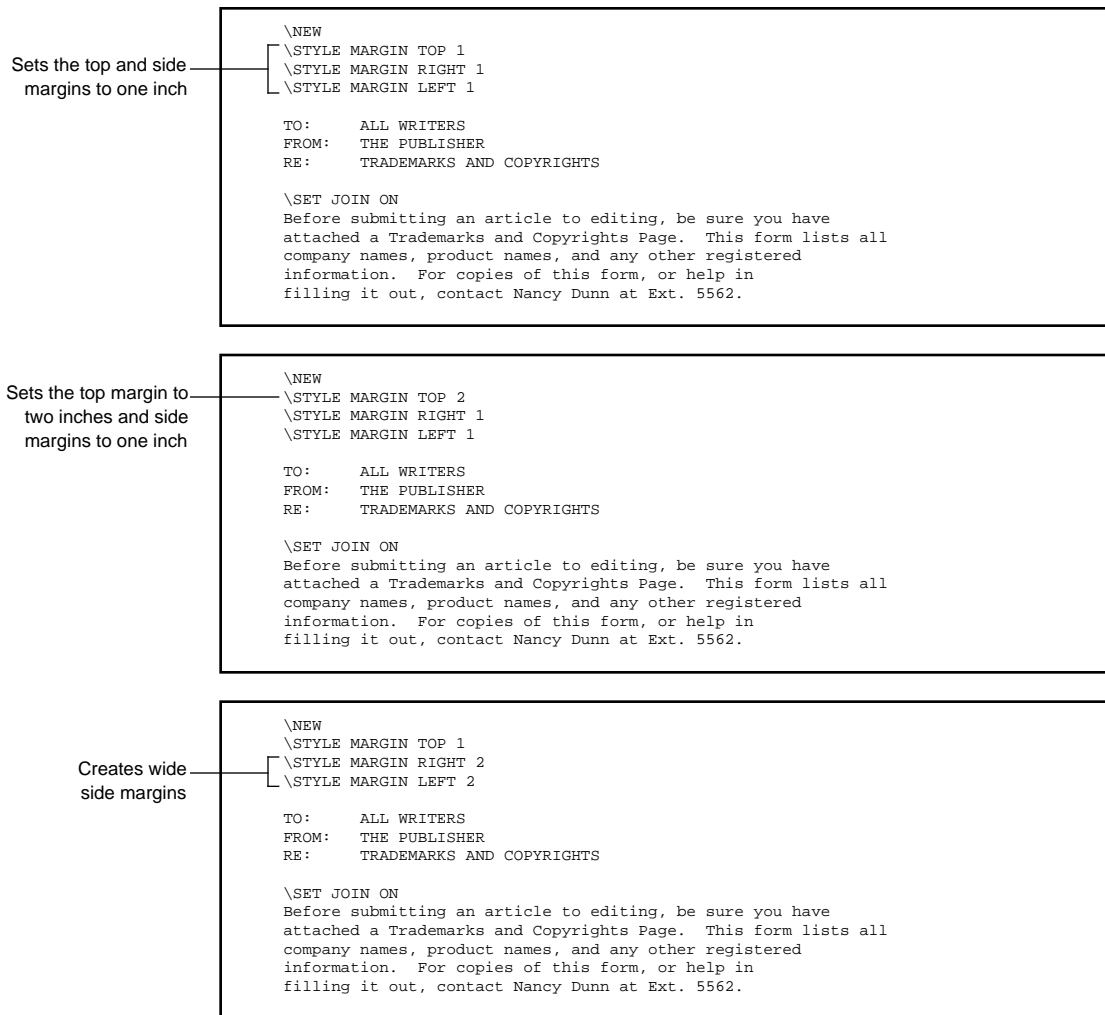
You can, of course, combine commands and use abbreviations. Entering this command line makes all your margins one inch:

```
\STY MAR TOP 1; STY MAR BOT 1; STY MAR RIG 1;  
STY MAR LEF 1
```

If you enter any of the STYLE MARGIN commands without a value, TFORM resets the margins to their default settings.

Figure 8-4 illustrates three edit files using different margin settings. Figure 8-5 shows the resulting three printed documents.

Figure 8-4. Setting Margins



---

Figure 8-5. Printed Documents Using Different Margin Settings

TO: ALL WRITERS  
FROM: THE PUBLISHER  
RE: TRADEMARKS AND COPYRIGHTS

Before submitting an article to editing, be sure you have attached a Trademarks and Copyrights Page. This form lists all company names, product names, and any other registered information. For copies of this form, or help in filling it out, contact Nancy Dunn at Ext. 5562.

TO: ALL WRITERS  
FROM: THE PUBLISHER  
RE: TRADEMARKS AND COPYRIGHTS

Before submitting an article to editing, be sure you have attached a Trademarks and Copyrights Page. This form lists all company names, product names, and any other registered information. For copies of this form, or help in filling it out, contact Nancy Dunn at Ext. 5562.

TO: ALL WRITERS  
FROM: THE PUBLISHER  
RE: TRADEMARKS AND COPYRIGHTS

Before submitting an article to editing, be sure you have attached a Trademarks and Copyrights Page. This form lists all company names, product names, and any other registered information. For copies of this form, or help in filling it out, contact Nancy Dunn at Ext. 5562.

---

**Using Relative Margin Settings** You can change margins from their current settings without entering the actual number of inches for the new margin setting. Instead, you can indicate how much space you want to add or subtract from the current margin settings. When you add or subtract values from the current setting, you are using *relative values*. For example, suppose you want to leave room at the top of specific pages to accommodate a large logo. You enter this command at the top of the first page on which you want to increase the top margin:

```
\STYLE MARGIN TOP +2
```

This command increases the top margin an additional two inches; so, if the top margin is one inch, this command increases it to three inches. The command stays in effect until you set a new top margin.

Suppose you want to leave white space in sections of a document to provide room for notes. On the first page of the section, enter this command:

```
\STYLE MARGIN LEFT +1.5
```

The left margin increases by one and one-half inches. This margin size stays in effect until you enter a new `STYLE MARGIN LEFT` command.

To decrease margins using relative values, put a minus sign in front of the number:

```
\STYLE MARGIN BOTTOM -2
```

This command resets the bottom margin, subtracting two inches from the current setting.

Figure 8-6 shows an edit file of three pages using relative margin settings. Figure 8-7 shows the resulting printed document.



## Designing the Page Layout

### Using Relative Margin Settings

Figure 8-6. Using Relative Margin Settings

```
\NEW
\STYLE MARGIN LEFT 1

TO:    ALL WRITERS
FROM:  THE PUBLISHER
RE:    TRADEMARKS AND COPYRIGHTS

\SET JOIN ON
Before submitting an article to editing, be sure you have
attached a Trademarks and Copyrights Page. This form lists all
company names, product names, and any other registered
information. For copies of this form, or help in
filling it out, contact Nancy Dunn at Ext. 5562.
\NEW
\STYLE MARGIN LEFT +1
You must also complete and submit an Art Summary Sheet for
each article. Be sure and include the current status of each
piece of art. Check with Production if you have any questions about
this form or about the artwork for a specific article.
\NEW
\STYLE MARGIN LEFT +.5
Finally, complete an Author Identification Sheet. This information
is essential because it provides the codes for the production
tracking database.

Although this new procedure requires somewhat more time prior to
submission, I'm sure we'll all appreciate the time it saves in
production. Thanks.
```

Sets a one inch left margin

Increases the margin to two inches

The effect is cumulative. After this command, the left margin is 2.5 inches.

Figure 8-7. Printed Document Using Relative Margin Settings

TO: ALL WRITERS  
FROM: THE PUBLISHER  
RE: TRADEMARKS AND COPYRIGHTS

Before submitting an article to editing, be sure you have attached a Trademarks and Copyrights Page. This form lists all company names, product names, and any other registered information. For copies of this form, or help in filling it out, contact Nancy Dunn at Ext. 5562.

You must also complete and submit an Art Summary Sheet for each article. Be sure and include the current status of each piece of art. Check with Production if you have any questions about this form or about the artwork for a specific article.

Finally, complete an Author Identification Sheet. This information is essential because it provides the codes for the production tracking database.

Although this new procedure requires somewhat more time prior to submission, I'm sure we'll all appreciate the time it saves in production. Thanks.

## Designing the Page Layout

### Determining Spacing in a Document

#### Determining Spacing in a Document

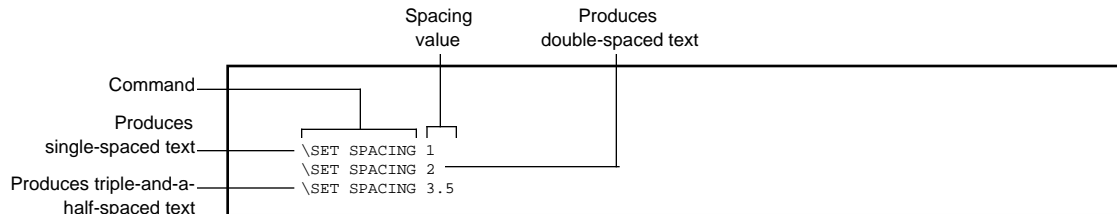
<b>Default:</b>	Line spacing	Single-spaced
	Sentence spacing	Two spaces after period
	Paragraph spacing	Same spacing as entered in edit file

Several factors influence the spacing you choose for a document. If your document is very large, for example, you might want to print it single-spaced to keep the total page count down. When printing the first version of a document, you might want to use double-spaced pages to leave room for editing comments. If your document contains figures and illustrations, you might want to vary the spacing from page to page. In addition to the spacing between lines, you can also increase or decrease the spacing between sentences, or the gap between paragraphs.

**Note** Spacing in this discussion refers to settings that affect your entire document or significant sections within a document. For information on leaving empty lines within your text (the SPACE command), see Section 9, "Adjusting the Text."

#### Setting Line Spacing

The SET SPACING command controls *interline spacing*—the space between lines in your document. A value indicating the spacing you want follows the command. This value can range from 1 to 32. Some printers also allow you to space in half-line increments. Here are some examples of spacing commands:



SET SPACING with no value resets the default interline spacing.

Figure 8-8 illustrates an edit file containing the commands for single and double spacing. Figure 8-9 shows the printed document.

Figure 8-8. Setting Line Spacing

Sets single spacing — \SET SPACING 1  
                                  \SET JOIN ON  
                                  All final drafts must be single-spaced, with three-quarter inch  
                                  margins. Articles must not exceed 15,000 words in length. Please  
                                  attach the Art Summary Sheet, the Trademarks and Copyright Page,  
                                  and the Author Identification Sheet.

Sets double spacing — \SET SPACING 2  
                                  When submitting first drafts for review, please be sure the text is  
                                  double-spaced and all margins are set at one-inch. This allows room  
                                  on the page for questions, comments, and suggestions. You should  
                                  submit four copies of each article.

Figure 8-9. Printed Document Using Different Spacings

All final drafts must be single-spaced, with three-quarter inch margins. Articles must not exceed 15,000 words in length. Please attach the Art Summary Sheet, the Trademarks and Copyright Page, and the Author Identification Sheet.

When submitting first drafts for review, please be sure the text is double-spaced and all margins are set at one-inch. This allows room on the page for questions, comments, and suggestions. You should submit four copies of each article.

#### Setting the Spacing Between Sentences

When you're entering text into your file, you probably put two spaces after a period before beginning the next sentence. Two spaces after a period is also the default setting in TFORM. You can alter the spacing between sentences using the `STYLE SENTENCES` command.

TFORM uses a special formula to decide what makes up a sentence. Sentences must meet all these conditions:

- A sentence consists of words (words must contain two or more characters).
- The words are followed by a period, colon, question mark, or exclamation point.
- The punctuation mark is followed by a single space.
- The space is followed by a capitalized letter or word.

How TFORM recognizes sentences need not concern you, except when sentence structure affects the spacing between your sentences. Watch out for sentences that end with a single letter, or that begin or end with a number. TFORM might not adjust the spacing correctly in these cases. For example, TFORM does not recognize these entries as sentences:

The meeting was held in Room A.  
1988 will be a banner year for new product introductions.  
He's arriving on flight 369.

TFORM places a single space after these sentences because it views this text as words, and words are followed by single space.

TFORM can also misinterpret a period within a sentence. For example, if you have an abbreviated form of address followed by a capitalized name, TFORM inserts two spaces between the period and the name:

Come meet Ms. Riveria and welcome her to our company.

See the discussion of the required blank metacharacter in the *PS TEXT FORMAT Reference Manual* for ways of overcoming these problems.

To change the spacing between sentences, use the `STYLE SENTENCES` command. This command overrides the actual number of spaces in the edit file. You can change the spacing to 0, 1, or 2. To put a single space between your sentences, enter this command:

```
\STYLE SENTENCES 1
```

---

**Note** You must have joining on to use the `STYLE SENTENCES` command. See Section 9, "Adjusting the Text," for information on using the `SET JOIN` command.

---

`STYLE SENTENCES` alone, with no value, resets the default sentence spacing.

Figure 8-10 illustrates three edit files, each set for a different sentence spacing. Figure 8-11 shows how the sentence spacings appear in the printed document.

## Designing the Page Layout

### Setting the Spacing Between Sentences

---

**Figure 8-10. Setting Sentence Spacing**

By default, sentences are separated by two spaces.	<pre>\SET JOIN ON As of May 15, authors must produce all articles on one of the specified word processors.  Authors should run a spellcheck on the text prior to submission.  A disk copy of the file must accompany the hard copy.</pre>
Inserts a single space between sentences	<pre>\NEW \STYLE SENTENCES 1 \SET JOIN ON As of May 15, authors must produce all articles on one of the specified word processors.  Authors should run a spellcheck on the text prior to submission.  A disk copy of the file must accompany the hard copy.</pre>
Eliminates all spaces between sentences	<pre>\NEW \STYLE SENTENCES 0 \SET JOIN ON As of May 15, authors must produce all articles on one of the specified word processors.  Authors should run a spellcheck on the text prior to submission.  A disk copy of the file must accompany the hard copy.</pre>

---

Figure 8-11. Printed Document Using Different Sentence Spacings

As of May 15, authors must produce all articles on one of the specified word processors. Authors should run a spellcheck on the text prior to submission. A disk copy of the file must accompany the hard copy.

As of May 15, authors must produce all articles on one of the specified word processors. Authors should run a spellcheck on the text prior to submission. A disk copy of the file must accompany the hard copy.

As of May 15, authors must produce all articles on one of the specified word processors. Authors should run a spellcheck on the text prior to submission. A disk copy of the file must accompany the hard copy.



## Designing the Page Layout

### Setting the Spacing Between Paragraphs

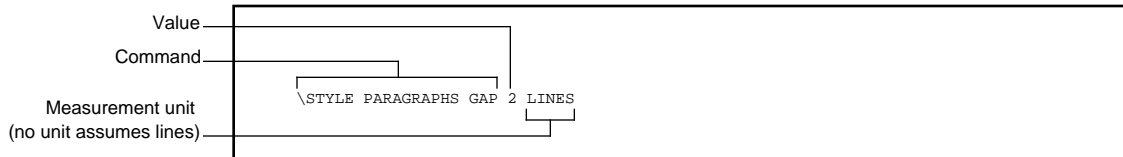
**Setting the Spacing Between Paragraphs** When you enter paragraphs in your edit file, you must insert at least one blank line between paragraphs. TFORM defines paragraphs as text coming between blank lines. If you don't indicate otherwise, TFORM keeps the exact number of blank lines you put between paragraphs.

**Note** There are some commands—such as FOOTER, HEADER, NEW, STYLE, and TITLE—that automatically signal the beginning of a new paragraph. When you're using these commands, you don't need a blank line to indicate a new paragraph.

You can change the *paragraph gap*—spacing between paragraphs—by using the STYLE PARAGRAPHS GAP command. Entering this command overrides the actual number of blank lines between paragraphs in your edit file. The command must include a value. If you want, you can include a measurement unit. If you don't enter a measurement unit, TFORM assumes you mean lines.

When interpreting the STYLE PARAGRAPHS GAP command, TFORM takes into account the interline spacing you've set for the document. If your document is single-spaced, the gap is exactly the number of single-spaced lines that you enter. If your document is double-spaced, the gap is the number of double-spaced lines plus one. In triple spacing, the gap is the number of triple-spaced lines plus two, and so on.

For example, in a single-spaced document, the following command results in two lines between paragraphs. In a double-spaced document, it results in five lines (two double-spaced lines plus one):



If you enter `STYLE PARAGRAPHS GAP 0`, TFORM removes all space from between paragraphs, even the blank line that defines the paragraph end.

**Note** You must have joining in effect to use the `STYLE PARAGRAPHS` command. See Section 9, "Adjusting the Text," for information on using the `SET JOIN` command.

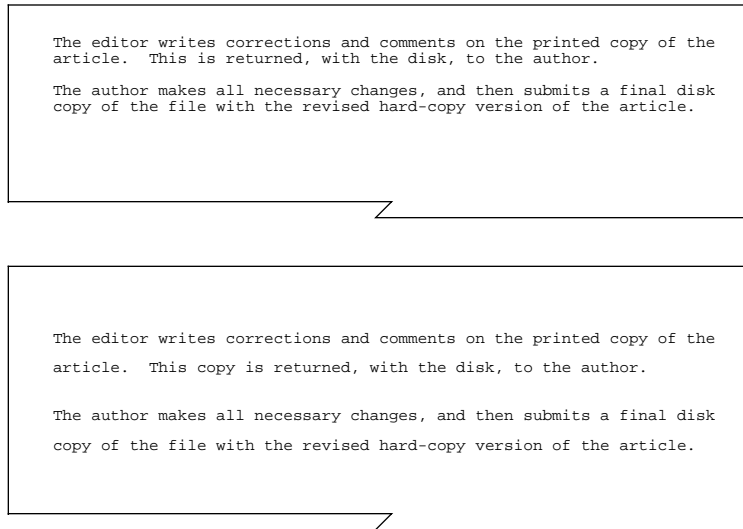
If you enter the `STYLE PARAGRAPHS GAP` command with no values, TFORM returns to the default setting.

Figure 8-12 illustrates two versions of the same edit file using `STYLE PARAGRAPHS GAP` with single-spaced and double-spaced text. Figure 8-13 shows the resulting printed documents.

**Figure 8-12. Setting the Paragraph Gap**

<p>Inserts a single line between paragraphs in a single-spaced document</p>	<pre> \SET JOIN ON \SET SPACING 1 \STYLE PARAGRAPHS GAP 1 The editor writes corrections and comments on the printed copy of the article. This copy is returned, with the disk, to the author.  The author makes all necessary changes, and then submits a final disk copy of the file with the revised hard-copy version of the article.         </pre>
<p>Inserts three lines between paragraphs in a double-spaced document (one double-spaced line plus one)</p>	<pre> \SET JOIN ON \SET SPACING 2 \STYLE PARAGRAPHS GAP 1 The editor writes corrections and comments on the printed copy of the article. This copy is returned, with the disk, to the author.  The author makes all necessary changes, and then submits a final disk copy of the file with the revised hard-copy version of the article.         </pre>

**Figure 8-13. How Interline Spacing Effects Paragraph Gap**



---

A `SPACE` command at the beginning of a paragraph (not preceded by a blank line) overrides the gap setting. The `SPACE` command is treated as text, so `TFORM` does not recognize a new paragraph beginning. Instead, `TFORM` inserts the number of empty lines indicated by the `space` command, not the number set in the `STYLE PARAGRAPHS GAP` command.

`TFORM` considers a `SPACE` command at the end of the paragraph (followed by a blank line) as part of the preceding paragraph. `TFORM` inserts the number of lines in the `SPACE` command, and then puts in the appropriate gap indicated by the `STYLE PARAGRAPHS GAP` command.

For more information on the `SPACE` command, see Section 9, “Adjusting the Text.”



## Designing the Page Layout

---

### Starting on the Next Even or Odd Page

**Starting on the Next Even or Odd Page** If you include EVEN or ODD in the NEW command, TFORM starts a new page with the next even or odd number. The commands look like this:

```
\NEW ODD  
\NEW EVEN
```

Before moving to the next even or odd page, TFORM first checks where you are in your document. For example, if you are on an odd page (or at the top of an even page) when you enter a NEW ODD command, TFORM inserts an even-numbered page. This page contains no text, but it does include any headers or footers, and it is numbered sequentially. TFORM then goes to the next page, the odd page, and resumes formatting text.

The NEW EVEN command starts a page with the next even number, inserting a blank odd page if necessary.

---

**Note** In order to place text on specific pages using the NEW EVEN or NEW ODD command, you need to print out a copy of your document to see where TFORM inserts automatic page breaks. You can then enter the commands to change the page layout.

---

Figure 8-14 illustrates using NEW commands in the edit file. Figure 8-15 shows the resulting pages in the printed documents.

Figure 8-14. Starting New Pages

<p>Moves to the next page</p>	<pre> \HEADER " PAGE-# " \SET JOIN ON The editing process is shown in the figure on the next page. \NEW Figure 8-2 illustrates the relationship between the writing, editing, and production schedules. Each part of the editing process is tied to specific milestones.         </pre>
<p>Moves to the next odd page, inserting a blank even page if necessary</p>	<pre> \HEADER " PAGE-# " \PAGE 3 \SET JOIN ON Using these guidelines to complete a technical edit is explained in the next chapter. \NEW ODD \CENTER THE TECHNICAL EDIT  This chapter discusses the technical edit. In this edit, you must verify that all processes, procedures, and techniques presented in the article are correct.         </pre>

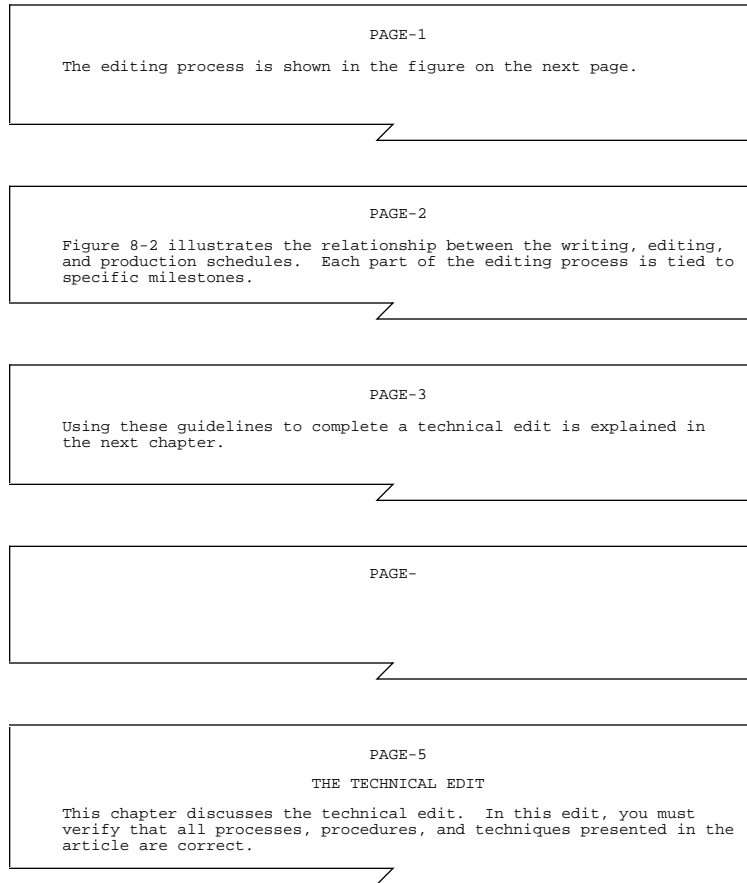
## Designing the Page Layout

---

Starting on the Next Even or Odd Page

---

**Figure 8-15. Printed Results Using the NEW Command**



**Keeping Lines Together** To ensure that text, figures, or tables stay together on a page, use the KEEP command. Of course, your text, figures, or tables must be small enough to fit on a single page. If a KEEP command requires more lines than remain on the current page, TFORM inserts a page break and puts the text on the next page.

You can enter the KEEP command in two ways:

- Tell TFORM to keep items together for a specified number of lines.
- Turn the KEEP command on before the entry, then turn it off after the entry.

When entering a value for the number of lines to keep together, be sure to include blank lines in the total. The total number of lines to keep together cannot exceed the total number of lines that fit on one page. The following command tells TFORM to keep the next ten lines together:

```
\KEEP 10
```

When you don't know how many lines the text, figure, or table will take up, you can turn the KEEP command on and off:

```
\KEEP ON  
\KEEP OFF
```

---

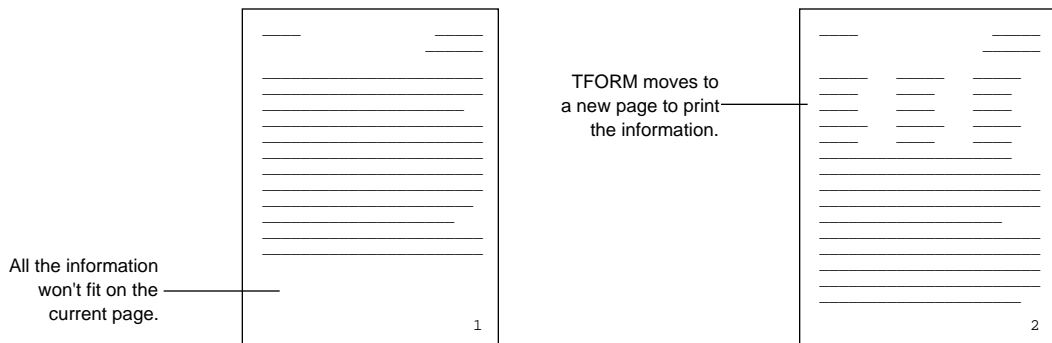
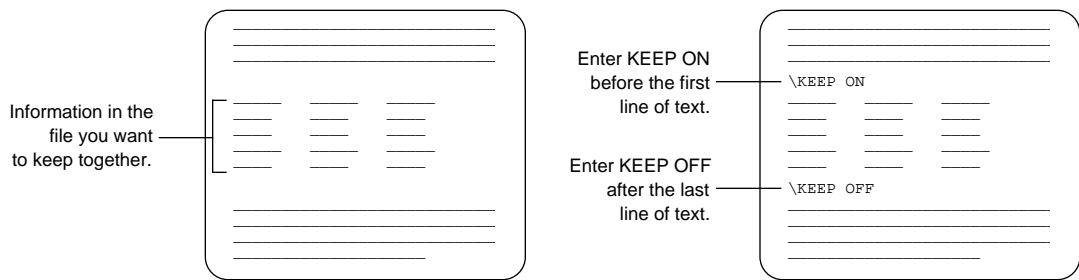
**Note** Don't include any form of the NEW command inside of a KEEP command. TFORM not only ignores the NEW command but also gives you an error message.

---



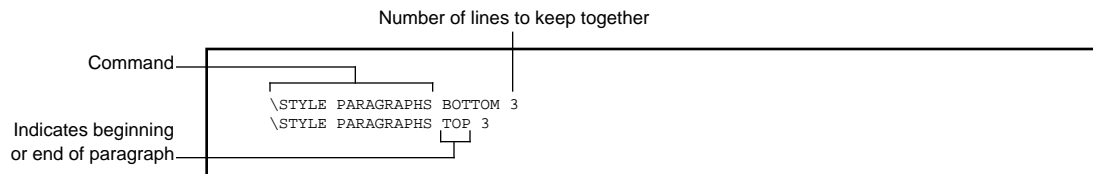
Figure 8-16 illustrates the concept of keeping lines together.

Figure 8-16. Keeping Lines Together



**Splitting Paragraphs** Paragraphs can present special problems when they are interrupted by page breaks. When a page break leaves the first line of a paragraph alone at the bottom of a page, the line is called an *orphan*. When a page break carries over only the last line of a paragraph to the top of a new page, the line is called a *widow*. By default, TFORM widows and orphans contain two lines.

If you want widows or orphans to consist of more than two lines, you change the settings using the STYLE PARAGRAPHS command. To control orphans, enter the STYLE PARAGRAPHS TOP command. To control widows, enter the STYLE PARAGRAPHS BOTTOM command. For example, to make sure that at least three lines of a paragraph stay together at the bottom of a page, or at least three lines are carried over to the next page, you enter the following commands:



You can, of course, control the number of lines by putting a KEEP command before the lines you want together. The STYLE PARAGRAPHS command is most useful when you want to set widow and orphan control for an entire document.

If you enter the STYLE PARAGRAPHS BOTTOM or STYLE PARAGRAPHS TOP command with no values, TFORM reverts to the default setting.

---

**Note** You must have SET JOIN ON to use the STYLE PARAGRAPHS BOTTOM or STYLE PARAGRAPHS TOP command.

---

---

**Summary** In this section you learned how to control the page layout of your document. Table 8-1 provides a quick summary of the commands discussed in this section.

---

**Table 8-1. Commands Used in Page Layout**

To Control	Use This Command
Paper Size	STYLE FORM WIDTH STYLE FORM LENGTH
Margins	STYLE MARGIN TOP STYLE MARGIN BOTTOM STYLE MARGIN LEFT STYLE MARGIN RIGHT
Line Spacing	SET SPACING
Sentence Spacing	STYLE SENTENCES
Paragraph Spacing	STYLE PARAGRAPHS GAP
Page Break	NEW
Text Breaks	KEEP
Widows and Orphans	STYLE PARAGRAPHS BOTTOM STYLE PARAGRAPHS TOP

---

---

# 9 Adjusting the Text

---

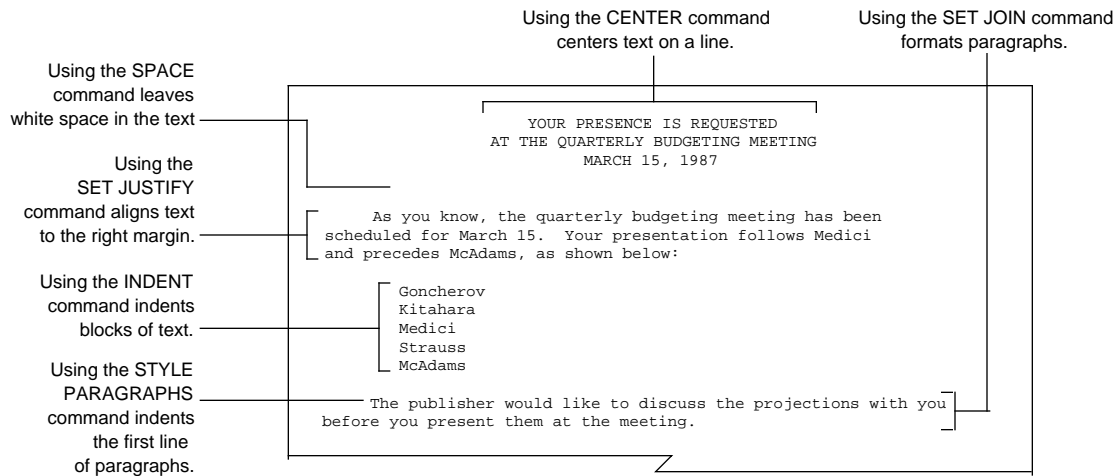
**Overview** This section discusses the TFORM commands you use to format text in an edit file. These commands control the appearance of the text in a printed document. You can adjust text in the following ways:

- Join text in paragraphs
- Justify text to the right margin
- Indent text
- Center text
- Insert white space in the text

Figure 9-1 shows a page of printed text formatted by the TFORM commands discussed in this section.

---

**Figure 9-1. Using TFORM Commands to Format Text**



## Adjusting the Text

---

How TFORM Works With Your Edited Text

---

### How TFORM Works With Your Edited Text

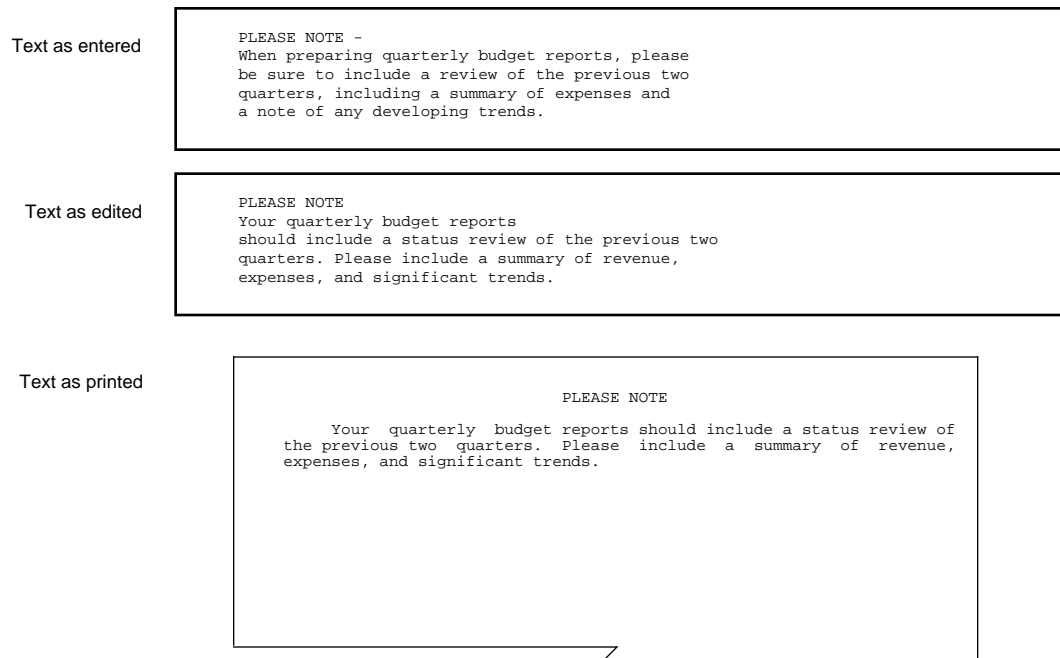
When entering text in an edit file, you determine the end of each line by pressing `RETURN` or `ENTER`. If you want to see all the text as you enter it, you should limit the lines to what is visible on the terminal screen (about 70 characters). Your lines are about the same length, but your right margin varies depending on when you press `RETURN` and `ENTER`. While you're entering text, you don't have to bother with alignment, indentation, or spacing.

When editing the text in your file, you insert, delete, and move text. This rearrangement could include only a few lines or might involve large blocks of text. Editing your text can result in a somewhat disjointed text display.

You don't need to "clean up" the text in your edit file before printing it. Using a few simple TFORM commands, you can produce a document with indentation, uniform paragraphs, and justified text, without rearranging the lines in your edit file. Figure 9-2 shows three stages in the creation of a printed document:

1. The text as entered
2. The text as edited
3. The text as printed

Figure 9-2. Three Stages of a Document



The TFORM commands you use to format text are described in the rest of this section. For your convenience, default settings for the commands appear at the beginning of each topic. If you use the default settings, you don't have to enter any text formatting commands.

## Adjusting the Text

---

### Joining Text

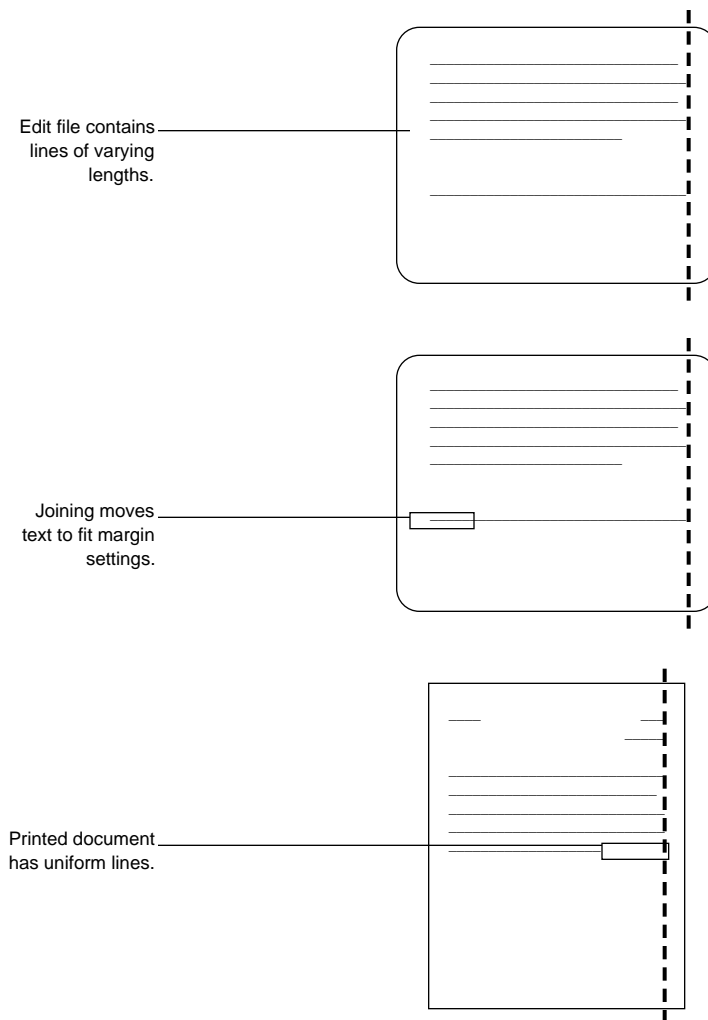
---

**Joining Text** **Default:** Joining is not in effect. Text lines appear exactly as entered in the edit file.

*Joining* is the process of moving words either up or down a line so that they fit into the space allotted by the margin settings. When joining is in effect, TFORM evaluates each line as it formats the text. If the line is too long, TFORM breaks off one or more words and moves them down to the next line. If the line is too short, TFORM brings up words from the next line. TFORM continues joining text until it reaches the end of a paragraph. Joining starts again with the first line of the next paragraph.

Figure 9-3 illustrates the process of joining text.

Figure 9-3. Joining Text





## Adjusting the Text

---

### Turning Joining On and Off

**Turning Joining On and Off** You control joining with the SET JOIN command. To begin joining, enter this command:

```
\SET JOIN ON
```

Joining stays in effect until you turn it off by entering this command:

```
\SET JOIN OFF
```

Entering SET JOIN ON at the beginning of your file turns on joining for the entire document. When you want to stop joining—for example, to enter a table—enter SET JOIN OFF. Return to joining by entering another SET JOIN ON command.

You can also interrupt joining temporarily by using the BREAK command. Entering BREAK (or simply the \ character) alone on a line stops joining for the previous line. You don't have to turn joining back on after a BREAK line; TFORM automatically resumes joining on the following line.

Figure 9-4 shows the join commands in an edit file. Figure 9-5 illustrates how joining looks in the printed document when the text is joined to the default width of 70.

Figure 9-4. Setting Joining

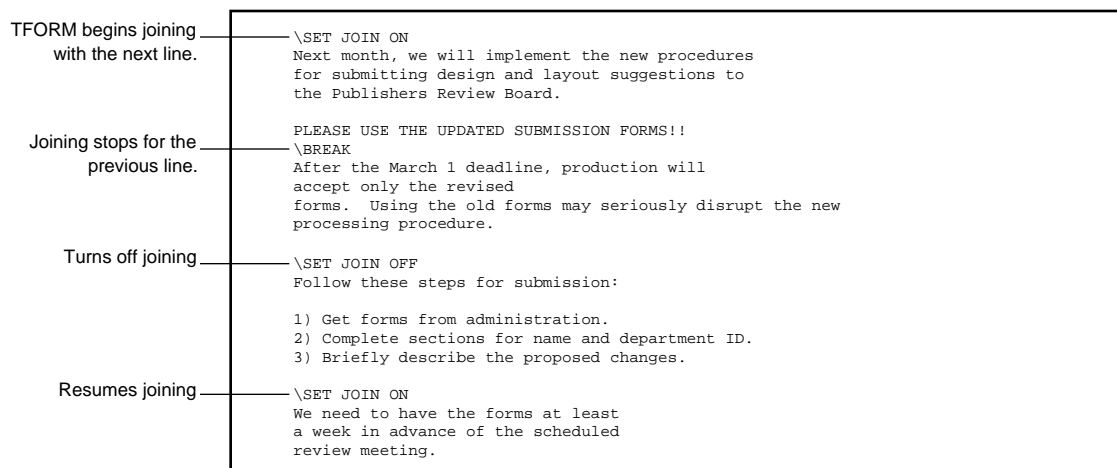
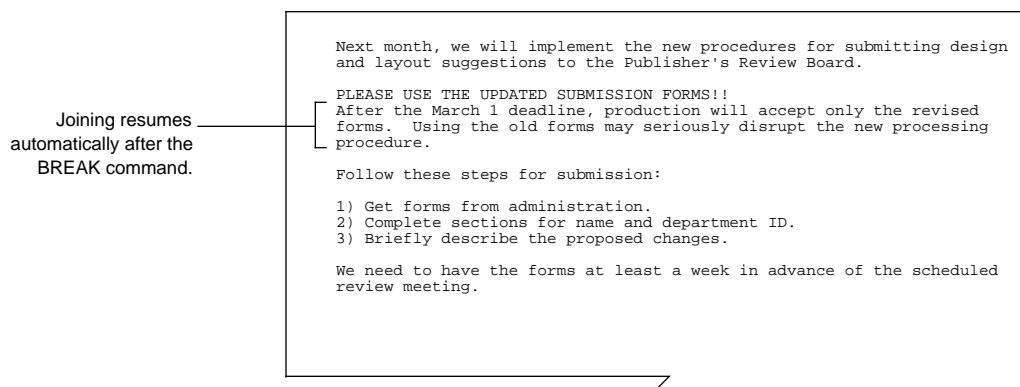


Figure 9-5. Printed Document Using Joining



**Hints on Using Joining** When your file consists of straight text in paragraphs, you'll probably want to set joining and justification on at the beginning of your file and leave it at that. There are situations, however, when joining must be off.

#### **When Joining Should Be On**

You must have joining on for commands to work properly. Here are some examples of when joining should be on:

- You frequently receive the error message "Line was longer than the text area by n columns." This message tells you that the line length you are using in your edit file is longer than the space allotted by your margin settings in TFORM. Joining automatically adjusts your lines.
- You use `STYLE SENTENCES` or `STYLE PARAGRAPHS` in your page layout commands. These commands do not work properly unless joining is on.
- You use the `LIST` command. To format lists correctly, TFORM requires that joining be turned on.
- You set margins that reduce the space for text to less than 70 characters. TFORM can't move text unless you have joining in effect.
- You get unexpected space at the top of a page. When joining is on, TFORM eliminates blank lines if they appear at the beginning of a new page.

### When Joining Should Be Off

Sometimes, joining can cause unexpected results. The following are examples of when joining should be off:

- Your file contains columns of text. Joining causes columns to be run together.
- Your file contains text you've indented using the space bar or the TAB key. Joining can move text that you want to stay on separate lines.
- Your file contains examples or illustrations in which the placement of text is critical. When you want the printed document to look exactly like the display in your edit file, do not turn on joining.

Remember, you can stop joining by using SET JOIN OFF, or you can interrupt joining for a single line by using the BREAK command.

---

**Justifying Text** **Default:** Text is always left-justified. Right justification is not in effect.

*Justification* is the process of spacing out the text so that each line aligns to the respective right and left margins. TFORM always justifies text to the left margin, and normally leaves a *ragged margin* on the right. Turning on justification causes text to be justified to both the left and right margins. During justification, TFORM adds extra spaces between characters until the line only fills the space allotted by the margin settings. TFORM never justifies the last line of a paragraph.

**Turning Justification  
On and Off** To begin right justification, enter this command:

```
\SET JUSTIFY ON
```

To return to only left justification, enter this command:

```
\SET JUSTIFY OFF
```

Entering SET JUSTIFY ON at the beginning of your file puts right justification in effect for your entire document. When you want to stop justification temporarily, enter SET JUSTIFY OFF. Right justification resumes with the next SET JUSTIFY ON command.

Figure 9-6 shows the justification commands in an edit file. Figure 9-7 illustrates justified text in a printed document.

Figure 9-6. Setting Justification

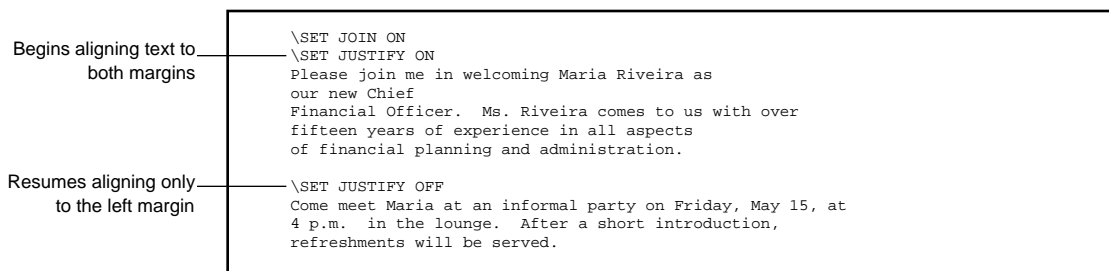
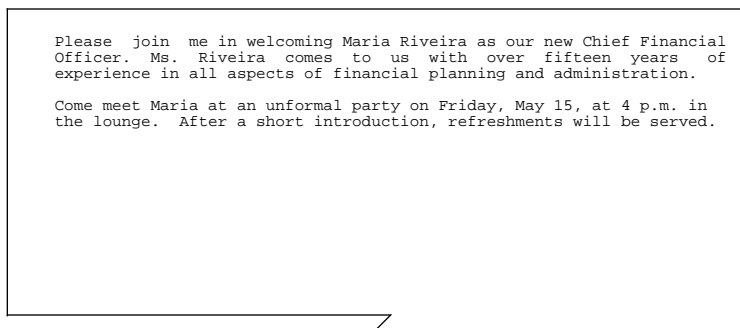


Figure 9-7. Printed Document Using Justification



## Adjusting the Text

### Using Joining and Justification Together

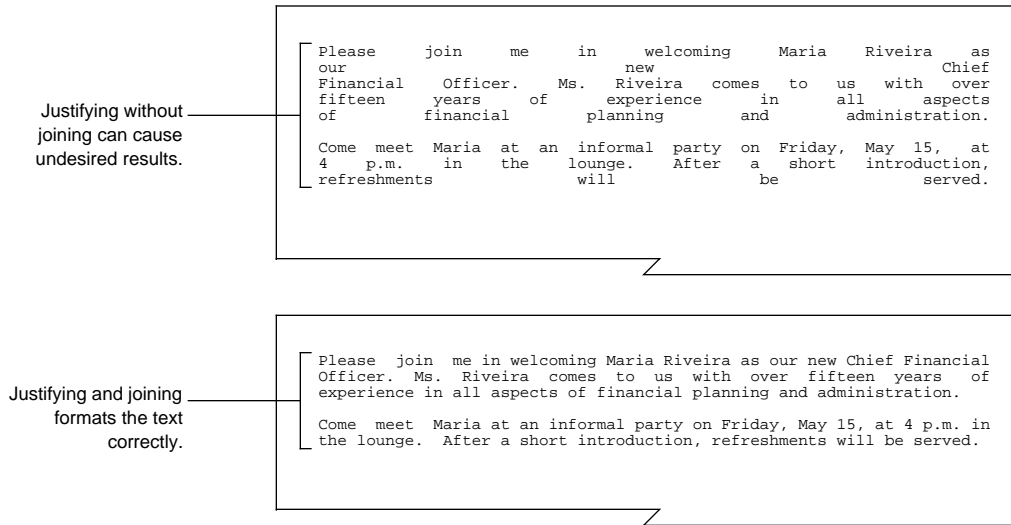
#### Using Joining and Justification Together

Justification works hand in hand with joining. Trying to justify text without joining often results in an unacceptable amount of space between words in a short line. Without joining in effect, TFORM can't bring up words to fill the space on the line.

In most cases, you use both commands as a unit. If you turn both joining and justification on, a SET JOIN OFF command also turns off justification. The next SET JOIN ON command resumes both joining and justification (unless you've entered a separate SET JUSTIFY OFF command).

Figure 9-8 shows two printed documents. In the first one, joining is off and justifying is on. In the second document, both joining and justifying are in effect.

Figure 9-8. Using Joining and Justification



---

**Indenting Text** **Default:** No indentation is in effect. Text appears as entered in edit file.

TFORM has several commands you use to *indent* text, depending on the specific task you want to perform. The following paragraphs discuss using indentation in these ways:

- Indenting the first line of paragraphs
- Indenting single lines of text
- Indenting a block of text

**Indenting the First Line of Paragraphs**

To set a standard indentation for the first lines of paragraphs, use the STYLE PARAGRAPHS INDENT command. This command requires a value to indicate how far you want to indent from the left margin. TFORM assumes this value to be in characters, but you can enter other units of measurement. To indent all paragraphs eight characters from the left, enter this command at the beginning of the edit file:

```
\STYLE PARAGRAPHS INDENT 8
```

The STYLE PARAGRAPHS INDENT command lets you use relative values. A plus sign (+) before the value moves the text to the right. A negative sign (-) before the value moves the text to the left. If you want a *hanging indent*—an extension of the text into the left margin—enter STYLE PARAGRAPHS INDENT with a minus before the value. Hanging indents can be a useful device to draw attention to specific text.

When you use plus and minus values with the STYLE PARAGRAPHS INDENT command, TFORM adds or subtracts the values from the current setting. To return to the default setting, which indents paragraphs the number of spaces you put in the edit file, enter STYLE PARAGRAPHS INDENT with no value. Figure 9-9 presents the same text in two edit files. In the first file, each paragraph is indented using a positive value. In the second file, each paragraph is indented using a negative value. Both settings assume that prior to these commands, no paragraph indentation was in effect. Figure 9-10 shows the results of these settings.



## Adjusting the Text

---

### Indenting the First Line of Paragraphs

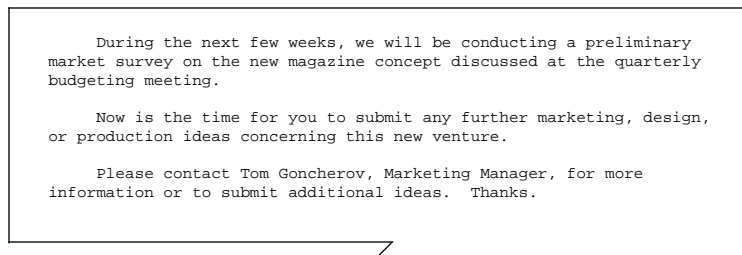
---

**Figure 9-9. Indenting Paragraphs**

Indents to the right five characters	<pre>\NEW \STYLE PARAGRAPHS INDENT 5 \SET JOIN ON During the next few weeks, we will be conducting a preliminary market survey on the new magazine concept discussed at the quarterly budgeting meeting.  Now is the time for you to submit any further marketing, design, or production ideas concerning this new venture.  Please contact Tom Goncherov, Marketing Manager, for more information or to submit additional ideas. Thanks.</pre>
Indents to the left five characters	<pre>\NEW \STYLE PARAGRAPHS INDENT -5 \SET JOIN ON ATTENTION!! During the next few weeks, we will be conducting a preliminary market survey on the new magazine concept discussed at the quarterly budgeting meeting.  NOW is the time for you to submit any further marketing, design, or production ideas concerning this new venture.  PLEASE contact Tom Goncherov, Marketing Manager, for more information or to submit additional ideas. Thanks.</pre>

---

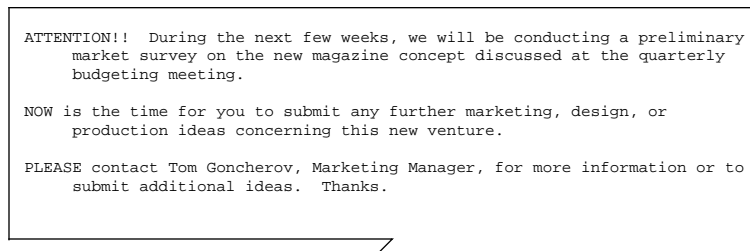
Figure 9-10. Printed Documents Showing Indented Paragraphs



During the next few weeks, we will be conducting a preliminary market survey on the new magazine concept discussed at the quarterly budgeting meeting.

Now is the time for you to submit any further marketing, design, or production ideas concerning this new venture.

Please contact Tom Goncherov, Marketing Manager, for more information or to submit additional ideas. Thanks.



ATTENTION!! During the next few weeks, we will be conducting a preliminary market survey on the new magazine concept discussed at the quarterly budgeting meeting.

NOW is the time for you to submit any further marketing, design, or production ideas concerning this new venture.

PLEASE contact Tom Goncherov, Marketing Manager, for more information or to submit additional ideas. Thanks.

## Adjusting the Text

### Indenting a Single Line

**Indenting a Single Line** There are two ways to indent a single line of text. You can:

- Use the TI (Temporary Indent) command
- Use the INDENT command without entering any value for number of lines

The TI command indents only a single line and only from the left. For example, to move the line of text immediately following this command in twelve characters from the left margin, enter:

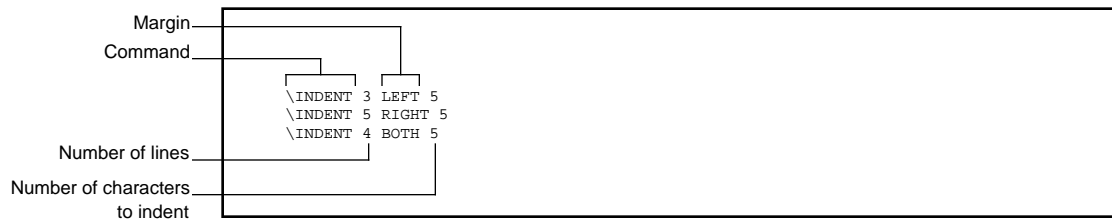
```
\TI 12
```

You can use the INDENT command to indent a single line, but it's easier to use TI, which avoids the problem of unexpected indentation. This problem is discussed later in this section and is illustrated in Figures 9-11 and 9-12.

**Indenting Blocks of Text** The STYLE MARGIN command controls the length of the lines in a printed document. When you want to change the left or right margins temporarily, you use the INDENT command.

Usually, you indent text toward the center of the page, which increases the size of your margins. In TFORM, you can also indent outwards so that text appears in the margin areas.

The INDENT command contains the number of lines you want to indent, the margin to indent from, and a value indicating how far to indent. You can indent from the LEFT, RIGHT, or BOTH margins. The value indicates characters, unless you enter a different unit of measurement. Here are some examples of INDENT commands:




An INDENT command with no value for number of lines indents only the next line. If you don't include a margin in the command, TFORM indents from the left. For example:

A single value indicates the number of characters to indent on the next line only.

The first value indicates the number of lines; the second value indicates the number of characters.

```
\INDENT 5  
\INDENT 5 5
```

No margin indicated; TFORM indents from the left.



**Note** The INDENT command indents the lines immediately following the command, including blank lines as well as text. If the text you want indented includes blank lines, be sure to count them when you enter the value in the command.

## Adjusting the Text

### Indenting a Single Line

---

To TFORM, the value you enter as part of an INDENT command refers to the number of lines in your edit file. This value might not be the same number of lines that the text takes up when it's printed. This difference in input lines and output lines can cause some unexpected results, as shown in Figures 9-11 and 9-12.

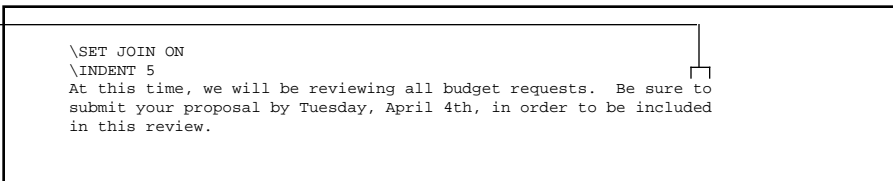
In Figure 9-11, you want to indent the first line of text five characters to the left. TFORM assumes you want to indent all of the text on the first line. Once it inserts the five-character space, however, the text no longer fits within the margins. So, TFORM moves a word or two down to the next line. TFORM then indents the second line too, because it contains text originally on the first line, as shown in Figure 9-12.

---

**Figure 9-11. Indenting Commands in the Edit File**

When formatted,  
this text is moved  
to the second line.

```
\SET JOIN ON
\INDENT 5
At this time, we will be reviewing all budget requests. Be sure to
submit your proposal by Tuesday, April 4th, in order to be included
in this review.
```

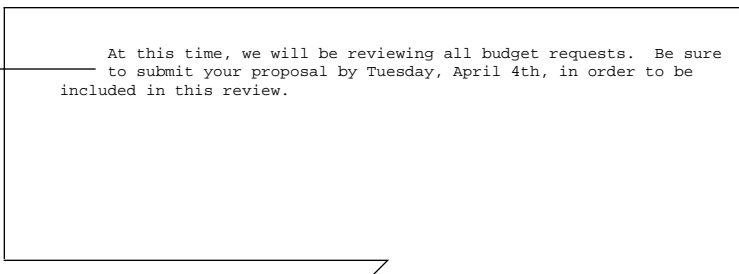


---

**Figure 9-12. Unexpected Indentation in the Printed Document**

TFORM indents all  
the text from the first  
line in the edit file.

```
At this time, we will be reviewing all budget requests. Be sure
to submit your proposal by Tuesday, April 4th, in order to be
included in this review.
```

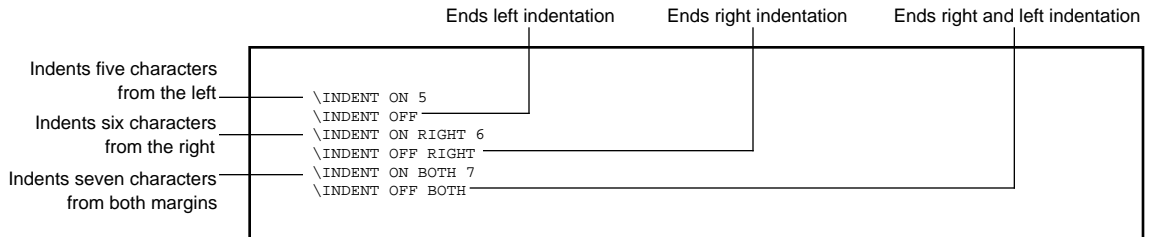


To avoid indentation problems resulting from the difference in the number of input and output lines, be sure to leave space in the line for the indentation. If you're indenting only a single line, you can use the TI (Temporary Indent) command, as explained earlier in this section.

**Turning Indentation On and Off**

When you're not sure how many lines you want to indent, you can control indentation by using ON and OFF. Once you enter an INDENT ON command, all lines are indented until TFORM reaches an INDENT OFF command. You must include a value in the INDENT ON command so TFORM knows how far to indent, but you don't need a value in the INDENT OFF command.

INDENT ON and INDENT OFF, with no margin indicator, refers only to indentation from the left margin. If you want to turn on and turn off indentation from the right margin, or from both margins, you must include the margin indicator in the INDENT command. For example:

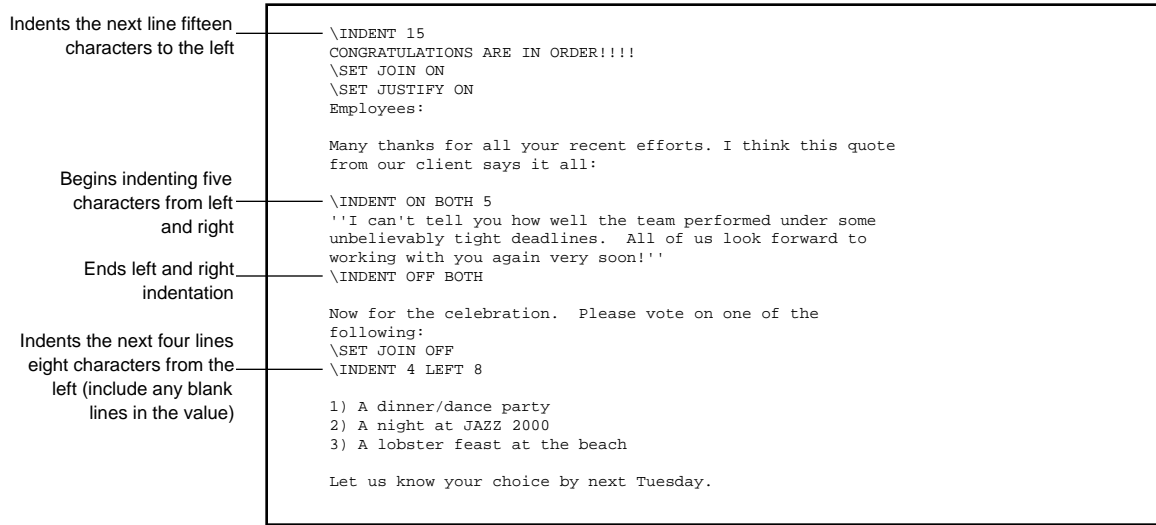


## Adjusting the Text

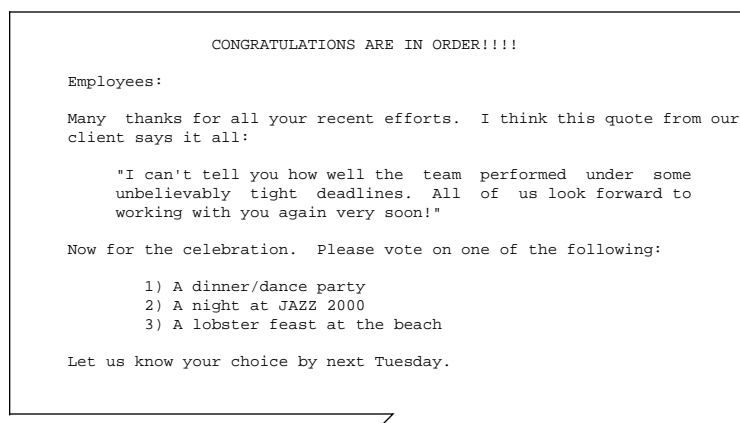
### Turning Indentation On and Off

Figure 9-13 shows indentation commands in an edit file. Figure 9-14 illustrates how the indentation appears in the printed document.

Figure 9-13. Indenting Text



---

**Figure 9-14. Indented Text in a Printed Document**

---

**Using Relative Values** TFORM indents relative to the current margin settings. You can use the plus sign (+) or the minus sign (-) before the number of characters to indicate the direction you want to indent. A plus sign indents toward the center of the page. TFORM assumes positive indentation when you're indenting from the original margins. For example, each of these two INDENT commands indents three lines, five characters to the left:

```
\INDENT 3 5
\INDENT 3 +5
```

When you want to indent your text outwards—that is, beyond the current margins—you use negative values. The following command moves the text outward, three characters from both the left and right margins:

```
\INDENT ON BOTH -3
```



## Adjusting the Text

### Using Relative Values

Once you have an `INDENT` command in effect, you can use plus and minus signs to create additional indentations within an already indented area of text. Including indented text within another indentation is called *nested indentation*. For example, if you are using an `INDENT` command that indents five characters from the left, entering another `INDENT` command with a `+3` value increases the indentation to eight. In the same way, an `INDENT` command of `-3` moves the text out three characters from the current indentation.

Entering the `INDENT` command by itself, with no values or margins, cancels all indentation. Margins are reset to the settings in the last `STYLE MARGIN` commands.

Figure 9-15 shows an edit file using relative indentation settings. Figure 9-16 shows the results of nested indentation.

Figure 9-15. Using Relative Values to Indent Text

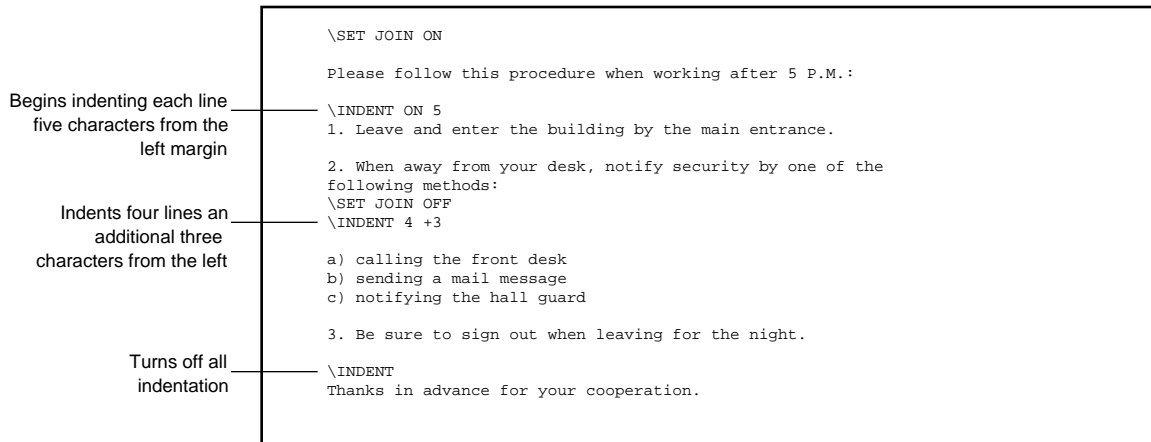
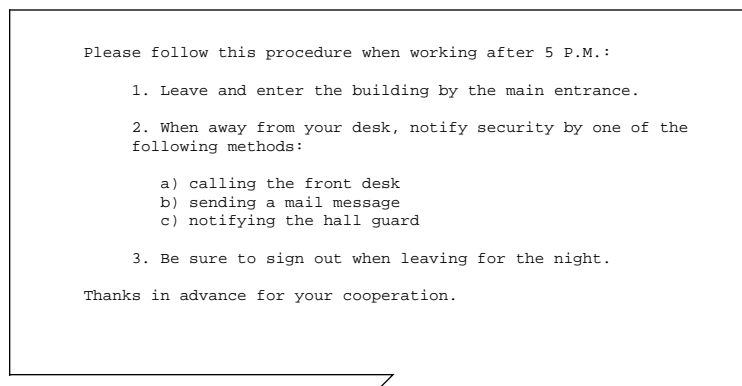


Figure 9-16. Nested Indentation in a Printed Document



## Adjusting the Text

### Centering Text

---

**Centering Text** **Default:** Centering is not in effect. Text is aligned to the left.

You use the `CENTER` command to center lines or blocks of text in your printed document. You specify the number of lines you want centered by including a number in the command. If you enter `CENTER` with no value, only the next line is centered. For example:



---

**Note** The `CENTER` command centers the lines immediately following the command, including blank lines as well as text. If the text you want centered includes blank lines, be sure to count them when you enter the value in the command.

---

You can also use the `CENTER` command with `ON` and `OFF`. `TFORM` centers all the text lines between the `CENTER ON` and `CENTER OFF` commands. Figure 9-17 illustrates an edit file using both forms of the `CENTER` command. Figure 9-18 shows the resulting printed document.

Figure 9-17. Centering Text

The diagram shows a rectangular box containing text. On the left side, three labels with arrows point to specific lines of code within the text:

- Centers the next three lines** points to the line `\CENTER 3`.
- Begins centering** points to the line `\CENTER ON`.
- Ends centering** points to the line `\CENTER OFF`.

The text inside the box is as follows:

```
\CENTER 3
STAFF MEETING
WEDNESDAY - APRIL 20
ROOM 1223, MAIN BUILDING

Due to some scheduling conflicts, our staff meeting will be
held on Wednesday afternoon this week. Please be on time.

\CENTER ON
*****SPECIAL ANNOUNCEMENT*****
THE LOBSTER WINS BY A LONG SHOT
COME CELEBRATE ON SATURDAY AT KING BEACH
*****FRIENDS AND FAMILIES WELCOME*****
\CENTER OFF

Get maps and other info from Debbie. See you there!
```

Figure 9-18. Centered Text in a Printed Document

The diagram shows a rectangular box containing the text from Figure 9-17, but with all text centered horizontally. The text is as follows:

```
STAFF MEETING
WEDNESDAY - APRIL 20
ROOM 1223, MAIN BUILDING

Due to some scheduling conflicts, our staff meeting will be
held on Wednesday afternoon this week. Please be on time.

*****SPECIAL ANNOUNCEMENT*****
THE LOBSTER WINS BY A LONG SHOT
COME CELEBRATE ON SATURDAY AT KING BEACH
*****FRIENDS AND FAMILIES WELCOME*****

Get maps and other info from Debbie. See you there!
```

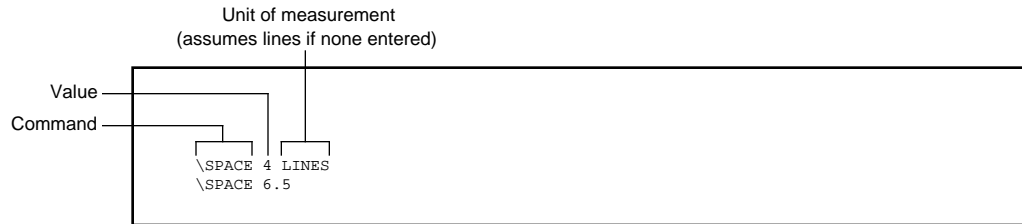
## Adjusting the Text

### Adding White Space to the Text

#### Adding White Space to the Text

Most of the breaks in text in your edit file occur when you insert a blank line by pressing the `RETURN` or `ENTER` key. For example, you use blank lines to define the beginning and end of paragraphs.

You can add larger areas of *white space* to your document by using the `SPACE` command. This command tells TFORM to move the paper a certain distance. The value you enter tells TFORM how many lines down to move the paper. For example:



`SPACE` always inserts single-spaced lines, no matter what line spacing you're using in the document. If you enter `SPACE 5`, for example, you get five single-spaced lines, even if you're printing a double-spaced document.

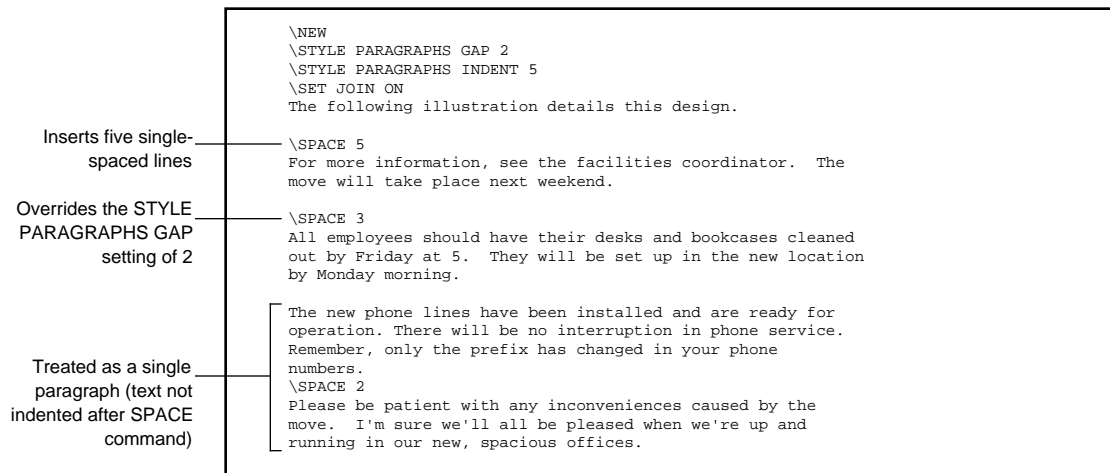
The `SPACE` command is useful if you want to leave some white space at the top of a page—for example, to include an illustration. Normally, TFORM discards any blank lines when they occur at a page break. The `SPACE` command, however, always produces white space, even at the top of a new page.

**Note** Do not use the `SPACE` command to separate paragraphs in your edit file. TFORM views the empty lines created by `SPACE` as text. Therefore, features that work on paragraphs, such as joining or paragraph styling commands, treat the areas before and after the white space as part of a continuous paragraph.

A `SPACE` command at the beginning of a paragraph overrides the `STYLE PARAGRAPHS GAP` setting.

Figure 9-19 shows an edit file using three separate SPACE commands. Figure 9-20 shows the resulting printed page.

Figure 9-19. Using the SPACE Command



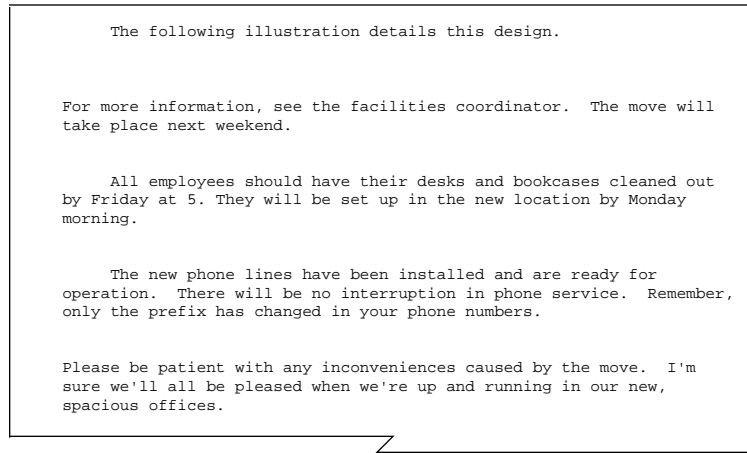
## Adjusting the Text

---

### Adding White Space to the Text

---

**Figure 9-20. White Space in a Printed Document**



**Summary** This section has discussed the commands you use in your edit file to adjust how your text looks when it is printed. Table 9-1 presents a quick summary of the commands discussed in this section.

---

**Table 9-1. Commands Used in Adjusting Text**

To Control	Use This Command
Joining	SET JOIN
Justification	SET JUSTIFY
Indentation	INDENT TI STYLE PARAGRAPHS INDENT
Centering Text	CENTER
White Space	SPACE

---



---

# 10 Adding Headers and Footers

---

**Overview** This section discusses the commands you use to put headers and footers in your printed document. A *header* is text that appears along the top of a page. Generally, headers identify the document or contain information about what is on that specific page. Headers are sometimes called running heads. You enter headers by using the `HEADER` command.

A *footer* is text that appears along the bottom of a page. Footers often contain page numbers, identify the version of the document, or display notices of copyright or confidentiality. You enter footers by using the `FOOTER` command.

With `TFORM HEADER` and `FOOTER` commands, you can do the following:

- Place text in three locations within headers and footers
- Include page numbers, time, and date in headers and footers
- Enter up to ten headers and ten footers for each page
- Place headers or footers on selected pages
- Set separate margins for headers and footers

Header and footer commands entered either at the beginning of a file or before any text on the first page take effect immediately. Headers or footers entered on any other page, or after text on the first page, take effect with the page following the command.

Headers and footers follow the same rules for entry, placement, and use, so they are discussed together in this section.

---

**Note** `TFORM` puts headers and footers only on pages containing text. If you want a header or footer on a blank page, make sure the page contains at least one line of text. `TFORM` counts a blank line or the `SPACE` command as text.

The exception to this rule is when you use the `NEW EVEN` or `NEW ODD` command to generate blank pages. `TFORM` prints headers or footers on these pages. For more information on the `NEW` command, see Section 8 of this manual.

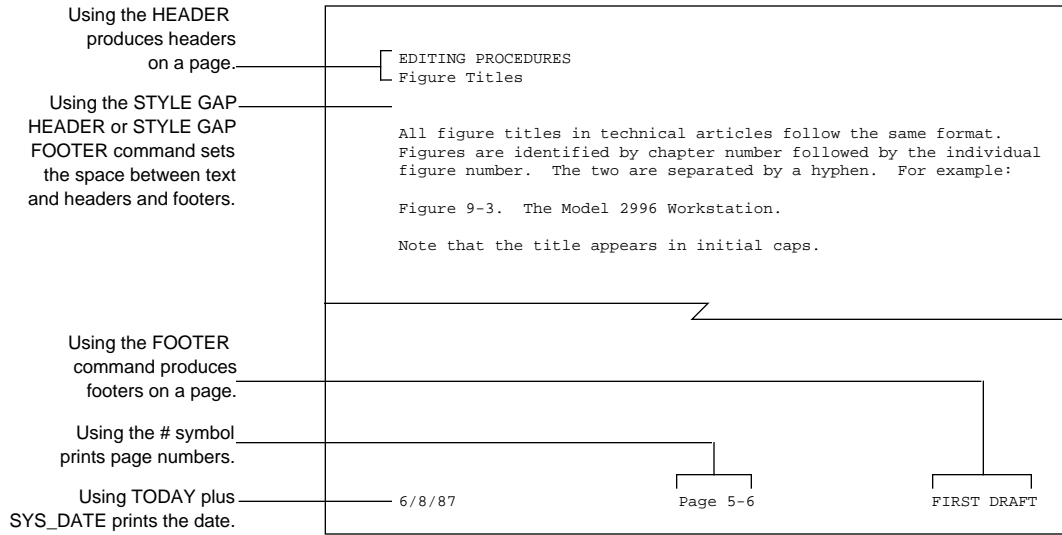
---

## Adding Headers and Footers

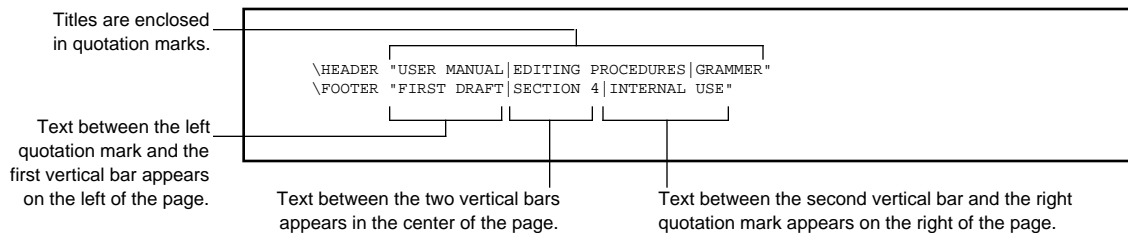
Overview

Figure 10-1 shows a page of a document and lists the TFORM commands used to place headers and footers on a printed page.

Figure 10-1. Using TFORM Commands to Enter Headers and Footers



**Entering Titles** The entry you make into a single header or footer is called a *title*. Titles consist of three areas or *fields*: left, center, and right. Vertical bars separate one field from another. You enclose the entire title in quotation marks. Here is an example of a title for a header and a footer:



You can use any combination of one, two, or three fields. Fields can be empty or filled with spaces. If you leave out the vertical bars and simply enter text, TFORM puts the text on the left. For example, either of these two commands prints a header at the far left top of the page:

```
\HEADER "CORRECT USE OF COMMAS | | "
\HEADER "CORRECT USE OF COMMAS "
```

To center the text, enter it between the vertical bars:

```
\HEADER " |CORRECT USE OF COMMAS | "
```

To place the header on the right, the text follows the vertical bars:

```
\HEADER " | |CORRECT USE OF COMMAS "
```

If you wish, you can put the entries for each field in a title on a separate line. You still need to include the vertical bars so TFORM knows where to put the fields. For example, the following commands result in a single header across the page:

```
\HEADER "USER MANUAL"
\HEADER " |EDITING PROCEDURES | "
\HEADER " | |CORRECT USE OF COMMAS "
```

## Adding Headers and Footers

---

### Entering Titles

---

**Note** Using titles is not limited to header and footer entries. You can put titles anywhere in the text of your document by using the TITLE command. See the *PS TEXT FORMAT Reference Manual* for more information on this command.

---

A header or footer title can be as wide as your margins allow. In practice, however, a title extending across the entire page is difficult to read, especially if it contains more than one field. If your header or footer gets too long, or if the fields overlap, TFORM gives you an error message.

When you have a lot of information to include in headers or footers, consider dividing the information into several separate headers or footers. See “Using Multiple Headers and Footers” later in this section for instructions on entering up to ten headers and footers on a single page.

Figure 10-2 illustrates six different edit files, each containing a header or footer command. Figure 10-3 illustrates the result of each command on a printed page.

---

**Note** If you’re trying out the examples presented in this section, be sure to include some text in your edit file after the HEADER and FOOTER commands. If you don’t include text, the header or footer won’t appear because TFORM does not place headers and footers on blank pages.

---

Figure 10-2. Entering Headers and Footers

Places header title at all three locations	<code>\HEADER "USER MANUAL EDITING PROCEDURES CHECKING FIGURES"</code>
Places header at top center of page	<code>\HEADER " VERIFYING FIGURE NUMBERS "</code>
Places header at far left top of page	<code>\HEADER "EDITING FIGURE TITLES  "</code>
Places footer at all three locations	<code>\FOOTER "FIRST DRAFT SECTION 5 FOR INTERNAL REVIEW"</code>
Places footer at bottom center of page	<code>\FOOTER " COPYRIGHT 1987 "</code>
Places footer at bottom right of page	<code>\FOOTER "  COMPANY CONFIDENTIAL"</code>

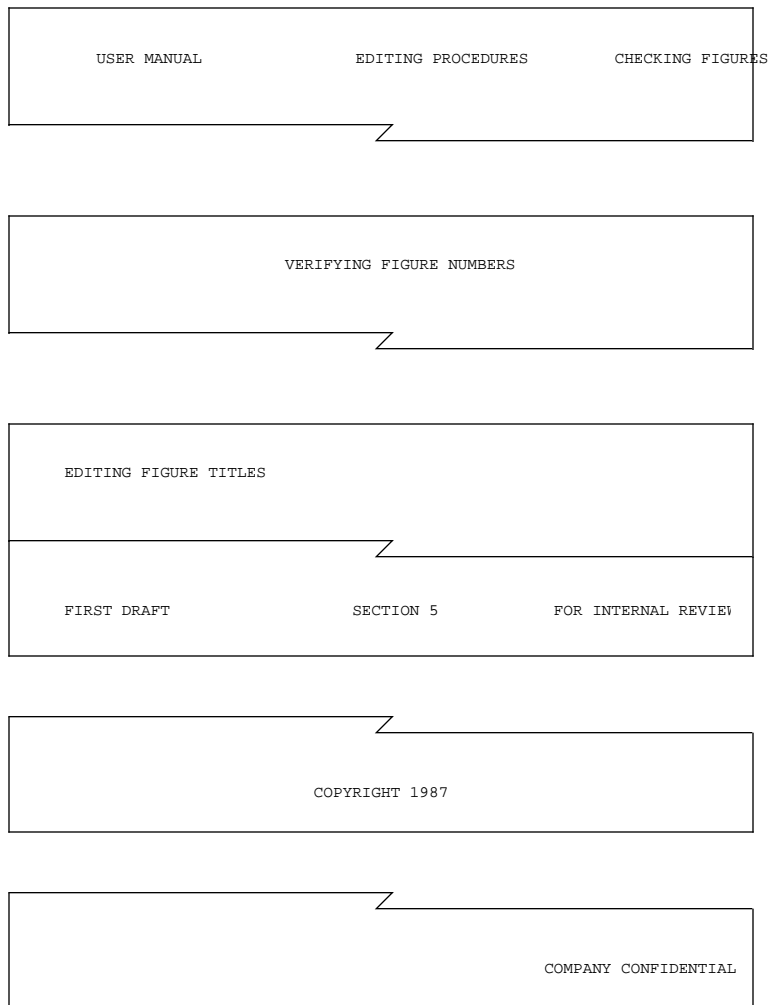
## Adding Headers and Footers

---

Entering Titles

---

**Figure 10-3. Printed Headers and Footers**



---

**Note** When you want a header or footer to reflect information on a specific page, you'll need to make a printed copy of your document to determine the page layout. Then enter the correct title in the HEADER or FOOTER command.

---

---

**Changing Titles** Once you've entered a header or footer, it stays in effect until you enter a new **HEADER** or **FOOTER** command **and** change the entry in the field. **TFORM** carries over fields of a header or footer that you don't change, making it easy for you to replace only selected fields in the title. If you don't put anything between the vertical bars, the existing field stays in the new header or footer. Any text, even a blank space, replaces the field; see Figure 10-4.

The **HEADER** or **FOOTER** command, by itself, eliminates all headers and footers.

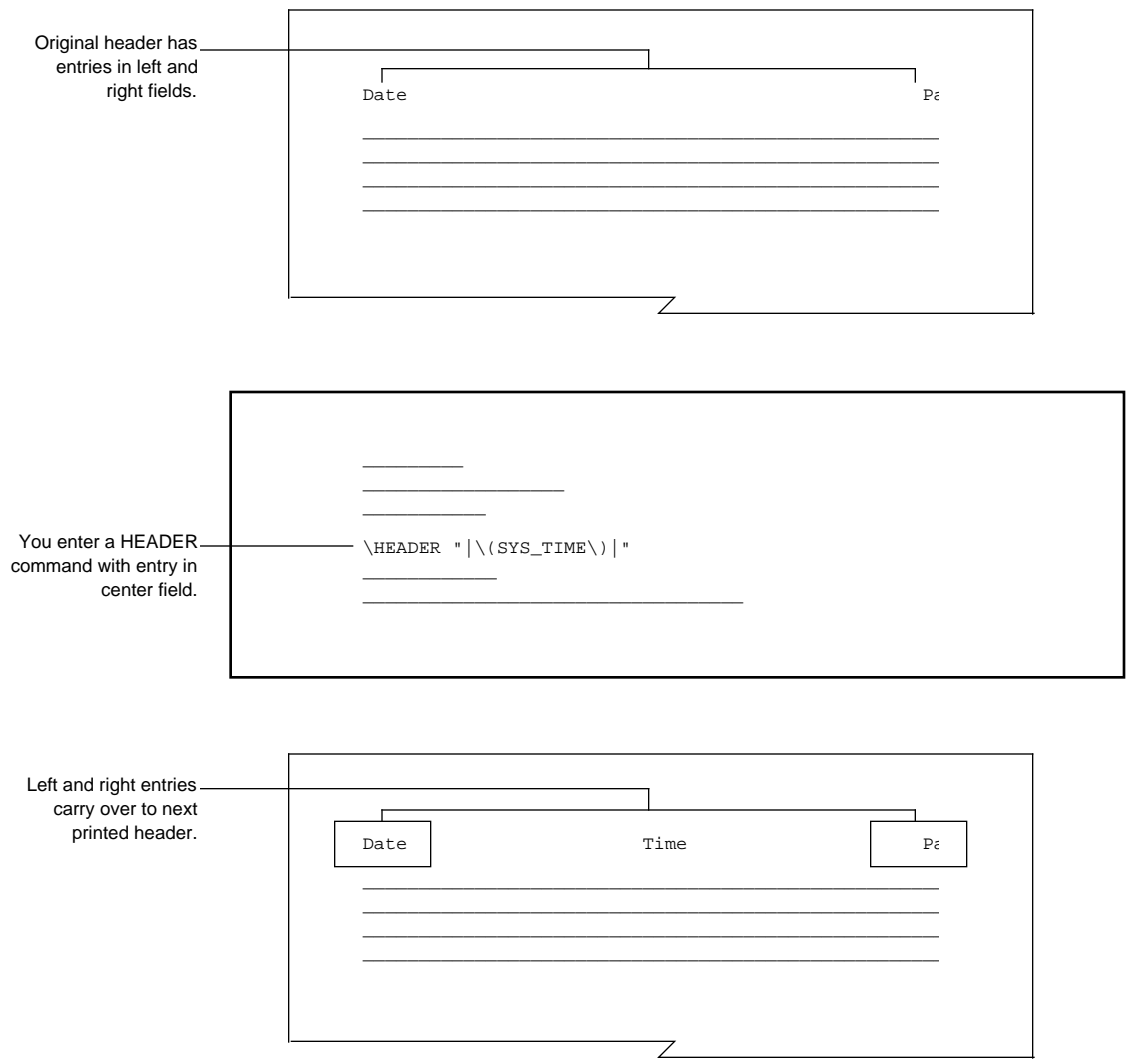
---

**Note** Remember, only headers or footers entered before text on the first page take effect immediately. Headers or footers entered on succeeding pages take effect on the next page. Changes in header or footer titles, therefore, begin on the first page following the new entry, not on the page where the command appears. You can override this delayed appearance by using the **NOW** command, which is discussed later in this section.

---



Figure 10-4. Titles Carry Over in Headers and Footers



## Adding Headers and Footers

---

Including Page Numbers, Date, and Time

---

**Including Page Numbers, Date, and Time** You can put a variety of information into header and footer titles. Title contents might include document names, descriptive comments, or even legal notices such as copyrights. Most documents also include page numbers in either the header or footer, and many documents also contain the time and date when the document is printed.

**Displaying Page Numbers** Use the pound symbol (#) to include a page number in a header or footer. Enter this symbol in the title field where you want to display the page numbers. TFORM keeps track of the pages and automatically numbers them sequentially. The following command puts the page number in the middle of the bottom of each page:

```
\FOOTER " | # | "
```

You can combine the pound sign with other text or numbers in a field. For example, to include the word "page" or the chapter number in page numbering, enter one of these commands:

```
\FOOTER " 9-# | "  
\HEADER " Page-# "
```

---

**Note** You can turn page numbering on and off, or you can begin numbering with a different number than the one currently stored in the page counter. See the PAGE command in the *PS TEXT FORMAT Reference Manual* for more information.

---



**Displaying the Date** TFORM automatically keeps track of the current date—called the *system date*—and stores this information in a *variable* named SYS\_DATE. By using the TODAY command with SYS\_DATE, you can print the current date in headers or footers.

There are several ways to display the date. The most common way is called the TODAY SHORT USA *date format*. This format uses numbers to represent the month, day, and year. The numbers are separated by slashes. It is printed like this:

```
4/8/87  
12/15/88
```

You might, however, want to use the TODAY LONG USA format if you prefer to print out your date using words. When printed, the format looks like this:

```
April 24, 1987  
December 25, 1987
```

---

**Note** For a complete listing of the date formats available in TFORM, see the *PS TEXT FORMAT Reference Manual*.

---

To put a date in your header or footer, first enter the TODAY command. Then enter SYS\_DATE in the location where you want the date to appear. When using SYS\_DATE, begin with a backslash followed by an opening parentheses, enter SYS\_DATE, and end with a backslash followed by the closing parentheses:

```
\(SYS_DATE\)
```

Figure 10-7 shows two edit files. The first file uses the TODAY SHORT USA format to put the date in a footer. The second file uses the TODAY LONG USA format and places the date in a header. Figure 10-8 illustrates the resulting printed header and footer.

Figure 10-7. Entering the Date in a Header or Footer

The TODAY command determines the format of the date.

```

\TODAY LONG USA
\HEADER "Marketing Memo|\(SYS_DATE\)|Please Read"
    
```

The TODAY command must precede the header or footer command containing SYS\_DATE.

```

\TODAY SHORT USA
\Footer "\(SYS_DATE\)|INTERNAL USE"
    
```

Figure 10-8. Displaying the Date in a Header or Footer

Marketing Memo	June 9, 1987	Please Read
/		
6/9/87		INTERNAL USE

Current date replaces SYS\_DATE in the printed document.

**Note** As long as you have previously entered the TODAY command in your edit file, you can include the date anywhere in the text of your document, as well as in headers and footers. Simply enter SYS\_DATE in the text. TFORM substitutes the current date for SYS\_DATE each time it prints the document.

## Adding Headers and Footers

---

### Displaying the Time

**Displaying the Time** TFORM automatically keeps track of the current time—called the *system time*—and stores this information in a variable named `SYS_TIME`. By using the `TIME` command with `SYS_TIME`, you can print the time in headers or footers.

To put the time in your document, first enter the `TIME` command. Then enter `SYS_TIME` in the location in your header or footer where you want the time to appear. When you enter `SYS_TIME`, be sure to enclose it in parentheses and begin and end with backslashes:

```
\(SYS_TIME\)
```

Figure 10-9 shows two edit files. The first file contains the commands used to include both the time and the date in a header. The second file puts the time in a footer. Figure 10-10 shows the printed results of these two files.

---

**Figure 10-9. Entering the Time in a Header or Footer**

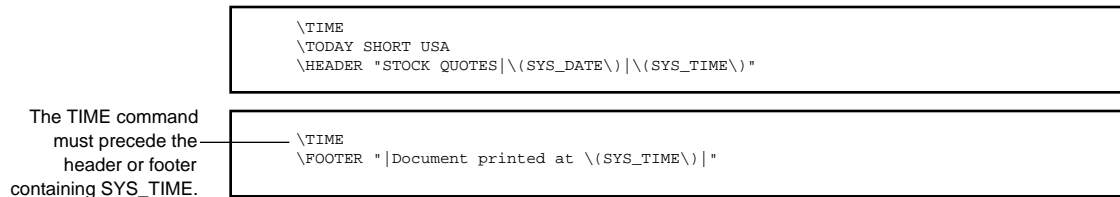
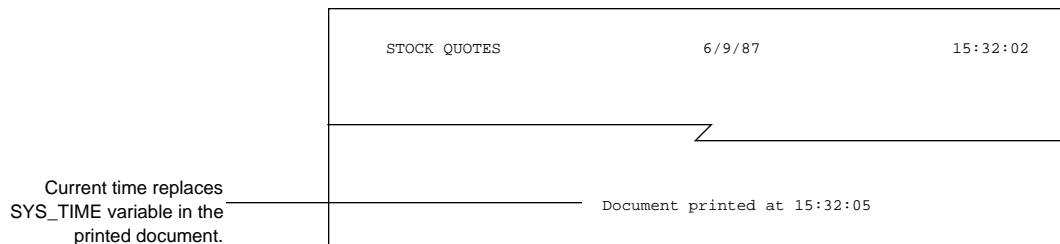


Figure 10-10. Displaying the Time in a Header or Footer



**Note** As long as you have previously entered the TIME command in your edit file, you can include the time anywhere in the text of your document, as well as in headers and footers. Simply enter SYS\_TIME in the text. TFORM substitutes the current time for SYS\_TIME each time it prints the document.

## Adding Headers and Footers

### Using Multiple Headers and Footers

#### Using Multiple Headers and Footers

Each page of a document can contain up to ten headers and ten footers. While you probably won't need this many headers and footers on a page, you might want to divide long headers and footers into multiple lines. Shorter headers and footers are both clearer and more concise.

**Note** You can enter the fields for a single header or footer title on separate lines, as explained earlier in this section in "Entering Titles." Multiple headers and footers means headers and footers are stacked on the page, each containing an individual title.

To enter multiple headers or footers, put a number after the HEADER or FOOTER command. The following command prints a two-line header that appears at the top left of the page:

Command	<pre>\HEADER 1 "Minutes of Production Meeting" \HEADER 2 "For Internal Use Only"</pre>
Sequential number of header or footer	

You don't have to number multiple headers or footers consecutively; TFORM always prints them in numeric order. For example, if you number headers 1 and 3, TFORM prints 3 right under the first header.

Headers and footers are always single spaced, no matter what spacing you have between the lines of text in your document.



Figure 10-11 shows you an edit file containing multiple line headers and footers. Figure 10-12 shows the printed results.

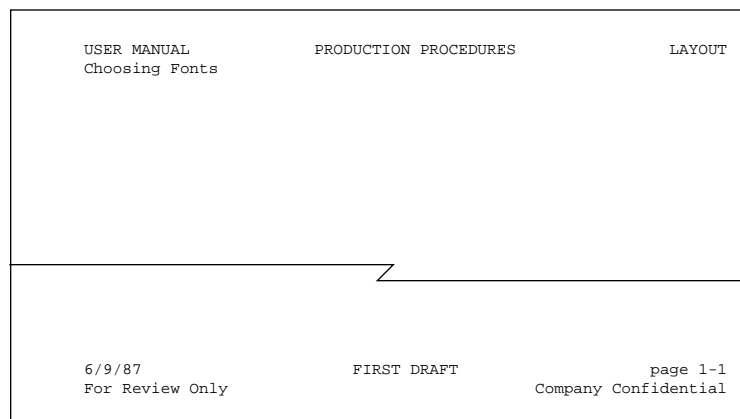
Figure 10-11. Using Multiple Headers and Footers

```
\TODAY SHORT USA
\HEADER 1 "USER MANUAL|PRODUCTION PROCEDURES|LAYOUT"
\HEADER 2 "Choosing Fonts"
\Footer 1 "\{SYS_DATE\}|FIRST DRAFT|page 1-#"
\Footer 2 "For Review Only||Company Confidential"
```

Headers and footers are printed in numeric order.

Each header or footer contains a unique title.

Figure 10-12. Printed Multiple Headers and Footers



## Adding Headers and Footers

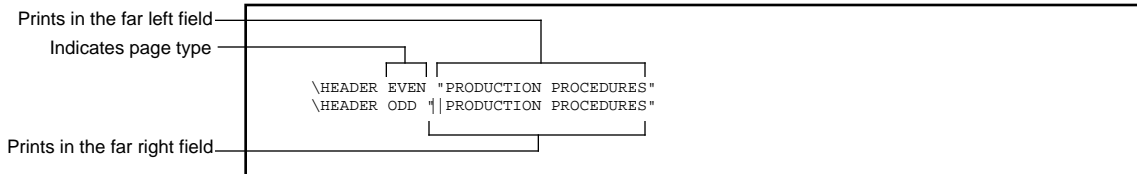
### Putting Headers and Footers on Specific Pages

#### Putting Headers and Footers on Specific Pages

By default, headers and footers appear on all pages in the same location. You can specify headers or footers for alternating pages. You can also tell TFORM to begin headers and footers on a particular page.

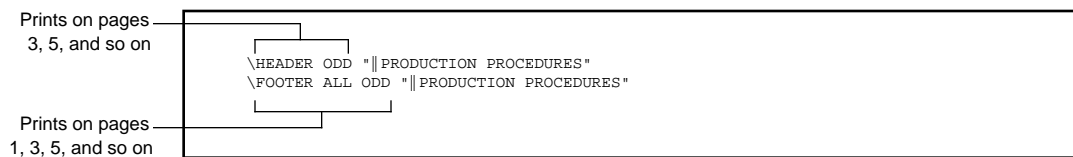
#### Printing on Even and Odd Pages

Often you want the same title for headers or footers, but you want the location to alternate from page to page. For example, in documents printed on both sides of a sheet, it's easier to read headers or footers on odd-numbered pages when they appear on the far right. Even-numbered headers and footers read better on the far left. To print alternating headers and footers, include EVEN and ODD in the HEADER or FOOTER command. You must also indicate any change in title location by putting the title in the correct field. For example:



Of course, you are not limited to having the same text on both even and odd pages, or to using a single field in the title. The addition of EVEN or ODD to the HEADER or FOOTER command simply tells TFORM that all even pages have one particular header or footer, and all odd pages have another.

When you include ODD in the command, TFORM assumes that you don't want the header or footer to appear on the first page of your document (page number 1). TFORM begins printing the header or footer on page 3, then on page 5, page 7, and so on. If you want the header or footer on page 1 also, include ALL in the command:



**Note** You don't need to include ALL in an even-paged header or footer because TFORM automatically begins printing it on page 2.

Figure 10-13 provides two examples of edit files using alternating headers and footers. The even and odd pages resulting from each set of commands appear in Figure 10-14.

## Adding Headers and Footers

### Putting Headers and Footers on Specific Pages

Figure 10-13. Entering Alternating Headers and Footers

**Example 1: Monthly Financial Review**

Text stays the same; the location changes on even and odd pages.

This footer prints on all pages.

```
\TODAY SHORT USA
\HEADER 1 ALL ODD "||MONTHLY FINANCIAL REVIEW"
\HEADER 2 ALL ODD "||\(\SYS_DATE\)"
\HEADER 1 EVEN "MONTHLY FINANCIAL REVIEW"
\HEADER 2 EVEN "\SYS_DATE\"
\Footer "COMPANY CONFIDENTIAL||PAGE #"
\SET JOIN ON

With this document, the Finance Department begins a new
policy of issuing monthly status reports. We hope you will
find this information useful in monitoring your current
budgets and in the preparation of future budget requests.

\NEW

The first section of this report reviews the advertising revenue
and circulation figures for each magazine.
```

This two-line header prints on odd pages, including page 1.

This two-line header prints on even pages.

**Example 2: Editing Procedures Manual**

This two-line header prints on odd pages, including page 1.

This footer prints on odd pages.

```
\TODAY SHORT USA
\HEADER 1 ALL ODD "||EDITING PROCEDURES MANUAL"
\HEADER 2 ALL ODD "||REVISED:\(\SYS_DATE\)"
\HEADER 1 EVEN "CHECKING GRAMMAR"
\HEADER 2 EVEN "CORRECT USE OF COMMAS"
\Footer ALL ODD " |PAGE 4-#|FINAL DRAFT"
\Footer EVEN "COMPANY CONFIDENTIAL|PAGE 4-#|"
\SET JOIN ON

The comma signals the reader that what follows is the main
clause of the sentence. When commas are omitted, you run
the risk that the reader may not pause as you intended,
either misreading the sentence or having to start again from
the beginning to get the meaning straight.

\NEW

Commas separate words, so you want to be sure not to separate
words that function together as a unit, such as the subject
and verb of a sentence.
```

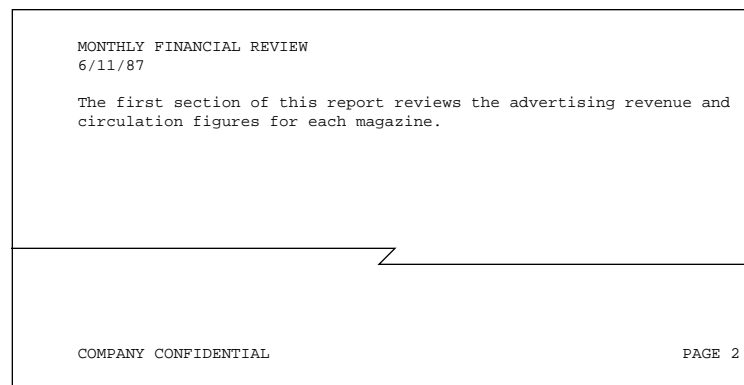
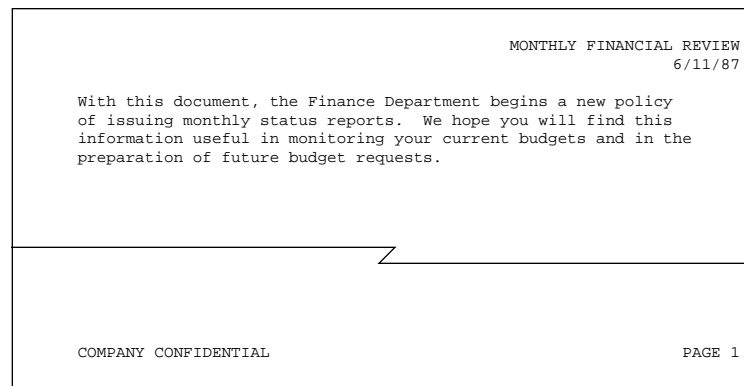
This two-line header prints on even pages.

This footer prints on even pages.

## Adding Headers and Footers

### Putting Headers and Footers on Specific Pages

Figure 10-14. Displaying Alternating Headers and Footers (Page 1 of 2)



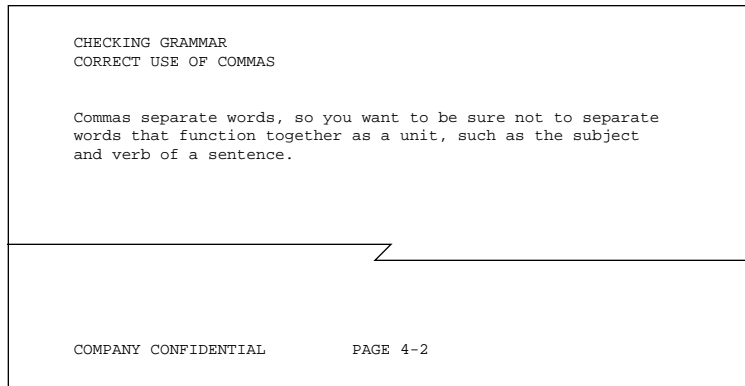
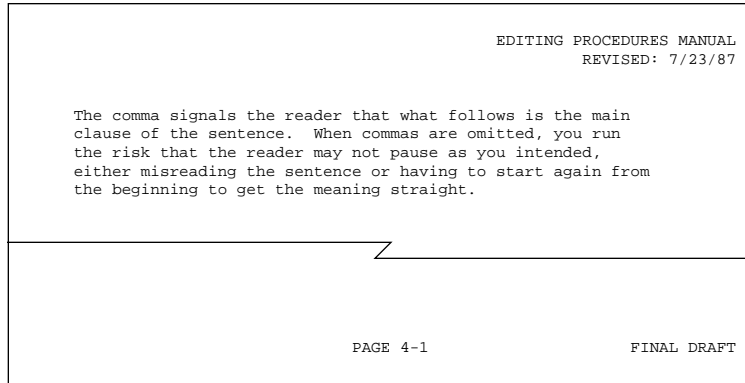
## Adding Headers and Footers

---

### Putting Headers and Footers on Specific Pages

---

**Figure 10-14. Displaying Alternating Headers and Footers (Page 2 of 2)**



**Printing on the First Page** You can tell TFORM to print a special header or footer on the first page of a document. To do this, include **FIRST** in the **HEADER** or **FOOTER** command. The following commands print this header and footer only once, on page number 1:

```
\HEADER FIRST " |PRELIMINARY QUARTERLY RESULTS| "
\FOOTER FIRST " |FOR INTERNAL DISTRIBUTION ONLY| "
```

Note that **FIRST** indicates page number 1. If the first page of the document is page 3, this header won't print.

---

**Note** **FIRST** prints the header or footer only on page 1. If you want to print a header or footer on all odd pages, including page 1, use the **HEADER ALL ODD** or **FOOTER ALL ODD** command.

---

**Printing on the Current Page** Headers or footers that you enter at the beginning of a file, or before the start of the first page, appear on all pages. If you enter subsequent headers or footers in your document, TFORM starts printing them on the page after the one you enter them on. If you include **NOW** in the command, the header or footer starts printing on the current page. The command looks like this:

```
\FOOTER NOW " |INSERT THIS DRAWING AFTER PAGE 14| "
\HEADER NOW " |THIS PAGE LEFT INTENTIONALLY BLANK| "
```

These headers and footers appear on the current page and on all following pages.

---

**Note** Place the **HEADER NOW** or **FOOTER NOW** command immediately after the **NEW** command that begins the current page.

---

**Printing After a Specified Page Number** TFORM can start adding page numbers to a header or footer with a page number other than 1. To insert such a number, use the **PAGE** command with the **HEADER** or **FOOTER** command. See the *PS TEXT FORMAT Reference Manual* for more information.

## Adding Headers and Footers

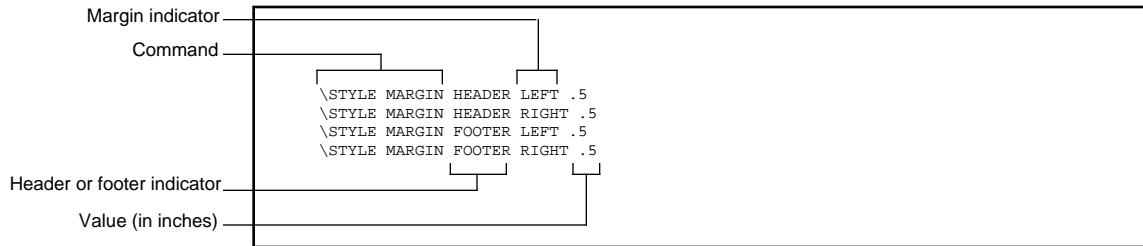
### Setting Margins for Headers and Footers

#### Setting Margins for Headers and Footers

TFORM uses the same left and right margin settings for headers and footers that you set for your text margins. And, while headers and footers don't have top and bottom margins, TFORM does maintain a specific gap between the end of the last header and the first line of text, and between the last line of text and the beginning of the first footer. You can change any of these settings.

#### Entering Side Margins

You use the `STYLE MARGIN HEADER` command to alter header or footer margins. Like setting text margins, you indicate which margin you want to change by including `LEFT` or `RIGHT` in the command. You must also enter the size of the new margin in inches. You can use other units of measurement if you wish. To move in the headers and footers one-half inch from the edges of the paper, enter these commands:



If you enter these commands without any value, TFORM resets the default header and footer margins, which are .75 inch.

Instead of a number, you can increase the margins using relative values. For example, to add one inch to the current left or right margins, enter a number with a plus sign. To decrease the margins from their current setting, put a minus sign before the number:

```
\STYLE MARGIN HEADER LEFT +1
\STYLE MARGIN FOOTER RIGHT -1
```

**Note** As with any `STYLE` command, `STYLE MARGIN HEADER` or `STYLE MARGIN FOOTER` must follow a `NEW` command.



Figure 10-15 shows the command for making the header margin different from the text margins. Figure 10-16 shows the resulting printed document.

Figure 10-15. Changing Header Margins

Moves the header margin 1.5 inches toward the left

```
\NEW
\TODAY SHORT USA
\STYLE MARGIN HEADER LEFT - 1.5
\HEADER 1 "MONTHLY FINANCIAL REVIEW"
\HEADER 2 "\{SYS_DATE\}"
\SET JOIN ON
```

The text is printed at the original margin settings.

With this document, the Finance Department begins a new policy of issuing monthly status reports. We hope you will find this information useful in monitoring your current budgets and in the preparation of future budget requests.

Figure 10-16. Displaying Different Header and Text Margins

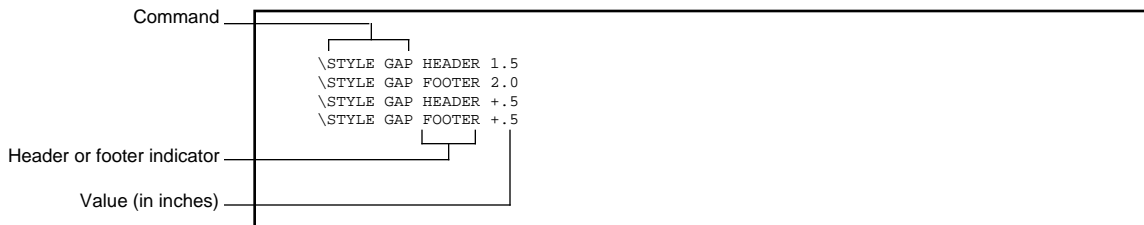
MONTHLY FINANCIAL REVIEW  
6/11/87

With this document, the Finance Department begins a new policy of issuing monthly status reports. We hope you will find this information useful in monitoring your current budgets and in the preparation of future budget requests.

## Adding Headers and Footers

### Changing the Gap

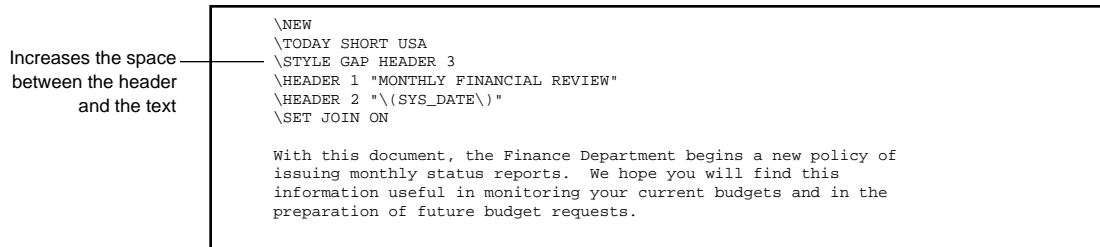
**Changing the Gap** If a page has a header or a footer, TFORM automatically leaves an inch of space between the text and the headers or footers. This space is called the *header gap* or the *footer gap*. To change this gap, you use the STYLE GAP HEADER and STYLE GAP FOOTER commands. You can enter an absolute value for the size of the gap in inches, or you can add or subtract from the current setting by preceding the value with a plus sign or a minus sign. Here are some examples of this command:



**Note** As with any STYLE command, STYLE GAP HEADER or STYLE GAP FOOTER commands must follow a NEW command.

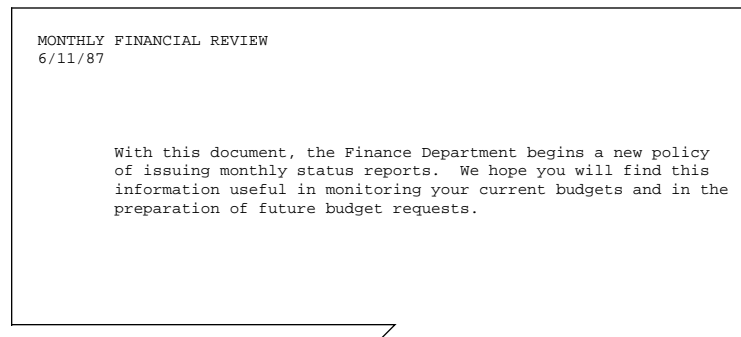
Figure 10-17 shows the command used to increase the space between the text and the header. Figure 10-18 shows how the document looks with a bigger gap.

**Figure 10-17. Changing the Header Gap**



---

**Figure 10-18. Results of Increasing the Header Gap**



## Adding Headers and Footers

---

### Summary

---

**Summary** This section told you how to add footers and headers to your printed document. Table 10-1 presents a quick summary of the commands discussed in this section.

---

**Table 10-1. Commands Used to Add Headers and Footers**

To Control	Use This Command
Titles	HEADER "     " FOOTER "     "
Page Number Display	HEADER "#" FOOTER "#"
Date Display	TODAY + \(\SYS_DATE\)
Time Display	TIME + \(\SYS_TIME\)
Multiple Headers/Footers	HEADER 1          FOOTER 1 HEADER 2          FOOTER 2
Page Location	HEADER FIRST      FOOTER FIRST HEADER ODD        FOOTER ODD HEADER EVEN       FOOTER EVEN HEADER NOW        FOOTER NOW
Margins	STYLE MARGIN HEADER STYLE MARGIN FOOTER
Gap	STYLE GAP HEADER STYLE GAP FOOTER

---

---

# 11 Using Special TFORM Features

---

**Overview** This section describes the commands that allow you to use some of the presentation and formatting capabilities of TFORM. Using these features, you can indicate key points in your document and emphasize information so that it is easily understood and attractively displayed. TFORM special features allow you to do the following:

- Change the appearance of, or style, your text
- Display text in lists
- Draw boxes
- Specify level headings
- Create tables of contents
- Produce form letters
- Add comments to your edit file

None of these features are in effect by default. You must enter commands to use the capabilities discussed in this section.

Figure 11-1 shows you a printed document formatted using some of the TFORM commands covered in this section.

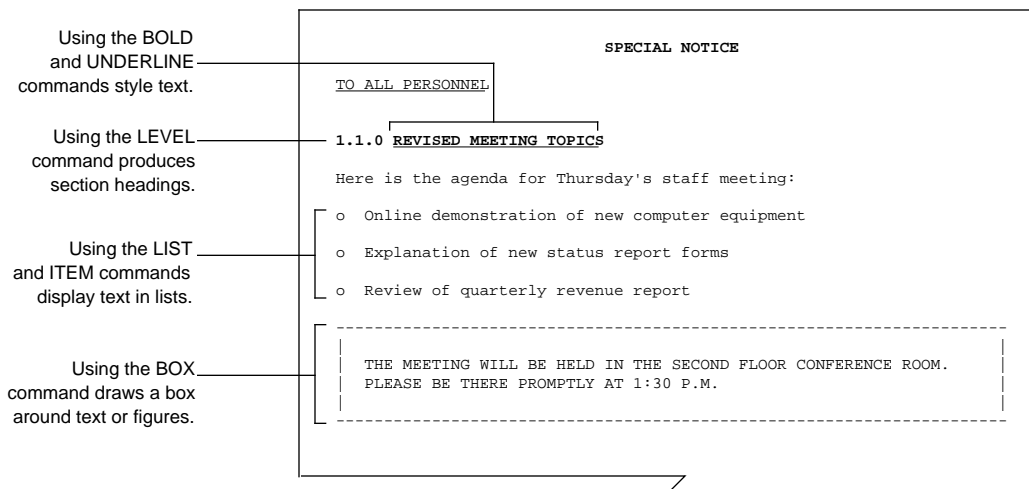
## Using Special TFORM Features

---

### Overview

---

Figure 11-1. Using TFORM Commands for Special Effects

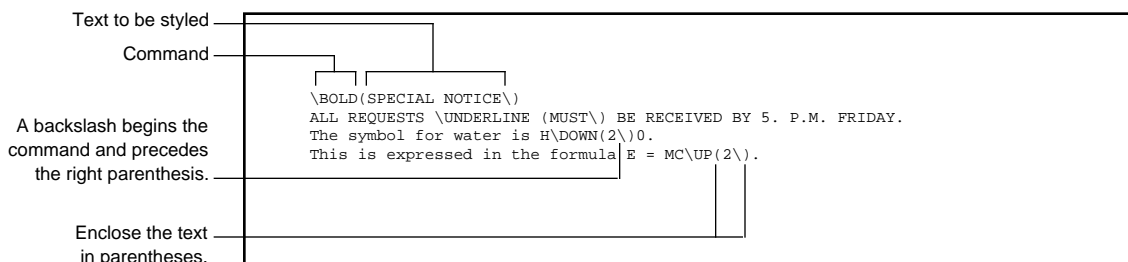


**Styling Text** Normally, all the text in your edit file is identical in the printed version of your document. By using TFORM *styling* commands, you can emphasize key text concepts and vary text appearance. The most commonly used styling capabilities of TFORM are:

- Boldface type
- Underlining
- Subscripts and superscripts

**Note** The availability of styling features in your printed document depends on the type of printer that you use. For example, your printer might not be able to print superscripts and subscripts. Check with your IMT contact for information about your printer's capabilities. When you send your formatted document to the terminal screen, some of the styling features either do not appear or might look slightly different. The text will be correct in the printed version of your document.

**Entering Styling Commands** You enter all of the styling commands in the same way. First, enter the command. Follow the command by a left parenthesis, the text you want styled, a backslash, and then the closing right parenthesis. Here are some examples of styling commands:



This is how the text looks when printed:

**SPECIAL NOTICE**

The symbol of water is H<sub>2</sub>O.

This is expressed in the formula  $E = MC^2$ .

Unlike other TFORM commands, styling commands need not start in the first column of a line. They can appear in any location within the text. The command must come immediately before the text you want to emphasize. Like other TFORM commands, you can use three-letter abbreviations for the commands.

**Using Styling Commands**

You can style several lines of text at once. Place the styling command at the beginning of the first line of text. After you've entered all the text, end the command with the backslash and a right parenthesis. For example:

```
\BOLD(The deadline for submissions is the third  
Friday of the month. Articles submitted after this  
time will not appear in the next issue of the  
magazine.\)
```

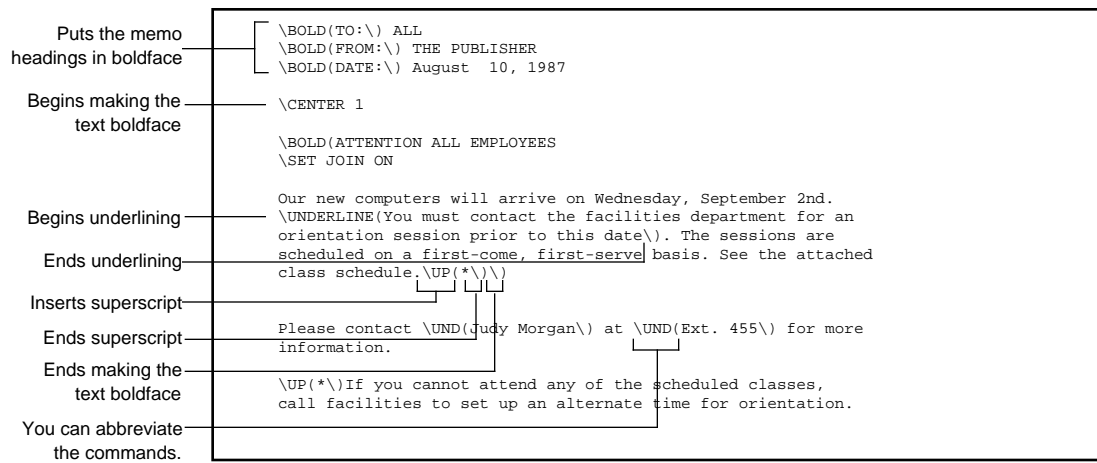
You can also mix and match styling effects by putting two or more commands in effect for the same text. For example, you can combine bolding and underlining or insert superscripts in bold type. When you put one command inside another command, you are *nesting* the commands. Each styling command must have its own set of backslashes and parentheses:

```
\BOLD(\UNDERLINE(Late Submissions Will Not Be  
Accepted!\)\)
```



Figure 11-2 shows an edit file containing styling commands. The printed results are shown in Figure 11-3.

Figure 11-2. Styling Text



---

Figure 11-3. How Styled Text Looks When Printed

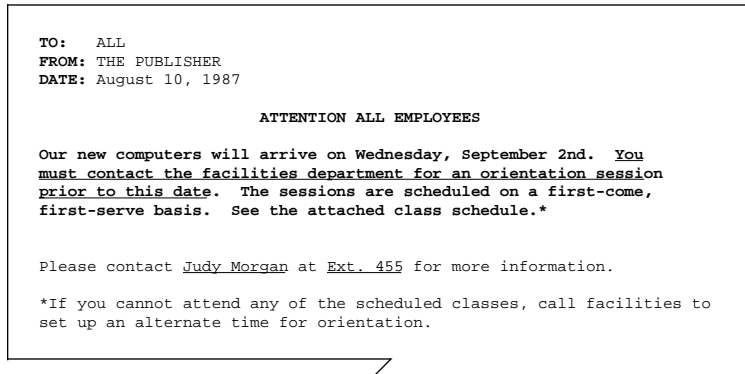
```
TO: ALL
FROM: THE PUBLISHER
DATE: August 10, 1987

ATTENTION ALL EMPLOYEES

Our new computers will arrive on Wednesday, September 2nd. You
must contact the facilities department for an orientation session
prior to this date. The sessions are scheduled on a first-come,
first-serve basis. See the attached class schedule.*

Please contact Judy Morgan at Ext. 455 for more information.

*If you cannot attend any of the scheduled classes, call facilities to
set up an alternate time for orientation.
```



---

**Note** You can override the change in line spacing when using the superscript and subscript commands. And, TFORM has a few more styling features, such as overstrike and shadow type. See the *PS TEXT FORMAT Reference Manual* for more information on these commands.

---

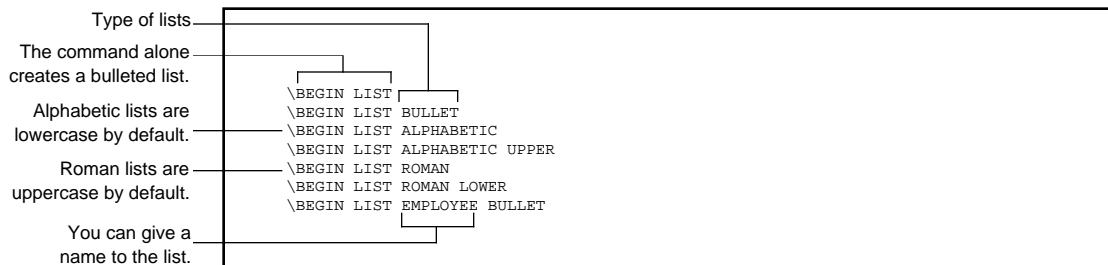
**Making Lists** TFORM has several different types of lists you can use to present text. A *list* can be any series of entries—such as words, sentences, or paragraphs—that you put together in a specific order. The most commonly used types of TFORM lists are:

- Bulleted
- Numeric
- Alphabetic
- Roman

See the *PS TEXT FORMAT Reference Manual* for a complete description of list types and formatting variations available for TFORM lists.

**Note** You must have SET JOIN ON for TFORM to format your lists correctly.

**Starting the List** No matter which type of list you create, the commands you use are the same. You start a list with the `BEGIN LIST` command. If you want to, you can give the list a name containing up to 32 characters. You must indicate the kind of list immediately after the command. The `BEGIN LIST` by itself creates a bulleted list. For example:



## Using Special TFORM Features

### Entering Items in a List

**Entering Items in a List** You enter the text you want to include in a list under the `BEGIN LIST` command. You must precede each entry with the `ITEM` command, which you enter alone on the line above the text. For example:

Begins a bulleted list	—	<code>\BEGIN LIST BULLET</code>
Indicates an item in the list	—	<code>\ITEM</code>
The text of the item	—	<code>Project Identification</code>

```
\ITEM
Number of Hours This Week
\ITEM
Total Number of Hours to Date
```

You enter all list items in this same way, no matter which type of list you're creating. When printed, each item in a list is separated from the next by a single line.

**Ending the List** All lists end with the `END LIST` command. You don't need to indicate the type of list in this command. Enter the `END LIST` command on the line below the text of the last item. For example:

```
\BEGIN LIST BULLET
\ITEM
Project Identification
\ITEM
Number of Hours This Week
\ITEM
Total Number of Hours to Date
\END LIST
```

**Nesting Lists** Just as you can nest styling commands, you can also create nested lists, which is one list inside of another. The lists can be of varying types and lengths. When you have a nested list, you must include a `BEGIN LIST` and an `END LIST` command for each individual list. To help you keep track of nested lists, you might want to give each list a name, and include the name in the `END LIST` command. All list entries follow a separate `ITEM` command.

Figure 11-4 shows you an edit file containing a bulleted list with a nested numeric list. Figure 11-5 presents the lists in their printed form.

Figure 11-4. Creating Lists

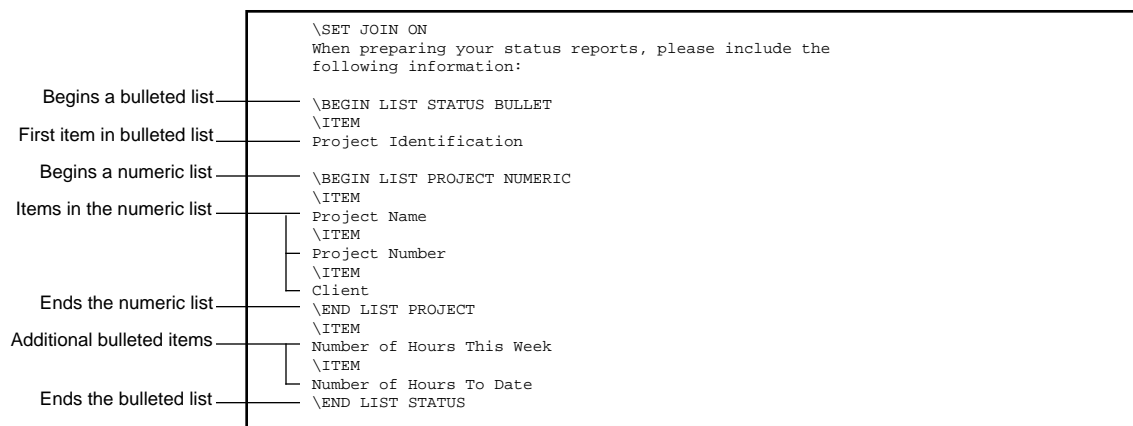
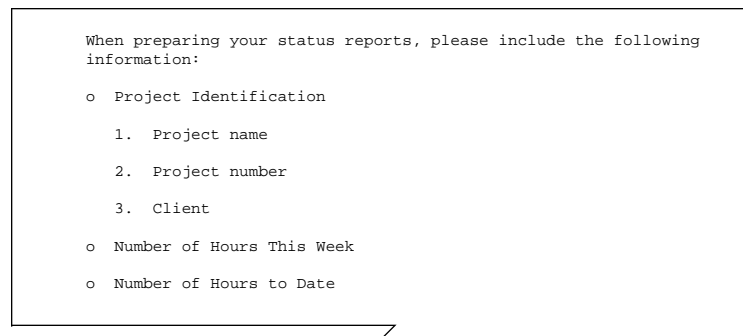


Figure 11-5. How Lists Look When Printed



**Drawing Boxes** Drawing a box around text, or enclosing a figure or illustration in a box, is an effective way of drawing attention to specific information in your document. TFORM boxes can be as small as two lines or as large as several pages.

TFORM draws hyphens across the width of the page for the top and bottom of the box; the sides are drawn with vertical bars. There is a single line between the box and the first line of text, and between the last line of the text and the bottom of the box. Everything inside the box is slightly indented to provide side margins. If joining and justification are in effect, they continue within the box.

**Note** You can change many of the characteristics of boxes, including the symbol used to draw them and the margin settings within the box. In addition, you can mark large boxes with a right arrow to indicate that they continue for more than one page. See the *PS TEXT FORMAT Reference Manual* for a complete discussion of using boxes in TFORM.

You use the SET BOX command to draw boxes. To start the box, enter SET BOX ON. To end the box, enter SET BOX OFF. All of the area between these two commands is enclosed in the box. For example:

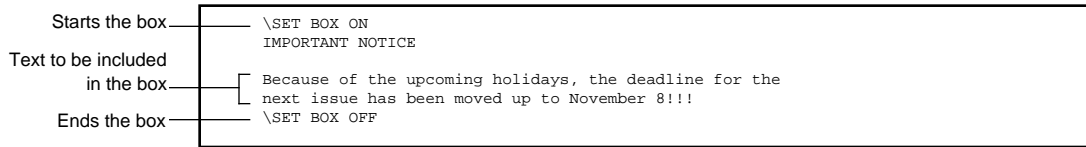


Figure 11-6 shows an edit file that uses a box to emphasize text. Figure 11-7 illustrates how the box appears in the printed document.

Figure 11-6. Drawing Boxes

```

\SET JOIN ON
The publisher requests that we begin using these new status reports
at the beginning of next month. An example of the status report
is shown below.
\SET JOIN OFF
\SET BOX ON

STATUS FOR THE WEEK OF _____

PROJECT IDENTIFICATION:
    Name _____
    Number _____
    Client _____

NUMBER OF HOURS THIS WEEK _____
NUMBER OF HOURS TO DATE _____

\SET BOX OFF
    
```

Begins drawing the box

Ends the box

Figure 11-7. How Boxes Look When Printed

The publisher requests that we begin using these new status reports at the beginning of next month. An example of the status report is shown below.

-----

STATUS FOR THE WEEK OF \_\_\_\_\_

PROJECT IDENTIFICATION:

Name \_\_\_\_\_

Number \_\_\_\_\_

Client \_\_\_\_\_

NUMBER OF HOURS THIS WEEK \_\_\_\_\_

NUMBER OF HOURS TO DATE \_\_\_\_\_

-----

TFORM puts one line between box border and first line of text.

Text within the box is indented two spaces.

TFORM puts one line between the end of text and box border.

## Using Special TFORM Features

### Defining Levels in Your Document

#### Defining Levels in Your Document

TFORM provides an easy way for you to define sections within your document by separating them with specific headings. These headings can range from chapter titles to paragraph subheadings. The different types of headings are called levels.

There are eleven possible *level headings*. Level 0 is usually used for chapter titles or other major divisions. Level 10 is rarely, if ever, used. In most cases, you'll probably use no more than three or four levels in a document.

#### Entering Level Headings

To define a level, enter the LEVEL command followed by the text you want in the heading. When creating a level-0 heading, you enter the text as a title. Just as in header and footer titles, a title consists of three fields separated by vertical bars. The title is enclosed in quotation marks. For example:

The third field contains the text.

The first field defines the section, chapter, appendix, and so on.

The second field contains an identifying number, letter, or Roman numeral.

```
\LEVEL 0 "SECTION|11|THE EDITING PROCESS"  
\LEVEL 0 "CHAPTER|14|CREATING GRAPHICS"  
\LEVEL 0 "APPENDIX|A|A SUMMARY OF BASICS"
```

For levels 1 through 10, simply enclose the text in quotation marks. The following are some examples of level entries:

```
\LEVEL 1 "Introduction"  
\LEVEL 2 "Editing for Style"  
\LEVEL 3 "Punctuation"
```

**Note** TFORM stores the text you enter in level headings as a special variable that you can reference. For example, you might want to use your level headings as headers or footers in the document. See the *PS TEXT FORMAT Reference Manual* for information on using variables.



**Printing Level Headings** When printed, the headings appear in different styles and formats, and with unique spacing settings, depending on the particular level you chose. Level 0, for example, is printed centered, with the first and second fields on one line and the third field on the line below. The text of Level-0 headings appears in uppercase boldface type, and it is underlined.

Level-3 headings, on the other hand, are printed to the far left. The text is uppercase and lowercase and is in boldface type.

All printed levels also include a number. In level 0, the number appears as part of the text of the heading. All levels below level 0 repeat this number and add additional identifying numbers. Figure 11-8 shows an edit file with several level headings. Figure 11-9 shows the printed headings.

---

**Figure 11-8. Entering Level Headings**

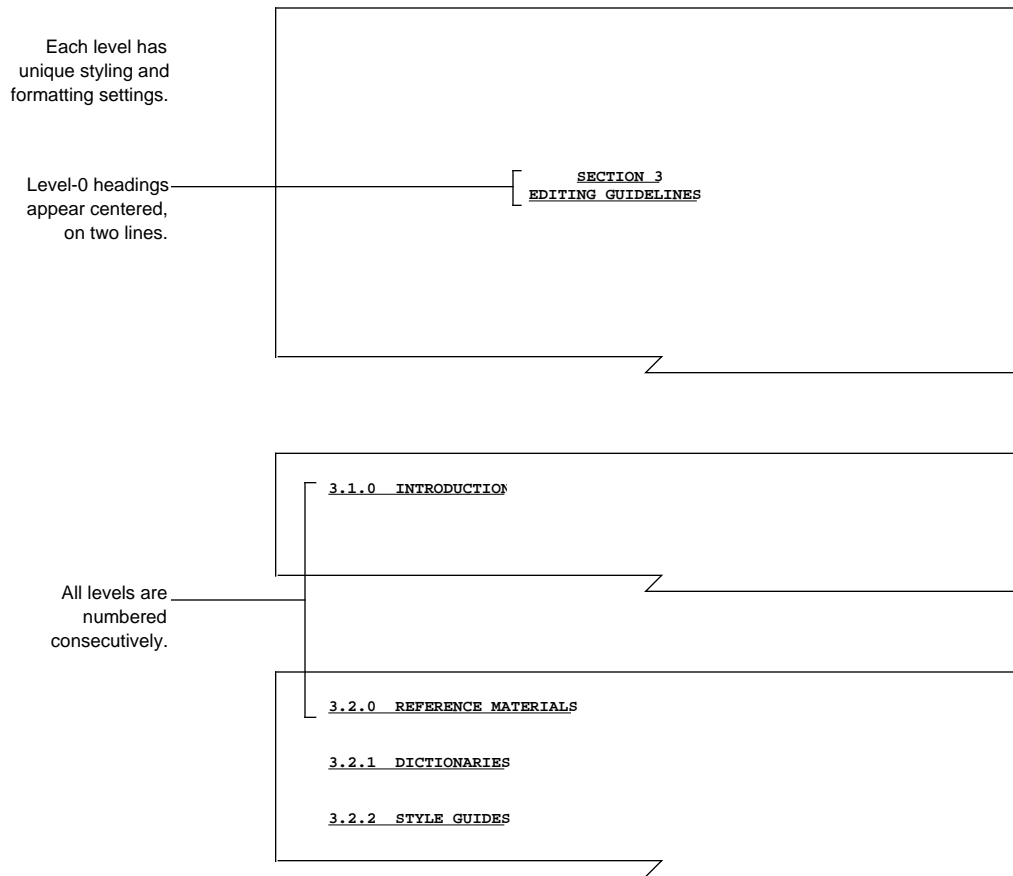
```
\LEVEL 0 "SECTION|3|EDITING GUIDELINES"  
\LEVEL 1 "Introduction"  
\LEVEL 1 "Reference Materials"  
\LEVEL 2 "Dictionaries"  
\LEVEL 2 "Style Guides"
```

## Using Special TFORM Features

### Printing Level Headings

---

Figure 11-9. Level Headings Appear in Different Styles



Level-0 headings produce a title page, with the heading alone on the page. TFORM moves to the next odd page after a LEVEL 0 command. TFORM also begins a new page for each level-1 heading. All level-2 and lower headings are printed sequentially, following the level-1 heading. TFORM moves to a new page when it reaches the next level-1 heading.

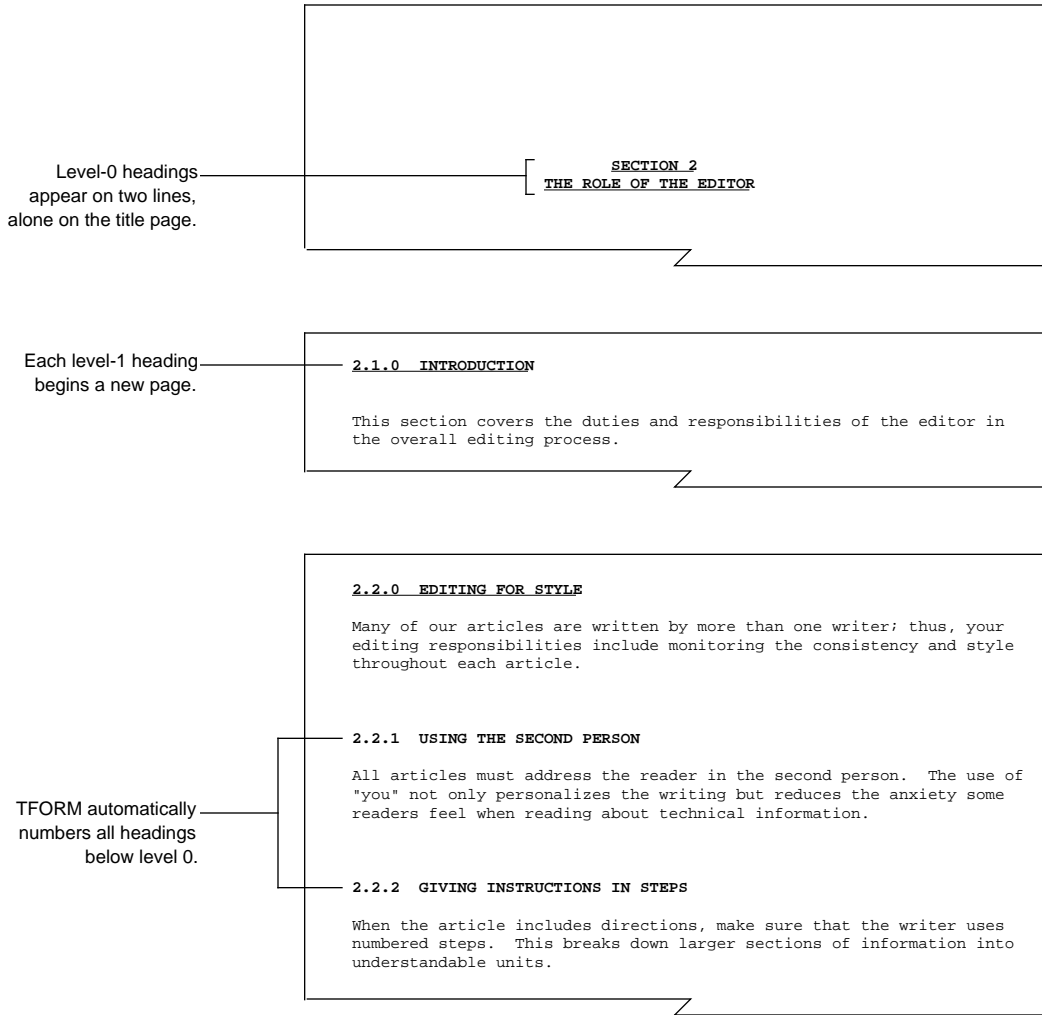
**Note** You can change the default settings for style, format, numbering, and spacing. See the *PS TEXT FORMAT Reference Manual* for complete details of the default settings and instructions for modifying level appearance.

Figure 11-10 shows an edit file which places level headings before sections of text. Figure 11-11 shows how the printed headings appear in the document and how text is grouped under level headings.

**Figure 11-10. Defining Level Headings in an Edit File**

```
\LEVEL 0 "SECTION| 2|  
THE ROLE OF THE EDITOR"  
\LEVEL 1 "Introduction"  
\SET JOIN ON  
This section covers the duties and responsibilities of  
the editor in the overall editing process.  
  
\LEVEL 1 "Editing for Style"  
  
Many of our articles are written by more than one writer; thus, your  
editing responsibilities include monitoring the consistency and style  
throughout each article.  
  
\LEVEL 2 "Using the Second Person"  
  
All articles must address the reader in the second person. The use  
of "you" not only personalizes the writing but reduces the anxiety  
some readers feel when reading technical information.  
  
\LEVEL 2 "Giving Instructions in Steps"  
  
When the article includes directions, make sure that the writer  
uses numbered steps. This breaks down larger sections of information  
into understandable units.
```

Figure 11-11. Levels Are Grouped Together



---

## Generating a Table of Contents

TFORM keeps track of your level headings. Using the LEVEL CONTENTS command, you can retrieve these headings to produce a table of contents for your document.

The LEVEL CONTENTS command is on by default. TFORM stores all level headings from level 0 through level 4. When printed, the level headings produce a table of contents with page numbers. Figure 11-12 shows the level headings in an edit file. Figure 11-13 presents the table of contents produced from those level headings.

---

**Figure 11-12. Level Headings Generate a Table of Contents**

```
\LEVEL 0 "SECTION|2|THE ROLE OF THE EDITOR"  
\LEVEL 1 "Introduction"  
\LEV@21EL 1 "Editing for Style"  
\LEVEL 2 "Using the Second Person"  
\LEVEL 2 "Giving Instructions in Steps"
```

## Using Special TFORM Features

### Generating a Table of Contents

Figure 11-13. Printed Table of Contents

The table of contents is single spaced.

Page numbers are included.

TFORM aligns level-0 headings to the left margin and indents all other headings.

All headings appear in uppercase.

SECTION 2. THE ROLE OF THE EDITOR . . . . .	3
2.1.0 INTRODUCTION . . . . .	4
2.2.0 EDITING FOR STYLE . . . . .	5
2.2.1 USING THE SECOND PERSON . . . . .	5
2.2.2 GIVING INSTRUCTIONS IN STEPS . . . . .	5

You can turn the LEVEL CONTENTS command ON and OFF if you want only parts of the document to be included in the table of contents. To stop the collection of level headings, enter LEVEL CONTENTS OFF. To begin storing headings again, enter LEVEL CONTENTS ON.

There are several ways you can modify the appearance of the table of contents. You might want to include either headings down to level 5 or only level-1 and level-2 headings. You can also control how TFORM indents and spaces the table of contents, and how page numbers appear. You can save the heading numbers and page numbers in a variable for later reference. See the *PS TEXT FORMAT Reference Manual* for more information on using the LEVEL CONTENTS command.

---

## Producing Form Letters

---

**Note** If you use TEDIT, and you have worked through the TEDIT part of this manual, you learned how to create form letters using SEARCH and REPLACE. This section discusses an alternative way to produce form letters using the same text example.

---

In a *form letter*, most of the text is the same for every letter. TFORM inserts items that change, such as names and addresses, into the letter at specific points.

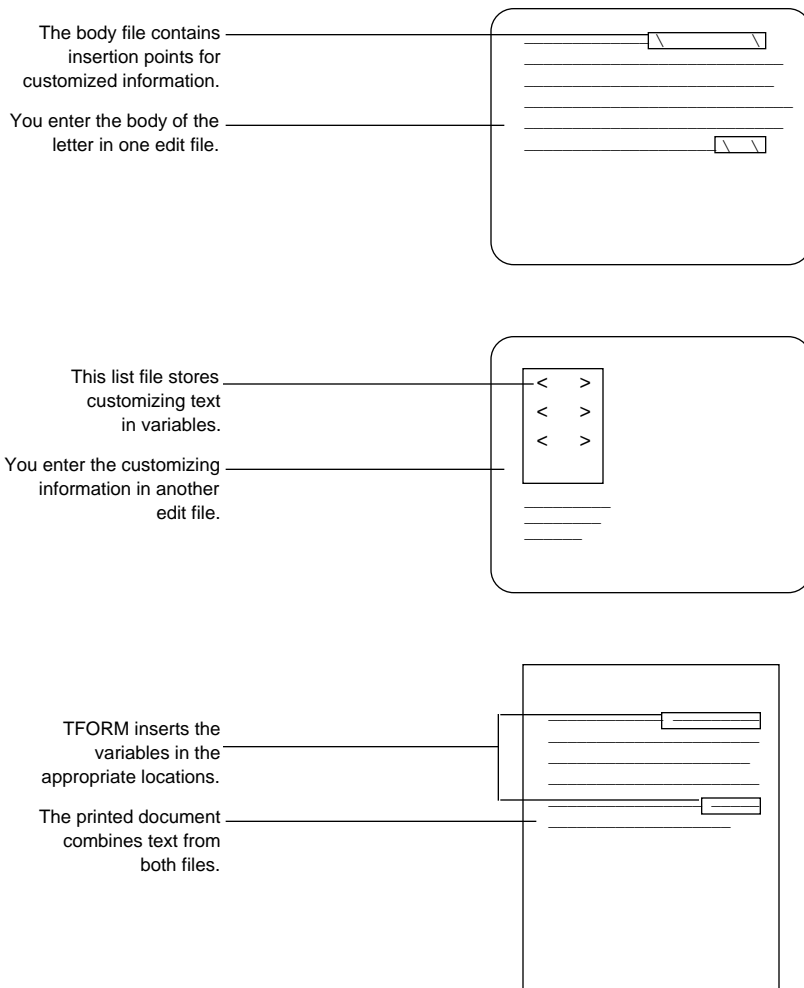
To produce form letters, you create two files. In one file, you enter the body of the letter. In the second file, you put the unique information to be inserted in only one letter. TFORM combines the information in the two files to produce the form letters. This interaction is shown in Figure 11-14.

## Using Special TFORM Features

### Producing Form Letters

---

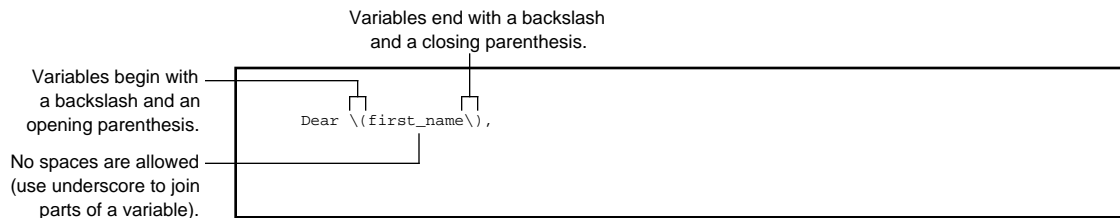
**Figure 11-14. How TFORM Produces Form Letters**





**Creating the Body of the Letter**

You store the text of your letter in a *body file*. Within this file, you indicate insertion points for the customized information you'll put in each letter. The points are identified by entering *variables* into the text. Variables follow a specific format. You can't have any blank spaces in the variable. For example:



You can put variables anywhere in your letter. Each time TFORM finds a variable, it inserts unique information.

Figure 11-15 shows the body file for a form letter containing three variables.

**Figure 11-15. The Body File for a Form Letter**

```
\NEW
\STYLE MARGIN RIGHT 1
\STYLE MARGIN LEFT 1
\ADDRESS
\SET JOIN ON

Dear \((first name)\),

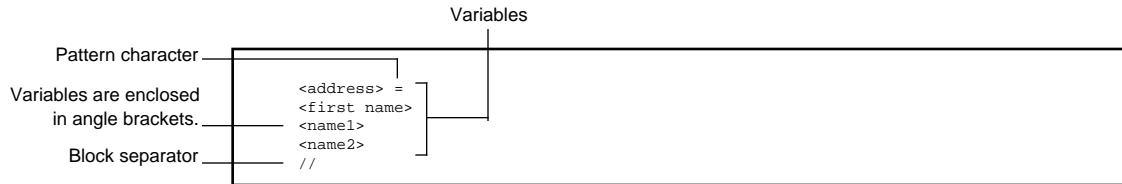
As you know, the quarterly budgeting meeting has been scheduled
for March 15. Your presentation will follow \((name1)\) and
precede \((name2)\). See the attached agenda.
```

## Using Special TFORM Features

### Creating the Variables File

**Creating the Variables File** You store the variables for the form letter in a *list file*.

Within the list file are blocks of information. The first block contains a template, which is a guide to the data. The *template block* tells TFORM the order in which the variables are stored in the file. For example, the template block for the letter in Figure 11-15 looks like this:



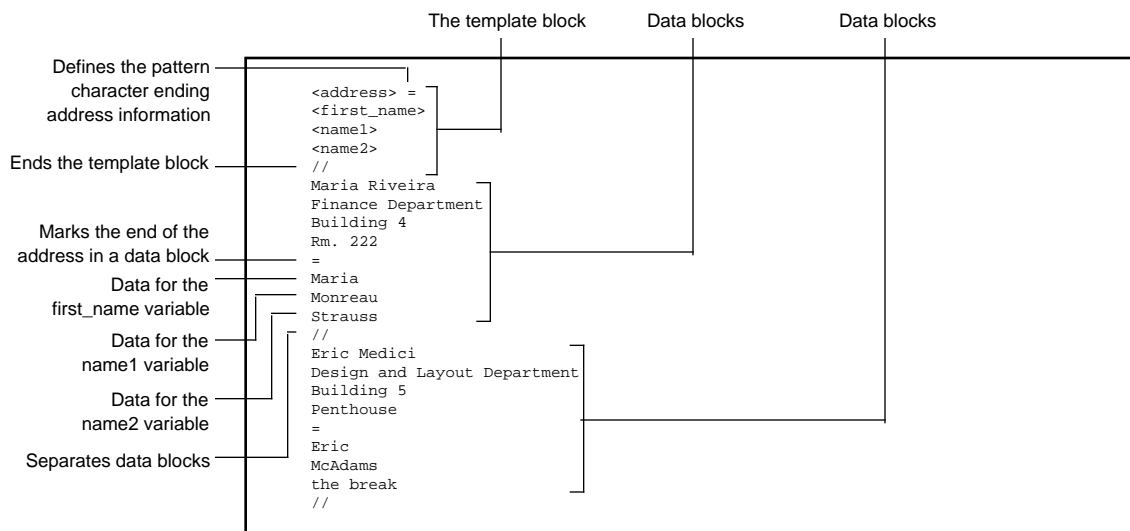
The first variable, <address>, works a little differently from the others; it is explained in the following paragraphs.

The equal sign (=) after the address variable is a *pattern character*. The pattern character tells TFORM where the address ends in each block. Enter the pattern character on the same line as the <address> variable, exactly one space after the right angle bracket. You can use any character you want for a pattern character; this example uses the equal sign. All information in the block after the pattern character is data for the other variables.

You separate the template block from the *data blocks* of information with a *block separator*. In this example, two slashes (/ /) form the block separator, which must be different from the pattern character. All the data blocks must be separated by the block separator.

Figure 11-16 shows the list file for this form letter. It contains the template block and data blocks for two letters.

Figure 11-16. The List File for a Form Letter



**Entering the Address** TFORM contains a separate ADDRESS command that you enter at the beginning of the body file immediately after any page formatting commands. For example, the beginning of your body file might look like this:

```
\NEW
\STYLE MARGIN LEFT 1
\STYLE MARGIN RIGHT 1
\ADDRESS
```

## Using Special TFORM Features

---

### Printing Form Letters

The ADDRESS command performs the following tasks:

- Tells TFORM to print the address variable in the list file (all the data from the block separator to the pattern character).
- Automatically inserts into the letter the current date, right-justified, in the TODAY LONG USA format. TFORM puts two lines between the date and the first line of the address.
- Sets up headers for subsequent pages of the letter.
- Turns on joining and justification.

**Printing Form Letters** You use the LETTER command to produce the printed form letters. This command is entered after the usual TFORM RUN command. You do not include an IN file in the command. See Section 12 of this manual to learn more about using the TFORM RUN command.

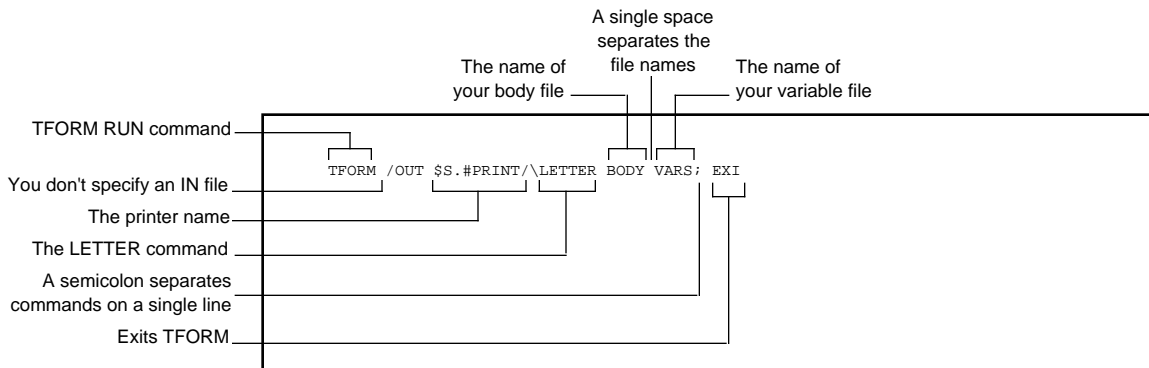
The LETTER command contains the name of your body file and the name of your variable file, separated by a single space. When you enter this command, TFORM begins printing the first letter, inserting the variables into the body text. It continues printing letters until it reaches the end of the variable file. Entering the EXIT command after the LETTER command enables you to leave TFORM.

---

**Note** If you don't include the EXIT command, TFORM returns the ? prompt when it reaches the end of the variable file, and waits for further instructions from you. You then enter EXIT or press `[CTRL]-[Y]` to leave TFORM.

---

The entire command line looks like this:



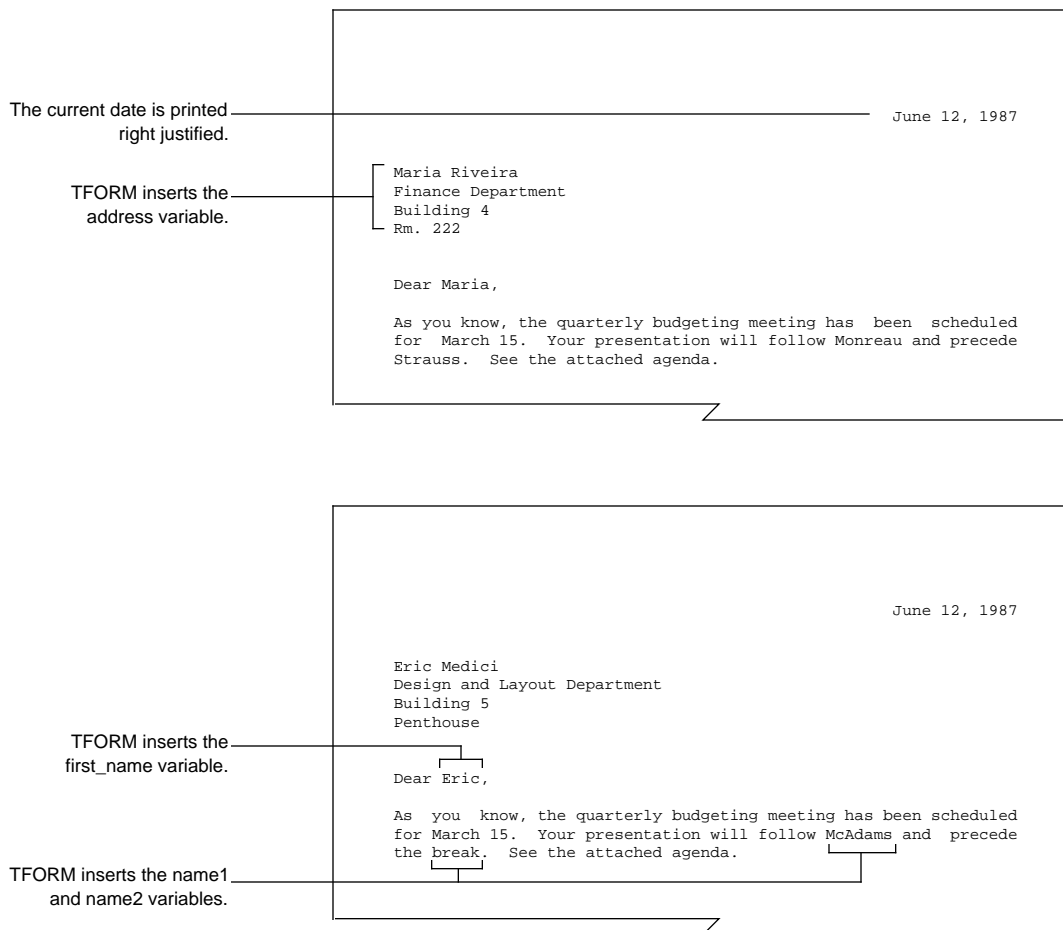
**Note** You can include options in the LETTER command that change the format of the date, or you can insert a date other than the current one. TFORM also allows you to specify top spacing or a minimum number of address lines. See the *PS TEXT FORMAT Reference Manual* for more information on using the LETTER command.

Figure 11-17 shows the printed form letters that result from the body file and the list file that appear in Figures 11-15 and 11-16.

## Using Special TFORM Features

### Printing Form Letters

Figure 11-17. Printed Form Letters



**Adding Comments to a File** TFORM provides a way for you to document the contents in an edit file. Perhaps you want to make notes on the data in the file, or to remind yourself of the way you've set up the commands. You might want to provide instructions to others who use the file. Anytime you want to include information in a file, but you don't want the information printed, you use the COMMENT command.

There are two versions of the COMMENT command:

```
\COMMENT  
\!
```

You can use either one of these versions when you're putting the comment on a line by itself. For example:

```
\COMMENT This figure will be provided by graphics.  
\! Check the page number of this reference.
```

You don't need to start a comment line at the far left of a line. You can insert comments anywhere in the file. If you're adding a comment to a line that begins with another command, you must use the COMMENT version. The exclamation point (!) version of the command can appear in either text lines or command lines:

```
\STYLE MARGIN TOP 3; COMMENT This leaves room for  
the logo  
\! You must \SET JOIN ON for this to format  
correctly.  
\ .....as explained in Chapter 4. \! Check chapter  
number
```

When you use the COMMENT version on a command line, you follow the usual rules for entering multiple commands. The backslash appears only in the first column, and semicolons separate the commands.

When you use the ! version of the command, whether in text lines or command lines, the backslash always stays with the exclamation point.

## Using Special TFORM Features

---

### Summary

---

**Summary** This section discussed some of the special features of TFORM. You use these features to emphasize information in a printed document, and to vary the appearance of the text. Table 11-1 presents a quick summary of the commands discussed in this section.

---

**Table 11-1. Commands Used for Special TFORM Features**

To Control	Use This Command
Text Styling	BOLD UNDERLINE DOWN UP
List Display	BEGIN LIST ITEM END LIST
Boxes	SET BOX ON SET BOX OFF
Level Headings	LEVEL
Table of Contents	LEVEL CONTENTS
Form Letters	LETTER ADDRESS
Comments	COMMENT !

---



---

# 12 Printing Your Document

---

**Overview** After you have entered the TFORM formatting commands into your edit file, you are ready to print out your document. If you send a file that does not contain any TFORM commands to a printer, the document is printed using the default settings. These settings are summarized in Table 7-2, which appears in Section 7.

This section discusses the printing process and includes the following topics:

- Running the TFORM program
- Sending the document to a printer
- Previewing the document on your screen
- Sending the document to another file
- Printing selected pages
- Shifting the text on the page

## Printing Your Document

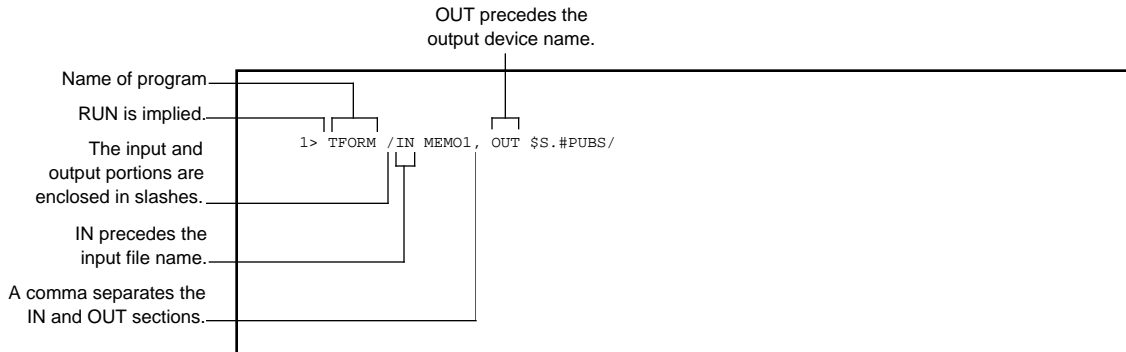
### Running TFORM

**Running TFORM** You run the TFORM program from your command interpreter prompt. To do this, you use an implied RUN command. “Implied” means you don’t actually need to enter RUN before the program name; instead, you simply enter TFORM. The command consists of three parts:

- The TFORM command entered after the command interpreter prompt
- The word IN followed by the name of the file
- The word OUT followed by the output device (the location you want to send the output)

You separate the IN information from the OUT information by a comma. The entire section is enclosed in slashes.

For example, suppose the name of your edit file is MEMO1 and the name of your output device is \$\$.#PUBS. Your command looks like this:



**Note** The RUN command has several options, such as NOWAIT and HOLD, that can be useful when you’re printing with TFORM. See the *GUARDIAN 90 Operating System Utilities Reference Manual* for a complete description of the RUN command.

---

**Sending the Document to a Printer**

In most cases, you send your edit file to a printer. To do this, enter the printer device name after OUT in the RUN command. For example, to send the MEMO1 file to a printer named \$\$.#FINAL, enter this command:

Sends document to the \$\$.#FINAL printer

```
1> TFORM /IN MEMO1, OUT $$.#FINAL/
```

You can send your document to any Tandem printer, but only the Tandem 5530 letter-quality printer can produce some TFORM special effects such as boldface, subscripts, and superscripts. When you send a file containing commands for these special features to a printer other than the 5530, the commands are ignored.

---

**Note** TFORM uses certain system parameters to define the device and printer type you're using. See the *PS TEXT FORMAT Reference Manual* for information on system parameters.

---

## Printing Your Document

### Previewing the Document on Your Screen

#### Previewing the Document on Your Screen

You can see how your formatted document looks without printing out the pages by sending the output to your terminal. This is called *previewing* your document. To preview a file, enter the TFORM command without the OUT portion. Because you haven't specified an output device, TFORM sends the output to your screen. For example, to display the MEMO1 file, enter this command:

You don't specify an OUT file to send the document to your terminal.

```
1> TFORM /IN MEMO1/
```

TFORM displays your document (including any error messages) on the screen. The entire document is sent at one time, so if you have a multiple page document, it can be hard to read the text as it scrolls. You can avoid this problem by adding the PRINT MANUALLY ON command to your TFORM command.

The PRINT MANUALLY ON command tells TFORM to display a single page of the file on your screen at one time. For example:

The backslash for additional commands follows the final slash of the TFORM command.

```
1> TFORM /IN MEMO1/ \PRINT MANUALLY ON
```

Displays the document a page at a time

The first page of text appears on the screen. At the bottom of your screen, TFORM displays this message:

```
Press RETURN to resume processing
```

After you press `RETURN`, the next page of the document appears on the screen. Using this method, you can page through the entire document.

**Sending the Document to Another File**

If you enter a file name as the output device, TFORM stores your edit file in another file. When you enter the file name, TFORM checks that the edit file name does not already exist so that it does not accidentally overwrite an edit file. To send MEMO1 to a file named STORAGE in your current subvolume, enter this command:

Sends document to a file named STORAGE

```
1> TFORM /IN MEMO1, OUT STORAGE/
```

## Printing Your Document

### Printing Selected Pages

**Printing Selected Pages** When you want to print only sections of a document, or a number of selected pages, you add the **PRINT PAGES** command to the **RUN TFORM** command. You can print a single page, a range of pages, or a series of pages. The selections can be combined in one command. The following commands show you how to print out portions of the **MEMO1** document:

Prints only page 1  
Prints pages 6  
through 12. Separate  
ranges with a slash.

```
1> TFORM /IN MEMO1, OUT $$.#FINAL/\PRINT PAGES 1
1> TFORM /IN MEMO1, OUT $$.#FINAL/\PRINT PAGES 6/12
1> TFORM /IN MEMO1, OUT $$.#FINAL/\PRINT PAGES 1, 6/12, 3, 22
```

Prints all the pages indicated.  
Separate series with commas.

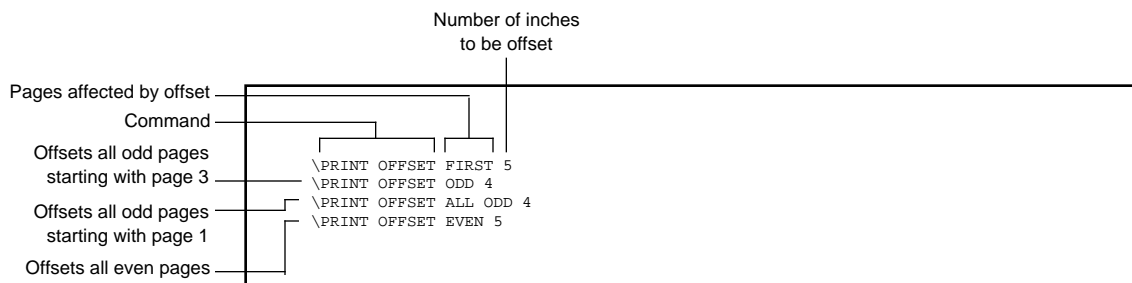
In addition to using the **PRINT PAGES** command as part of the **TFORM** command, you can also print selected pages by entering the command in your edit file. Enter **PRINT PAGES** at either the beginning of the file or immediately after a **NEW** command. Entering the command at either of these locations ensures that you print out all the pages you want because the **PRINT PAGES** command works only for pages in the file that follow the command.

### Shifting the Text on the Page

If your document is going to be bound or hole-punched, you might want to leave extra space along the edge of the page. The text is *print offset* when you shift its location on the page. Entering the PRINT OFFSET command causes TFORM to shift the text on the page to allow this extra room. The command must include the number of inches you want the text shifted.

Text on odd-numbered pages shifts to the right; text on even-numbered pages shifts to the left. You can shift the text on the first page only by using the PRINT OFFSET FIRST command. To shift text on all odd pages beginning on page three, use the PRINT OFFSET ODD command. To shift the text on all odd pages, including the first page, enter the PRINT OFFSET ALL ODD command. Use PRINT OFFSET EVEN to shift text on even pages.

Be sure you enter the command before the pages you want shifted. Here are some examples of this command:

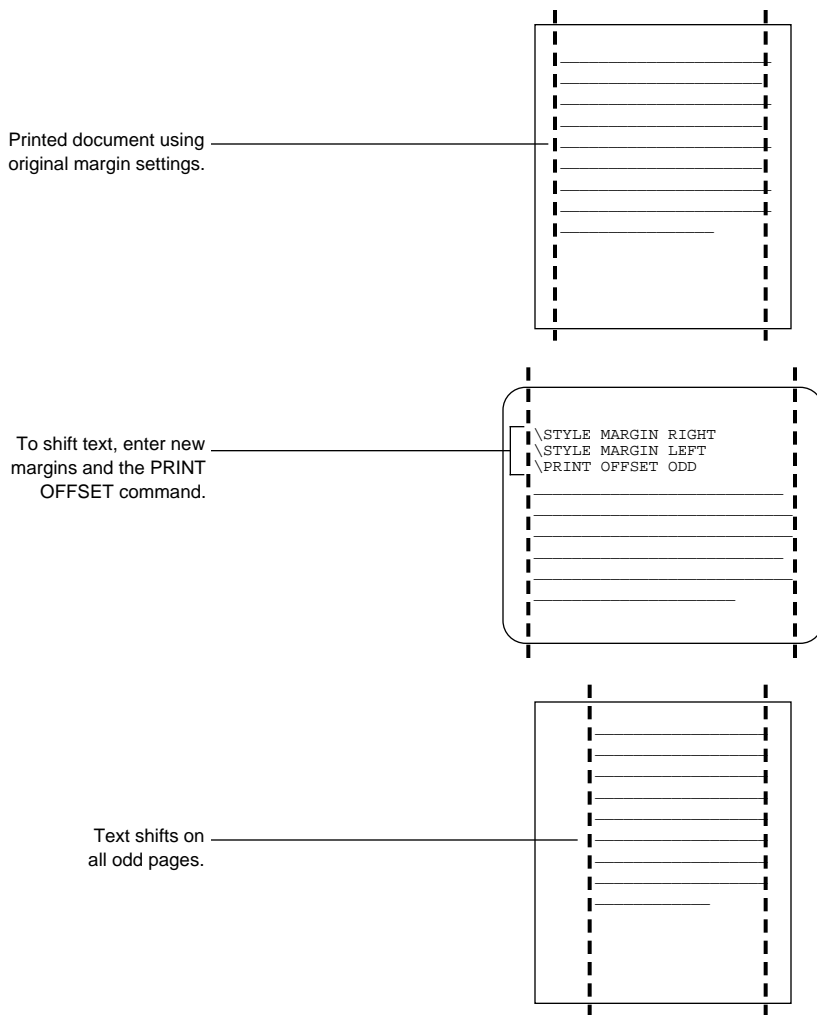


In practice, the PRINT OFFSET command sets different margins for even and odd pages. Because TFORM formats the text to fit the margins before implementing the PRINT OFFSET command, you must reset your margins so that the shifted text fits on the page. This concept is shown in Figure 12-1.

## Printing Your Document

### Shifting the Text on the Page

Figure 12-1. Offsetting the Text





For odd-numbered pages, you set a wide left margin and a narrow right margin. For even-numbered pages, set a narrow left margin and a wide right margin. You must set the margins on the first odd page and the first even page. The text on all subsequent pages shifts according to these command settings.

Figure 12-2 shows a two-page edit file. The commands on the first page shift text on odd pages. The commands on the second page shift text on even pages. Figure 12-3 illustrates two pages of shifted text.

Figure 12-2. Entering Print Offset Commands

Shifts text on odd pages	<pre> \NEW \STYLE MARGIN TOP 1 \STYLE MARGIN LEFT 1.75 \STYLE MARGIN RIGHT .5 \PRINT OFFSET ALL ODD .25 \HEADER " page # " \SET JOIN ON This manual discusses the editing procedures followed in all publications. It presents guidelines that you will follow through the entire editing process. </pre>
Shifts text on even pages	<pre> \NEW \STYLE MARGIN LEFT .5 \STYLE MARGIN RIGHT 1.75 \PRINT OFFSET EVEN .25 Depending on the type of article, you will see from one to three drafts. Technical material goes through two complete editing cycles. Fiction, on the other hand, is edited only once. </pre>

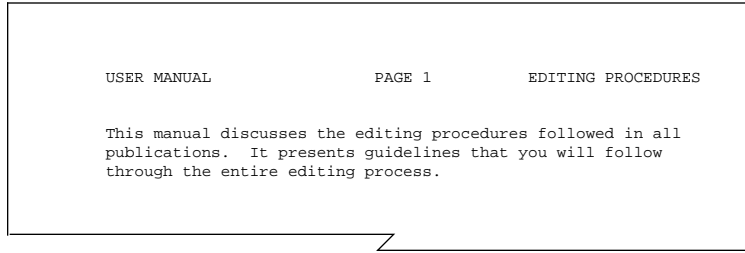
## Printing Your Document

---

### Shifting the Text on the Page

---

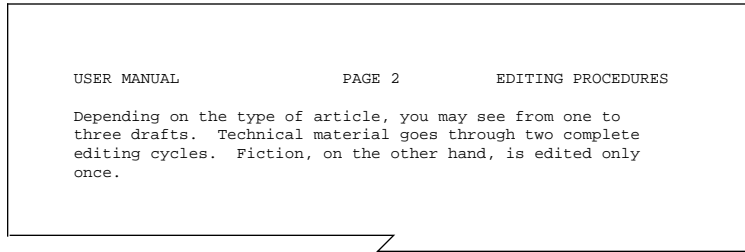
**Figure 12-3. The Shifted Text**



USER MANUAL PAGE 1 EDITING PROCEDURES

This manual discusses the editing procedures followed in all publications. It presents guidelines that you will follow through the entire editing process.

This figure shows a rectangular box representing a page of text. The text is centered horizontally. The top line contains the page header 'USER MANUAL PAGE 1 EDITING PROCEDURES'. The main body of text consists of two lines: 'This manual discusses the editing procedures followed in all publications. It presents guidelines that you will follow through the entire editing process.' The bottom right corner of the box has a small diagonal slash mark.



USER MANUAL PAGE 2 EDITING PROCEDURES

Depending on the type of article, you may see from one to three drafts. Technical material goes through two complete editing cycles. Fiction, on the other hand, is edited only once.

This figure shows a rectangular box representing a page of text. The text is centered horizontally. The top line contains the page header 'USER MANUAL PAGE 2 EDITING PROCEDURES'. The main body of text consists of two lines: 'Depending on the type of article, you may see from one to three drafts. Technical material goes through two complete editing cycles. Fiction, on the other hand, is edited only once.' The bottom right corner of the box has a small diagonal slash mark.

---

**Summary** This section discussed a number of ways you can print out a document using TFORM commands. It also told you how to create a TFORM macro and assign it to a function key. Table 12-1 summarizes the commands discussed in this section.

---

**Table 12-1. Commands Used for Printing**

To Control	Use This Command
Printing	TFORM (implied RUN)
Page Display on Screen	PRINT MANUALLY ON
Pages Printed	PRINT PAGES
Text Shift	PRINT OFFSET

---

---

# Appendix A Command and Function Key Equivalents

---

This appendix contains a table showing several different ways to perform TEDIT functions. Sometimes you can use one or more TEDIT commands to perform a function; sometimes you can use terminal keys or combinations of keys to perform the same functions as TEDIT commands. These function equivalents are especially useful when you are giving written commands on the \*C response line, assigning a series of commands to a function key, or creating an OBEY file.

## Moving the Cursor

TEDIT Functions	Terminal Functions
<code>RETURN</code> key (with RETURN function OFF) FORWARD + LINE NEWLINE	<code>TAB</code>
<code>RETURN</code> key (with RETURN function ON) FORWARD + CHARACTER BACKWARD + CHARACTER	<code>TAB</code> + <code>→</code> <code>→</code> <code>BACKSPACE</code> <code>←</code>
FORWARD + EOL BACKWARD + LINE BACKWARD + LINE + FORWARD FORWARD + LINE repeat count + FORWARD + LINE BACKWARD + LINE repeat count + BACKWARD + LINE	<code>CTRL</code> + <code>RETURN</code> <code>BACKTAB</code> + <code>BACKTAB</code> <code>BACKTAB</code> <code>↓</code> <code>↑</code>

## Command and Function Key Equivalents

---

### Scrolling Vertically

#### TEDIT Functions

#### Terminal Functions

PREVPAGE  
repeat count + PREVPAGE  
BACKWARD + WINDOW  
BACKWARD + repeat count +  
PARAGRAPH

PREV PAGE

FIRSTPAGE  
DISPLAYLINE 0  
DISPLAYLINE first  
MARKPOSITION n + GOTOPOSITION n

SHIFT + PREV PAGE

NEXTPAGE  
FORWARD + WINDOW  
FORWARD + repeat count + PARAGRAPH

NEXT PAGE

LASTPAGE  
DISPLAYLINE n  
DISPLAYLINE last  
MARKPOSITION n + GOTOPOSITION n

SHIFT + NEXT PAGE

ROLLUP

ROLL UP

ROLLDOWN

ROLL DOWN

repeat count + ROLLUP

SHIFT + ROLL UP

repeat count + ROLLDOWN

SHIFT + ROLL DOWN

### Scrolling Horizontally

RIGHTSCROLL  
SETPROFILE, tab stop n + TAB

LEFTSCROLL  
SETPROFILE, tab stop n + BACKTAB  
RETURN key (with RETURN function ON)

**Inserting and Deleting Characters**

TEDIT Functions	Terminal Functions
INSERT + CHARACTER	CHAR INS
INSCAR	SHIFT + CHAR INS
DELETE + CHARACTER	CHAR INS
DELCHAR	SHIFT + CHAR INS + BACKSPACE

**Inserting and Deleting Lines**

INSERT + LINE	LINE INS
INSLINE	
DELETE + LINE	DEL LINE
DELLINE	

**Erasing Text**

ERASE + EOL	CTRL + ERASE LINE
ERASE + WINDOW	CTRL + PAGE ERASE
ERASE + CHARACTER	SPACE bar

**Breaking Lines**

INSERT + EOL  
INSERT + SENTENCE  
BREAKLINE

## Command and Function Key Equivalents

---

### Aligning Lines

TEDIT Functions

Terminal Functions

---

ALIGN + LINE  
BREAKLINE (with auto word wrap ON)

### Repeating commands

repeat count + text command + text item  
command + REPEAT  
FC

### Moving Text to Another File

TEDIT file-a; OPENWINDOW file-b; COPY + text item (in file-a); RETRIEVE (in file-b)  
TEDIT file-a; COPY + text item (file-a); WRITE scratchpad, file-b  
TEDIT file-b; READ file-a, line range; RETRIEVE

---

# Glossary

---

This glossary contains terms from both parts of this manual—Part One, “Using PS TEXT EDIT” and Part Two, “Using PS TEXT FORMAT.” For definitions of all the TEDIT commands and the options associated with them, refer to the *PS TEXT EDIT Reference Manual*. For definitions of all the TFORM commands, refer to the *PS TEXT FORMAT Reference Manual*.

***Align width.*** The width is a specific column number to which TEDIT adjusts the line width at the right margin of the document when you give the ALIGN command or use the auto word wrap feature.

***Auto word wrap.*** If this feature is set to ON and the text on the line containing the cursor exceeds the align width, TEDIT automatically adjusts the right margin to the align width when you press RETURN. TEDIT also automatically adjusts the margin on any text you retrieve (after moving or copying it).

***Block separator.*** A character in the list file for a TFORM form letter separating the blocks of data inserted in each letter.

***Body file.*** The file containing the text for a form letter.

***Breaking a line.*** Dividing a line of text into two parts and moving the second part down to the next line. If there is text on the next line, it is moved down to accommodate the text from the broken line.

***Command.*** An instruction that tells the program to perform a specific action. A TEDIT command can be a combination of text command and text item, such as DELETE plus WORD, or it can be a single word, such as FINDNEXT, which does not require a text item for its object. TFORM commands begin with a backslash; for example, \SET JOIN ON.

***Command interpreter.*** The process that manages the interaction between you and the computer. You start TEDIT or TFORM by typing their names at the command interpreter prompt.



**Command syntax.** The rules that tell you the particular way to type commands so that TEDIT or TFORM understands them.

**Create (a file).** To set up a specific location on a disk where the document you write in a file is stored.

**Cursor.** A movable line or box (sometimes blinking) on the screen that indicates where the next character is to be inserted, replaced, or deleted.

**Data block.** The customized information to be inserted in a TFORM form letter. Data blocks are stored in the list file and are separated by a block separator.

**Date format.** The printed appearance of the date, as defined by the TFORM TODAY command. The most commonly used format, TODAY SHORT USA, prints the date as numerals separated by slashes (5/21/87).

**Default.** A preset condition or setting that is in effect until you specify something explicitly.

**Default functions.** The functions, such as commands or identification of text items, initially assigned to the TEDIT function keys by the TEDIT contact. The keys perform these assigned actions unless you change them using the RECONFIGURE FUNCTION-KEYS command.

**Default profile.** The profile assignments for options and function keys that TEDIT uses when you create a document unless you create and specifically request the use of another profile. The default profile is set up by your TEDIT contact.

**Default volume and subvolume.** The location where your files are stored unless you specify another location explicitly.

**Disk.** The device in which the files you create are stored.

**Document.** The text you write and save in a file.

**Edit file.** A file that contains your text. Files you create with TEDIT are edit files. You enter formatting commands into an edit file.

**Editing program.** A program, such as TEDIT, that allows you to create and modify a document and store it on the computer.

**Editing session.** The period of time during which you edit a specific document. The session starts when you open the file and ends when you close it by exiting.

**Editor.** The program you use to enter and edit text. TEDIT is an example of a full-screen editor.

**EOF.** Stands for end of file. When the last line of your file shows on the screen, TEDIT displays EOF on the status line. If you make a mistake when you ask TEDIT to open a file, you can exit from the editor by giving an EOF! command (hold down the CTRL key and type Y) at the "File:" prompt.

**Field.** In TEDIT, the designated area on a screen where you can type information. Any of the TEDIT response lines can be referred to as fields. Each area where you change a profile option is a field.

In TFORM, the three areas separated by vertical bars that define the right, center, and left entries in a title.

**File.** Related information stored under one name. The file (like a folder) resides in a subvolume (like a file drawer) that in turn resides in the disk volume (like a file cabinet).

**File security.** When you ask to edit a file, TEDIT checks to make sure you are allowed access to the file. TEDIT supports the security system that Tandem uses on files. You can read and modify (write to) files that you own; you can secure your files, for example, so that others can read them but not modify them.

**Footer.** Text printed along the bottom of a page in a document.

**Footer gap.** The distance on a page between the last line of text and the first line of the first footer.

**Form length.** The length of the paper you're printing on (usually expressed in inches).

**Form letter.** A customized set of letters containing the same basic text but individualized with such information as name, address, and position.

**Form width.** The width of the paper you're printing on (usually expressed in inches).

**Format.** The layout of information on a page.

**Formatter.** A program that controls the printed appearance of a document.

**Function.** The specific action performed by TEDIT or the terminal when you press a certain key or give a certain command.

**Function key.** A key on your terminal keyboard (usually labeled with its function) that performs a specific function, as opposed to one that enters a character or a number. The key can be controlled either by the terminal or by TEDIT. See also *TEDIT function key*, *Terminal function key*.

**Hanging indent.** An indentation that extends outward into the margin. Also called a negative indent.

**Header.** Text printed along the top of a page in a document.

**Header gap.** The distance on a page between the last line of the last header and the beginning line of text.

**Help.** The information that TEDIT or TFORM displays on your screen when you give the HELP command. You can specify a particular topic or command on which you want information.

**IMT Contact.** The person at your installation who administers IMT products, including TEDIT and TFORM.

**Indent.** The amount of space from the margin to the first character of text on a line (left indentation) or the last character of text to the right margin (right indentation).

**Interline spacing.** The number of lines between the text on a page. Most documents are single-spaced (no blank lines between text lines) or double-spaced (one blank line between text lines).

**Joining.** Adjusting the text in a paragraph to produce lines of similar length.

**Justification.** Adjusting the spacing in a line of text to produce even right margins in paragraphs.

**Level headings.** A text entry alone on a line that indicates the beginning of a new section or topic in the text. There are eleven types of level headings in TFORM.

**List.** A set of entries, such as words, sentences, or paragraphs, that you put together in a specific order.

**List file.** A file containing the unique information (variables) for each recipient of a TFORM form letter.

**Local profile.** A TEDIT profile that is stored in your default volume and subvolume as opposed to the system profile (your initial default profile), which is most likely stored in a different volume and subvolume.

**Log on.** To establish communication with the computer by identifying yourself to the computer's operating system.

**Nesting.** Placing a subset of TFORM commands or data within another set. For example, you can nest styling commands, or you can produce nested lists.

**Online help.** The information that TEDIT or TFORM displays on your screen when you give the HELP command. You can specify a particular topic or command on which you want information.

**Option.** The optional information that limits or expands the functioning of a command. Most commands have default options. You can change the options for some TEDIT commands in the options part of the profile, which TEDIT displays after you give the RECONFIGURE OPTIONS command. With other commands—SEARCH, for example—you type the options on the response lines for the command.

**Orphan.** The first line of a paragraph isolated at the bottom of a printed page by a page break.

**Page.** In TEDIT, the amount of information that is displayed on the screen at one time. You can change the page size by adjusting the Overlap on the RECONFIGURE OPTIONS screen. You scroll from one page to the next by pressing the `NEXT PAGE` or `PREV PAGE` keys.

In TFORM, a page is the printed text on a single sheet of paper.

**Page break.** The automatically generated move to a new page when the text area on the current page is filled. You can enter manual page breaks using the TFORM NEW command.

**Page form.** The size of the paper you're printing on. Page form includes page width and page length and is usually expressed in inches.

**Page layout.** The settings that determine how the printed page appears—for example, page form, margins settings, and interline spacing.

**Paragraph gap.** The number of blank lines separating paragraphs.

**Pattern.** Used in TEDIT with search-text or replace-text, patterns contain special characters (called wild cards) that represent or match more than one character.

**Pattern character.** The character that indicates the end of an address section in a TFORM form letter list file.

**Persistence.** A TEDIT feature that allows a text command or text item to remain effective once you use it. Thus, you can reuse commands or items without re-entering them. TEDIT displays the persistent command or item on the status line.

**Previewing.** Displaying a printed document on the terminal screen before sending it to a printer.

**Print offset.** Moving the text on the page to allow space for binding or hole-punching.

**Profile.** A profile contains settings that control how TEDIT performs certain commands. The profile consists of options for commands and settings for function keys. You change the profile using the RECONFIGURE OPTIONS and RECONFIGURE FUNCTION-KEYS commands. You can configure and save up to fifty different profiles.

**Profile file.** The file that contains all of your TEDIT profiles; it is usually called TEDPROFL and is stored in your default volume and subvolume.

**Ragged margin.** An uneven margin, usually along the right side of a page. Turning on justification produces a uniform right margin.

**Range.** A specification in TEDIT for a certain area defined by indicating the line or column numbers that delimit the area. You can refer to lines in the range by numbers or by special designators, such as FIRST or LAST.

**RECONFIGURE FUNCTION-KEYS screens.** The two screens where TEDIT displays the assignments for the function keys. You can change the settings by typing over the assignments that are currently displayed.

**RECONFIGURE OPTIONS screens.** The two screens where TEDIT displays the assignments for the options part of the profile. You can change the settings by typing over the assignments that are currently displayed.

**Relative values.** A value you enter that adds or subtracts from the current TFORM setting. You can use relative values to set margins and gaps, alter print offset, or change indentation.

**Replace-text.** The string of characters TEDIT puts in place of other characters (called the search-text) when you give the REPLACE command.

**Response line.** A highlighted line where TEDIT prompts you to type specific information. For example, on the \*C response line, you type commands or text items.

**RETURN function.** The ability to use the RETURN key as a TEDIT function key. If you have the RETURN function available on your terminal, you can assign any TEDIT function to the RETURN key.

**Ruler.** The highlighted line in TEDIT containing numbers and other characters that mark each of the columns across the screen.

**Scratchpad.** A temporary holding area for text that you are copying or moving.

**Screen.** The illuminated display surface (CRT) where you interact with TEDIT to create and modify documents. Also the text that is visible on the screen, referred to as a page. See also *Page*.

**Scroll.** A TEDIT function that moves the text forward and backward or right and left across the screen.

**Search-text.** The string of characters that you tell TEDIT to search for when you give a SEARCH command. Also the string of characters that TEDIT replaces with replace-text when you give the REPLACE command.

**Shifted function.** The function TEDIT performs when you press a function key while holding down the SHIFT key.

**Status line.** The highlighted display line at the bottom of the screen that gives you information about your TEDIT file, such as its name and the numbers of the lines that are currently visible.

**Styling.** Emphasizing text through appearance or placement. You can style text with TFORM using boldface type, underlining, shadow printing, subscripts, and superscripts.

**Subvolume.** Part of the designation that identifies where you store a document. The subvolume (like a file drawer) holds individual files (like folders); the drawers are part of the disk volume (like a file cabinet).

**System date.** The current date, as stored in the TFORM SYS\_DATE variable. The TODAY command determines the format of the date.

**System time.** The current time, as stored in the TFORM SYS\_TIME variable.

**TEDIT contact.** The person at your installation who sets up your initial TEDIT profile, who provides a template for the function keys, and who can answer questions you have about how to use TEDIT.

**TEDIT function key.** A key on your terminal keyboard (usually labeled with a number) that has a TEDIT command, text command, or text item assigned to it. When you press the key, TEDIT performs the assigned function.

**TEDPROFL.** The file that contains all of your TEDIT profiles; it is usually stored in your default volume and subvolume.

**Template.** A flexible strip that fits on the terminal keyboard and identifies the functions assigned to the TEDIT function keys.

**Template block.** The first block in a TFORM form letter list file that defines the order of the variables.

**Terminal function key.** A key on your terminal keyboard (usually labeled with its function) that performs a specific function, as opposed to one that enters a character or a number. The terminal function keys perform the same functions whether you are entering text on the screen or on TEDIT response lines. You cannot reassign functions to some of the terminal function keys like you can to the TEDIT function keys.

**Text command.** A word that tells TEDIT to perform a specific action on an object called a text item. The text command must be used in combination with a text item; for example, use the text command DELETE with the text item WORD to tell TEDIT to delete a word.

**Text formatter.** A program, such as TFORM, that formats text prior to printing.

**Text item.** A word that identifies a specific object on which you intend for TEDIT to perform some action. The text item must be used in combination with a text command; for example, use the text item WORD with the text command DELETE to tell TEDIT to delete a word.

**Title.** A TFORM entry, consisting of three fields, that contains text. You use titles for headers, footers, and level 0 section headings; or you can place them anywhere in your document by using the TITLE command.

**Topic.** One of several subjects on which TEDIT provides information in the online help, such as *scratchpad* or *patterns*.



**Unit of measurement.** An entry required by some TFORM commands that defines how a value is measured—for example, lines, inches, characters, picas, or points.

**Unshifted function.** The function TEDIT performs when you press a function key without pressing the SHIFT key.

**Value.** A numerical entry required by some TFORM commands.

**Variable.** A quantity whose value changes. For example, SYS\_DATE and SYS\_TIME are variables that always contain the current date and time. In TFORM form letters, variables contain the information that changes in each letter.

**Volume.** Part of the designation that identifies where you store a document. The volume (like a file cabinet) holds subvolumes (like file drawers) which contain files (like individual folders).

**White space.** An area in your document that contains no text.

**Widow.** The last line of a paragraph isolated at the beginning of a new printed page by a page break.

**Wild card.** A character in TEDIT that can represent or match more than one character. You can use wild cards in patterns to search for specific text patterns; for example, all the words that end with “ing.”

**Window.** A view of a file; you can have one or two windows open at a given time. With two open windows, you can see two views of one file or see views of two different files. The TEDIT text item WINDOW is the area between the cursor position and the bottom of the screen.

**Wrap.** The movement of the cursor from one line to the next (or previous) line as it continues in the same direction it was traveling when it reached the edge of the screen. See also *Auto word wrap*.

**Written commands.** Commands that you type on the \*C response line as opposed to those that are assigned to function keys and which TEDIT performs when you simply press the key.

---

# Index

---

## A

### Abbreviating

    SETPROFILE options 6-3

    TEDIT commands and items 3-9, 5-17/18

    TFORM commands 7-6

ADDRESS command 11-23/25

Address variable 11-22

Advice message (TEDIT) 3-13

ALIGN command 2-21/22, 4-18/19, 4-27

Align width, changing 2-22

Alphabetizing with MOVE LINE 3-8/9, 5-15/16

Angle brackets for named function keys 5-27

Arrow keys for moving cursor

    between windows 4-3/4

    in text 1-8

    within response lines 4-2

Audit, line renumbering 6-13

Auto word wrap

    adjusting line width while typing 4-7/11, 4-17

    replacing lines longer than align width 5-30/31

    when to turn off 5-35/36, 6-3

---

## B

Backslash 7-5

BACKSPACE key

    erasing characters with 1-11

    moving cursor with 1-8

BACKTAB command 5-26, 5-28/29

BACKTAB key 1-8, 4-19

BACKWARD command 2-17, 4-18/19, 6-5, 6-7/8

BEGIN LIST command 11-7, 11-8  
Blank pages  
    See NEW EVEN or NEW ODD command  
Block separator 11-22  
Body file 11-21  
BOF (beginning of file) 6-11  
BOLD command 10-3/6  
Boxes 10-10/11  
BREAK command 9-6  
    using to stop joining 9-6  
Breaking lines  
    with BREAKLINE and auto word wrap 4-10/11, 4-14/17  
    with INSERT SENTENCE 2-18, 4-14/17  
BREAKLINE command 4-10/11, 4-14, 4-14/17

---

## C

CANCEL command 2-20/21, 4-19, 5-15  
Canceling  
    persistence 2-20/21  
    SAVEPROFILE or USEPROFILE command 3-16/17  
CENTER command 9-24/25  
Centering text 9-24/25  
    default setting 9-24  
    turning ON and OFF 9-24  
CHAR DEL key 1-10  
CHAR INS key 1-9  
Characters  
    deleting 1-10  
    inserting 1-9

- 
- null 5-12, 5-14, 5-32/33
  - uppercasing 5-18
  - CLOSEWINDOW command 4-5, 4-20, 5-7, 6-8, 6-15
  - Closing a window 4-5, 4-20, 5-7, 6-8, 6-15
  - Columns
    - combining 5-10
    - deleting 5-12/13
    - determining width of 3-2/3
    - displaying by number 6-11
    - inserting 5-14
    - moving columns of text 5-11/14
    - numbers for
      - using to move cursor 6-9/15
      - using to set tab stops 3-5
    - typing in 3-2/6
  - Combining files 6-2/4
  - Command interpreter
    - commands
      - EOF! (CTRL + Y) 1-4
      - TEDIT 1-2
      - WHO 5-2
    - prompt 1-2, 1-16
      - giving several TEDIT commands from 5-3
  - Command prompt (TEDIT) 2-13
  - Commands
    - abbreviating TEDIT 3-9, 5-17/18
    - abbreviating TFORM 7-7
    - entering TEDIT
      - in a series using \*C 5-17/18
      - in a series using an OBEY file 5-21/22

in one step using \*C 5-3  
syntax and 5-3/4  
using function keys 1-13/14, 2-15/17  
using semicolon for separating 5-17  
written 2-15  
entering TFORM 7-5/7  
syntax of TFORM 7-5/6

Commands (TEDIT)

\*C 1-14  
abbreviating 3-9, 5-17/18  
ALIGN 2-21/22, 4-18/19, 4-26  
BACKTAB 5-26, 5-28/29  
BACKWARD 2-17, 4-18/19, 6-5, 6-7/8  
BREAKLINE 4-10/11, 4-14, 4-15/17  
CANCEL 2-21, 4-19, 5-15  
CLOSEWINDOW 4-5, 5-7, 6-8, 6-15  
COMPRESS 6-13  
COPY 2-8/10, 3-10/11, 4-1, 4-4/5, 5-7, 5-28/29, 6-14  
DEFINEREGION 5-5  
DELETE 1-10/11, 2-16, 2-20, 5-10  
DISPLAYLINE 6-12  
EMPTYSCRATCHPAD 3-9, 5-16  
EXIT 1-16, 6-13  
FC 5-19/20  
FINDNEXT 4-23  
FINDPREV 4-23  
FORWARD 2-17, 4-18/19, 5-18, 5-21, 6-5, 6-7/8  
GOTOPOSITION 6-5, 6-17/18  
HELP 1-15, 3-11/13, 4-24  
INSERT 2-4/5, 2-18, 3-14, 4-13/17, 5-10, 5-21/23, 5-29

---

LEFTSCROLL 5-24/26, 5-29, 6-11  
LOWERCASE 3-14  
MARKPOSITION 6-16/17  
MOVE 2-6/14, 3-8/9, 5-15, 6-8  
NEWLINE 4-10, 5-18, 5-21  
OBEY 5-21/22  
OPENWINDOW 4-2/3, 5-3, 5-21, 6-8, 6-15  
PURGEPROFILE 3-18  
READ 6-2/4, 6-20  
RECONFIGURE FUNCTION-KEYS 4-8/11, 5-13/14, 5-30,  
5-36  
RECONFIGURE OPTIONS 3-3/5, 3-15, 4-6/7, 4-14, 5-26,  
5-30, 5-35/36, 6-3, 6-5/6  
REDO 2-23  
RENUMBER 6-12/13  
REPEAT 5-20, 5-27  
REPLACE 4-21/22, 4-25/26, 5-11/14, 5-18, 5-21, 5-28,  
5-30/33  
RETRIEVE 2-7/9, 2-12, 3-9, 3-11, 4-5, 5-6/7, 5-16, 5-28/29,  
6-3, 6-8  
RIGHTSCROLL 5-24/26, 5-29  
RULER 3-2/3, 5-21/23  
SAVEPROFILE 3-15/18, 4-11, 4-14, 5-35  
SEARCH 4-20/24, 6-5, 6-7  
SETPROFILE 3-18, 5-26/27, 6-3  
SHOWNUMBER 6-9/10, 6-14/15  
SIZEWINDOW 5-6  
TAB 3-3, 3-6, 4-14, 4-19, 5-26, 5-28/29  
text command, defined 2-15  
UNDO 2-23, 3-9, 5-14, 5-16, 5-21, 6-8, 6-13  
UNMARKPOSITION 6-19

---

UPPERCASE 3-14, 5-18, 5-21  
USEPROFILE 3-17/18  
WRITE 6-2, 6-14, 6-18/19  
Commands (TFORM)  
ADDRESS 11-23/25  
BEGIN LIST 11-7  
BOLD 11-3/4  
BREAK 9-6  
CENTER 9-24/25  
COMMENT 11-27  
DOWN 11-3/4  
END LIST 11-8  
FOOTER 10-1/25  
    FOOTER ALL ODD 10-19  
    FOOTER EVEN 10-18/24  
    FOOTER FIRST 10-24  
    FOOTER NOW 10-25  
    FOOTER ODD 10-18/24  
HEADER 10-1/25  
    HEADER ALL ODD 10-19  
    HEADER EVEN 10-18/23  
    HEADER FIRST 10-25  
    HEADER NOW 10-25  
    HEADER ODD 10-18/23  
HELP 7-9/10  
INDENT 9-15/23  
ITEM 11-8  
KEEP 8-27  
LETTER 11-24/25  
LEVEL 11-12

---

LEVEL CONTENTS 11-17/18  
LIST 11-7/9  
    BEGIN LIST 11-7  
    END LIST 11-8  
    ITEM 11-7/9  
NEW 8-23/26  
    NEW EVEN 8-24/26  
    NEW ODD 8-24/26  
PAGE 10-24  
PRINT  
    PRINT MANUALLY 12-4  
    PRINT OFFSET 12-7/9  
    PRINT PAGES 12-6  
RUN  
    *See* TFORM RUN command  
SET  
    SET BOX 11-10  
    SET JOIN 9-4/9  
    SET JUSTIFY 9-10/12  
    SET SPACING 8-14  
SPACE 9-26/28  
STYLE  
    STYLE FORM LENGTH 8-4  
    STYLE FORM WIDTH 8-4  
    STYLE GAP FOOTER 10-26/27  
    STYLE GAP HEADER 10-26/27  
    STYLE MARGIN 8-7/13  
    STYLE MARGIN FOOTER 10-25/27  
    STYLE MARGIN HEADER 10-25/27  
    STYLE PARAGRAPHS BOTTOM 8-29



---

STYLE PARAGRAPHS GAP 8-20  
STYLE PARAGRAPHS INDENT 9-13/15  
STYLE PARAGRAPHS TOP 8-29  
STYLE SENTENCES 8-16/20  
TFORM (RUN) 11-24, 12-2/5  
TI 9-15/16  
TIME 10-14/15  
TODAY 10-12/13  
UNDERLINE 11-3/4  
UP 11-3/4  
COMMENT command 11-27  
COMMENT line in OBEY file 5-21  
Comments  
    adding to a document 11-27  
COMPRESS command 6-13  
Compressing files for storage 6-13  
CONFIRM option for REPLACE 4-25  
COPY command 2-8/10, 3-10/11, 4-1, 4-4, 5-6, 5-28/30, 6-14  
Copying  
    lines 3-10/11  
    paragraphs from one file to another 4-4  
    range of lines to another file 6-14  
    regions 5-4/7  
    text from one file to another 4-1, 5-4/7  
Creating  
    documents 1-1, 1-3  
    new files  
        AGENDA 3-1  
        DOCUMENT 6-2  
        LAZY 5-21

LETTER 4-2  
MEMO 1-2/3  
TABLE 5-2  
    using the ! shortcut 5-2  
oversized files 5-1/37  
profiles  
    AGENDA 3-4/5, 3-15/17  
    LETTER 4-6/11  
    TABLE 5-35/36  
CTRL key + LINE ERASE key 3-5  
CTRL key + RETURN key 2-12  
CTRL key + Y (EOF! command) 6-11  
Cursor  
    movement 1-7, 4-4  
    position when retrieving text 3-10  
Customizing  
    form letters with SEARCH and REPLACE 4-12, 4-20/26  
    function key assignments 4-10  
    profile options 3-3/5  
    profiles 3-4/5, 3-15/17, 4-6/11, 5-35/36

---

**D**

Data blocks 11-22

Date

    including in a header or footer 10-12

    including in document text 10-12

Date format

    and the TODAY command 10-12/13

Decimals in line numbers 6-10

Default

- assignments for options and functions 2-22
- paragraph size 4-14
- profile 1-2, 3-4, 3-18
- ranges, when to use 4-25/26
- settings
  - function keys 4-9/11
  - printing using 7-7/8, 12-1
  - profile options 3-4/5
  - table 7-8
- See also* individual commands
- volume and subvolume 1-3, 3-16, 5-2

DEFINEREGION command 5-5

DELETE command 2-16, 5-10

Deleting

- characters 1-10
- columns of text with REPLACE 5-11/14
- contents of scratchpad 3-9, 5-16
- lines 1-11
- marked positions 6-18
- profiles 3-18
- sentences 2-20
- text, keeping track of line numbers when 6-10
- words 2-16

Disk location for storing documents 1-4

Displaying

- columns by number 6-11/12
- lines by number 6-11
- list of marked positions 6-18
- list of profiles 3-17

DISPLAYLINE command 6-11/12

Documents

adding comments to 11-27

creating and storing with TEDIT 1-1, 1-3/4

formatting 7-4

including the date in 10-13

including the page number in 10-13

including the time in 10-14/15

printing 12-2/6

to a file 12-5

to a printer 12-3

to the terminal 12-4

using default settings 7-7/8

three stages of 9-3

DOWN command 11-3/5

---

E

Edit file

defined 7-2

entering TFORM commands into 7-5/8

printing with default settings 7-7/8

using TFORM with an 9-2/3

Editing

larger files 6-1

program 1-1

session 2-1, 2-3, 2-23, 3-5/6, 3-17

text wider than the screen 5-32/33

Editor

See Text editor

EMPTYSCRATCHPAD command 3-9, 5-16

END LIST command 10-8  
End-of-line (EOL) 2-11, 2-15, 2-17, 4-18/19, 5-18  
Entering  
    TEDIT commands 1-13/14, 2-13/15, 5-2/3, 5-17, 5-21/22  
    text in a file 1-7  
    TFORM commands 7-5/7  
EOF (end of file) 6-11  
EOF! command 1-4  
EOL  
    *See* End-of-line  
Erasing  
    profile settings 3-15  
    scratchpad 3-9/10  
    text 1-7  
EXIT command 1-16, 2-24  
Exiting from TEDIT 1-16, 2-24  
    saving profile before 3-5  
    UNDO command and 2-23  
EXTENDED-SEARCH  
    option for searching 4-24  
    wild-card characters and 5-13

---

**F**  
FC command 5-19/20  
Fields (TEDIT), moving between  
    on profile screens 3-4  
    on response lines 4-2/3  
Fields (TFORM) 10-3/7  
    changing entries in 10-8  
    defining with vertical bars 10-3/7

entering text in 10-5/7  
replacing text in 10-8  
    illustrated 10-9  
using in headers and footers 10-3/9

#### File

compressing for storage 6-13  
creating a new 1-3, 3-1, 4-2, 5-2, 5-21, 6-2/3  
entering text in 1-7  
extrawide, working with 5-22/25  
location for storing documents 1-1, 1-3/4, 1-4  
name, specifications for 1-2  
profile 3-16/17  
reopening 2-1, 2-3  
security 4-3  
using with TFORM 7-2

FINDNEXT command 4-23

FINDPREV command 4-23

Fix command

*See* FC

FOOTER command 10-1/25

Footer gap 10-26/28

#### Footers

changing the footer gap 10-26/28  
defined 10-1  
effect of interline spacing on 10-16  
including date in 10-12/13  
including page numbers in 10-10  
including time in 10-14/15  
numbering multiple 10-16  
placement within the edit file 10-1, 10-8, 10-24

- printing
  - after a specified page number 10-24
  - on blank pages 10-1
  - on even and odd pages 10-18/24
  - on specified pages 10-18/24
  - on the current page 10-24
  - on the first page 10-24
  - setting margins for 10-25/26
  - summary of commands 10-2, 10-29
  - using multiple 10-16
  - when take effect 10-1, 10-8, 10-25
- Form length 8-4
- Form letters 11-19/26
  - ADDRESS command 11-23/25
  - creating the body file 11-21
  - customizing with TEDIT 4-20/26
  - defined 11-19
  - entering the address variable 11-23/25
  - printing 11-24/26
    - illustrated 11-26
    - using the LETTER command 11-24/26
    - using the TFORM RUN command 11-24/26
- Form width 8-4
- Formatter
  - defined 7-2
  - uses 7-4
- FORWARD command 2-11, 2-17, 4-18/19, 5-18, 5-21, 6-5, 6-7/8

---

## Function keys

- about using 1-13/14
- assigning functions to
  - delete column 5-13
  - insert column 5-14
  - series of commands 5-17
- changing assignments to 2-22, 4-6, 4-8/9
- named in SETPROFILE command 5-27
- part of profile 4-9/11
- template for 1-2
- terminal versus TEDIT 1-7
- using with written commands 2-15/17

---

## G

### Gap

- footer 10-26/28
  - header 10-26/28
  - paragraph 8-20/22
- Going to marked positions 6-17/18
- GOTOPOSITION command 6-5, 6-17/18

---

## H

Hanging indent 9-13/15

HEADER command 10-1/25

Header gap 10-26/28

### Headers

- changing the header gap 10-26/28
- defined 10-1
- effect of interline spacing on 10-16
- including date in 10-12/13



including page numbers in 10-10  
including time in 10-14/15  
numbering multiple 10-16  
placement within the edit file 10-1, 10-7, 10-24  
printing  
    after a specified page number 10-24  
    on blank pages 10-2  
    on even and odd pages 10-18/24  
    on specified pages 10-18/24  
    on the current page 10-24  
    on the first page 10-24  
setting margins for 10-25/26  
summary of commands 10-2, 10-29  
using multiple 10-16  
when take effect 10-1, 10-7, 10-25  
Headings (TEDIT)  
    column headings in a table 5-22/32  
    row headings in a table 5-9/21  
Headings (TFORM)  
    *See* Level headings  
HELP command (TEDIT) 1-15, 3-11/13, 4-24  
HELP command (TFORM) 7-9/10  
Help, online (TEDIT)  
    list of available functions 3-11/13  
    on commands and topics 1-15, 3-11  
    on response line fields 4-24  
HOME key 1-8, 4-4  
Horizontal  
    rules 5-28/31  
    scrolling 5-22, 5-24/26, 5-28/29

---

**I**

IGNORE-CASE option for SEARCH, REPLACE 4-22, 5-18

INDENT command 9-15/23

Indenting

avoiding unwanted 9-16/19

blocks of text 9-16/19

default setting 7-7/8, 9-13

entering values for 9-16/18

first line of paragraphs 9-13/14

hanging indent 9-13/14, 9-21/23

negative indentation 9-13/14, 9-21/23

nested indentation 9-21/23

single lines 9-15/16

using the INDENT command 9-17

using the TI command 9-16

STYLE PARAGRAPHS INDENT command 9-13/15

turning indentation ON and OFF 9-19/21

turning ON and OFF 9-19/21

using absolute values 9-16/19

using relative values 9-21/23

values used in 9-16

INSERT command 2-4/5, 2-18, 3-14, 4-13/14, 4-15/17, 5-10,  
5-21/23, 5-29/30

Inserting

blank pages 8-23, 10-2

See NEW EVEN or NEW ODD command

See Page breaks

characters 1-9

columns of text with REPLACE 5-11/14

horizontal rules 5-28/31

---

lines 1-11, 2-4/5, 3-14, 4-15/17, 5-23  
paragraphs 4-13/14  
sentences 2-18, 4-15/17  
sentences to break lines 5-10, 5-22  
text, keeping track of line numbers when 6-10  
vertical rules 5-28, 5-30/33

Interline spacing  
changing 8-14/15  
default setting 7-7/8, 8-14

Item  
See Text item  
ITEM command 11-8

---

## J

Joining  
See joining text  
Joining lines  
See joining text  
Joining text 9-4/9  
default setting 7-7/8, 9-4  
defined 9-4  
hints on 9-8/9  
illustrated 9-5  
interrupting 9-6  
turning ON and OFF 9-6/7  
using with justification 9-12  
using with specific commands 9-8

---

Justifying text 9-10/12  
  default setting 7-7/8, 9-10  
  defined 9-10  
  turning ON and OFF 9-10/11  
  using with joining 9-12

---

## K

KEEP command 8-27/28  
  illustrated 8-28

---

## L

LEFTSCROLL command 5-24/26, 5-29, 6-11  
LETTER command 11-24/25  
  address 11-24/25  
LEVEL command 11-12/16  
LEVEL CONTENTS command 11-17/18  
Level headings  
  as a variable 11-12  
  defined 11-12  
  entering 11-12/13  
  grouping in printed document 11-15/16  
  numbering 11-13  
  printing 11-13  
Levels  
  *See* Level headings  
LINE DEL key 1-11  
LINE INS key 1-11

---

Lines (TEDIT)

- adjusting width of 2-21/22, 4-6/7, 4-19
- aligning 4-18
- breaking 2-18, 4-10/11
- copying range to another file 6-14
- deleting 1-11
- displaying by number 6-11
- inserting 1-11, 2-4/5, 3-14, 4-15/17, 5-23
- line numbers, using 6-9/15
- LINE, text item 2-4, 2-15
- moving 2-6/10, 3-8/9, 5-15/16
- rejoining broken 4-18
- renumbering 6-10, 6-12/13
- spacing between 8-14/15
- See also* Rules

Lines (TFORM)

- joining text in 9-4/9
- keeping together 8-27

LIST command 11-7/9

List file 11-22/23

Lists 11-7/9

- BEGIN LIST command 11-7
- END LIST command 11-8
- ending 11-8
- entering items into 11-8
- ITEM command 11-8
- making from a series in a sentence 4-17
- naming 11-7, 11-8

---

- nesting 11-8
- starting 11-7
- types 11-7

Local profile 3-16

Logging on 1-2

LOWERCASE command 3-14

---

## M

Margins 8-7/13

- header and footer 10-25
- text
  - changing temporarily 9-16/19
  - changing with absolute values 8-7
  - changing with relative values 8-7
  - default settings 7-7/8, 8-7

Marking

- boundaries of regions 5-5/6
- positions 6-16/17
- sections 6-5/7

MARKPOSITION command 6-16/17

Message, advice (TEDIT) 3-13, 3-16, 4-22

MOVE command 2-6/14, 3-8/9, 5-15, 6-8

Moving

- around in file by line and column number 6-11
- backward
  - along ends of lines 2-17
  - with BACKSPACE key 1-8
- between fields on profile screens 3-4
- columns of text 5-11/15
- contrasted with copying 3-8

cursor  
  moving between windows 4-4  
  with arrow keys 1-7/8  
  with terminal function key 1-7  
forward  
  to ends of lines 2-11  
  with SPACE bar 1-8  
how TEDIT moves text 2-8/10  
large blocks of text 6-5  
  contents of scratchpad 6-14  
  ranges of lines 6-14  
  REGION 5-4/7, 6-18/19  
  SECTION 6-7/8  
  WINDOW 6-8  
lines 2-6/10, 3-8/9, 5-15/16  
paragraphs 4-4/5  
scratchpad and 2-8/10  
sentences 2-10/12  
text from one window to another 4-1  
to ends of lines 2-11  
words 2-13/14, 2-16/17  
words into alphabetical order 3-8/10

---

**N**  
Named function keys 5-27  
Naming  
  files 1-2, 4-3  
  profiles 3-16

Negative indentation 9-13/14, 9-21/23

Nested

    indentation 9-22/23

    lists 11-8

    styling commands 11-4

NEW command 8-23/26

NEW EVEN command 8-24/26

NEW ODD command 8-24/26

NEWLINE command 4-10, 5-18, 5-21

NEXT PAGE key 3-12, 4-9, 5-6, 6-4/6, 6-16

Null character 5-12, 5-14, 5-33

Numbering pages 10-10/11

Numbers

    column, displaying 6-11

    line

        decimal places in 6-10

        displaying 6-9/10

    moving cursor to line and column 6-9/11

    renumbering 6-12/13

    saving copy of (auditing) 6-13

    status line and 4-2, 5-26, 6-9

    turning off display of 6-15

---

**O**

OBEY command 5-21/22

Offsetting the text 12-7/9

Online help (TEDIT)

*See* Help, online



Opening windows

two into same file 5-8

two into two different files 4-1/3, 4-20, 5-2/7, 5-21, 6-15

OPENWINDOW command 4-2/3, 4-20, 5-2, 5-8, 5-21, 6-7, 6-15

Options

customizing profile 3-3/5, 4-6/8

OPENWINDOW command 4-2/3

REPLACE command 4-25

SEARCH command 4-21/22

*See also* Searching, options to extend or limit

Orphan

controlling number of lines in 8-29

defined 8-29

STYLE PARAGRAPHS TOP command 8-29

---

**P**

Page (TEDIT) 5-6

Page breaks

automatically generated by TFORM 8-23

default setting 8-23

inserting 8-23/26

inserting blank pages

*See* NEW EVEN or NEW ODD commands

KEEP command 8-27/28

keeping items on a page 8-27/28

NEW command 8-23/26

on next even or odd page 8-24/26

preventing unwanted 8-27/28

starting a new page 8-23/26

PAGE command 10-24

- Page form 8-4/7
  - changing settings for 8-4
  - default settings 7-7/8
  - form length 8-4
  - form width 8-4
- Page layout
  - defined 7-2, 8-1
  - illustrated 8-2
  - summary of commands 8-30
  - using commands 8-3
- Page number
  - including in a header or footer 10-10
  - including in document text 10-10
- Paper size
  - See Page form
- Paragraph gap
  - defined 8-20
  - effects of interline spacing on 8-20
  - effects of SPACE command on 8-22
  - setting 8-20/22
- Paragraphs
  - adjusting width of lines in 2-21/22
  - as defined by TFORM 8-19
  - avoiding unwanted page breaks in 8-29
  - changing size definition of 4-14
  - controlling widows and orphans 8-29
  - copying 4-4/5
  - effect of SPACE command on 8-20

---

    indenting the first line of 9-13/15  
        hanging indent 9-13/15  
        using positive and negative values 9-13/15  
    inserting 4-13/14  
    joining text in 9-6/7  
    justifying text in 9-10/12  
    PARAGRAPH, text item 2-21  
    producing uniform lines in 9-6/7  
    realigning after using REPLACE 4-26  
    setting paragraph gap 8-20/22  
    setting spacing between 8-19/22  
    splitting 8-29  
Pattern character 11-22  
Patterns for searching 4-24  
Persistence  
    canceling 2-20/21  
    using 2-4, 2-20/21  
Personalizing form letters 4-20/26  
    *See also* Customizing and Form Letters  
Positions  
    cursor 1-7, 3-10  
    marked 6-15/17  
PREV PAGE key 3-12, 6-4/5, 6-16  
Previewing a document 12-4  
PRINT MANUALLY command 12-4  
PRINT OFFSET command 12-7/9  
PRINT PAGES command 12-6

## Printing

- documents 12-2/6
  - using default settings 7-7/8
- headers and footers 10-18/24
- level headings 11-13/16
  - grouping 11-15
  - numbering 11-13
- PRINT MANUALLY command 12-4
- PRINT PAGES command 12-6
- printer requirements 12-3
- redirecting the output 12-4/5
- selected pages 12-6
- sending document a file 12-5
- sending document to printer 12-3
- sending document to terminal 12-4
- setting parameters for 12-3
- summary of commands 12-11

## Profile

- creating and saving 3-3/5, 3-15/17, 4-6/11, 5-26/27, 5-35/36
- default 1-2, 3-18
- defining section marker in 6-5/6
- deleting 3-18
- file for storing profiles 3-16/17
- function keys part of 4-9/11
- functions
  - inserting vertical rules 5-36
  - moving columns of text 5-13/14, 5-36
  - RETURN and auto word wrap 4-8/11
  - ROLL UP and ROLL DOWN 5-30
  - turning off auto word wrap with RETRIEVE 6-3

---

- identifying for USEPROFILE list 3-15
- listing all your profiles 3-17
- local 3-16
- names 3-16/17
- options
  - auto word wrap 4-7/11, 5-36, 6-3
  - default 3-3/4
  - line width 4-6/7
  - paragraph size 4-13
  - RETURN function 4-7/11
  - section marker 6-6
  - tab stops 3-5, 4-6/7, 5-26/27, 5-36
- original assignments in 2-22
- using another 3-15/17
- Prompt
  - command interpreter (TACL) 1-2
  - TEDIT command 2-13
- PURGEPROFILE command 3-18

---

## Q

Qualifiers 7-6

---

## R

### Ranges

- copying lines to another file 6-14
- for SEARCH and REPLACE 4-23, 4-25/26

READ command 6-2/4, 6-20

Reading

- marked positions 6-19
- text from another file 6-2/4, 6-20

Recalling and reissuing commands (FC) 5-19/20  
RECONFIGURE FUNCTION-KEYS command 4-8/11,  
5-13/14, 5-31, 5-36  
RECONFIGURE OPTIONS command 3-3/5, 3-15, 4-6/8, 4-14,  
5-26, 5-30, 5-35/36, 6-3, 6-5/6  
REDO command 2-23  
Redoing changes that were undone 2-23  
Region  
    defining and copying 5-4/7  
    REGION, text item 5-4  
Remarks field (profile) 3-15  
RENUMBER command 6-12/13  
Renumbering lines 6-12/13  
REPEAT command 5-20, 5-27  
Repeat count 5-18, 5-29/30  
Repeating commands  
    using FC command 5-19/20  
    using REPEAT command 5-20, 5-27  
    using repeat count 5-18/19  
    with # 5-18  
REPLACE command 4-21/23, 4-25/26, 5-12/15, 5-18, 5-21,  
5-28, 5-30/32  
Replace-text 4-25/26, 5-14  
Replacing  
    by typing over existing text 1-7  
    realigning paragraphs after 4-26  
    space with parentheses 5-18  
    to insert vertical rules 5-31  
    to move columns of text 5-11/14  
    using OBEY file 5-21  
    words 4-25/26

Response line

\*C 2-14

bypassing 6-4

canceling before using 2-14

DISPLAYLINE 6-11

fields and RETURN function 3-5

MARKPOSITION 6-16/17

OPENWINDOW 4-2/3, 5-3

READ 6-2/4

RENUMBER 6-13

REPLACE 4-25/26, 5-12

saving information on 5-12, 5-14

SEARCH 4-21/22

secondary 6-4

topic (online help) 3-12

USEPROFILE 3-17

WRITE 6-14

RETRIEVE command 2-7/10, 2-12, 3-8, 3-11, 4-4, 5-6/7, 5-16,  
5-28/29, 6-3, 6-8

Retrieving text

turning off auto word wrap when 6-3

*See also* RETRIEVE command

RETURN function

auto word wrap and 4-8, 4-10, 4-19

horizontal scrolling with 5-25/26

response line fields and 3-4/5

returning to left text margin 3-6

starting a new line and 1-8

RETURN key 1-8

RIGHTSCROLL command 5-24/26, 5-29

---

ROLL UP and ROLL DOWN keys 5-29/30  
RULER command 3-2/3, 5-21/23  
Ruler for determining column placement 3-2/3  
Rules  
    horizontal 5-28/30  
    vertical 5-28, 5-30/32  
RUN command  
    *See* TFORM RUN command  
Running heads  
    *See* Headers

---

## S

SAVEPROFILE command 3-15/18, 4-11, 4-14, 5-35  
Saving  
    copy of marked positions 6-19  
    files 1-4  
    profiles 3-15/17  
Scratchpad  
    defined 2-8/10  
    emptying 3-9/10  
    features of 3-8/9  
    moving lines to, for alphabetizing 5-15/16  
    READ command and 6-2/4  
    WRITE command and 6-2, 6-14  
Scrolling  
    by line and column number 6-11  
    horizontally 5-22, 5-24/26, 5-28  
    to first page or last page of file 6-4  
    vertically with NEXT PAGE and PREV PAGE 3-12, 4-9, 5-6,  
        6-4/6, 6-16



with marked positions 6-15/17  
SEARCH command 4-20/24, 6-5/7  
Search-text 4-21/22, 4-25/26, 5-14  
Searching  
    extending or limiting  
        with patterns 4-24  
        with ranges 4-23  
    for specific lines or columns 6-11/12  
    for specific text 4-20/24  
    options to extend or limit  
        CONFIRM 4-25  
        EXTENDED-SEARCH 4-24  
        IGNORE-CASE 5-18  
        When-found 4-22  
        WORD-DELIMITED 4-22  
Section  
    markers 6-5/8  
    SECTION, text item 6-5  
Security, file 4-3  
Semicolon, separating commands with 5-3, 5-17/18  
Sentences  
    defined by TEDIT 2-11  
    defined by TFORM 8-16  
    inserting to break lines 2-18, 4-14/17, 5-10, 5-22  
    moving 2-10/12  
    reformatting into a list 4-17/18  
    SENTENCE, text item 2-10/11  
    spacing between 8-16/19  
    STYLE SENTENCES command 8-16/19

**Series of commands**

- \*C for giving 5-17/18

- OBEY file for giving 5-21/22

- semicolon for separating 5-3, 5-17/18

**Session, editing** 2-1, 2-3, 2-23, 3-5, 3-17**SET BOX** command 11-10/11**SET JOIN** command 9-6/7**SET JUSTIFY** command 9-10/12**SET SPACING** command 8-14**SETPROFILE** command 3-18, 5-26/27, 6-3**SHIFT** key 1-11, 1-13/14**Shifted function** 1-11, 1-13/14**Shifting the text** 12-7/9**Shortcuts**

- for creating files 5-2

- for giving profile commands 3-18

- for opening windows 5-2/3

**SHOWNUMBER** command 6-9/10, 6-14/15**SIZEWINDOW** command 5-6**SPACE** bar 1-8**SPACE** command

- at beginning of a paragraph 8-22, 9-26

- at end of a paragraph 8-22

- effect of interline spacing on 8-22

- effect on **STYLE PARAGRAPHS GAP** setting 9-26

- effects on paragraph gap 8-22, 9-26

- within a paragraph 8-22, 9-26

Spacing

- changing between sentences 8-16/18
- changing line 8-16/18
- changing paragraph gaps 8-19/22
- default settings 7-7/8, 8-14
- in documents 8-14/22
- interline 8-14

Special features

- summary of commands 11-2, 11-28
- using 11-1/28

Starting TEDIT 1-2

Status line 1-3

- determining line numbers with 6-9
- displaying BOF and EOF 6-11
- file name on 1-6
- persistence displayed on 2-4, 2-21/22
- showing column numbers for wide files 5-25
- window number on 4-2

Storing documents 1-1, 1-4, 1-5, 6-13

String

See Series of commands

STYLE FORM LENGTH command 8-4

STYLE FORM WIDTH command 8-4

STYLE GAP FOOTER command 10-26/28

STYLE GAP HEADER command 10-26/28

STYLE MARGIN command 8-7/13

STYLE MARGIN FOOTER command 10-25/27

STYLE MARGIN HEADER command 10-25/27

STYLE PARAGRAPHS BOTTOM command 8-29

STYLE PARAGRAPHS GAP command 8-20

STYLE PARAGRAPHS INDENT command 9-13/16

STYLE PARAGRAPHS TOP command 8-29

STYLE SENTENCES command 8-16/19

Styling commands 11-3/5

dependence on printer 11-3

entering 11-3/4

illustrated 11-5

location in edit file 11-4

nesting 11-4/5

summary 11-28

using 11-4/5

using multiple 11-4/5

Styling text

commands 11-3/5

Subvolume, storing documents in 1-4

Syntax

of styling commands 11-3/4

of TEDIT commands 5-2/3

of TFORM commands 7-5/6

System date 10-12

System time 10-14

SYS\_DATE 10-12/13

SYS\_TIME 10-14/15

T

TAB command 3-3, 3-6, 4-14, 4-19, 5-26, 5-28/29

TAB key 1-9, 3-4/6, 4-2, 4-14, 4-19

Tab stops

extra-wide files and 5-23, 5-26, 5-36

setting 3-2/6, 4-6/7, 5-26/27

Table of Contents

LEVEL CONTENTS command 11-17/18

modifying the appearance of 11-18

TEDIT

command (TACL) 1-2

contact 1-2, 2-22, 3-3, 4-9, 4-11

function keys 1-13/14

*See also* Function keys

TEDPROFL 3-16

Template block 11-22

Template for function keys 1-2, 1-13/14

Temporary Indent command

*See* TI command

Temporary text holding area

*See* Scratchpad

Terminal keys

arrows for moving cursor 1-8

BACKSPACE 1-11

BACKTAB 1-8, 4-19

CHAR DEL 1-10

CHAR INS 1-9

CTRL + ERASE LINE 3-5

CTRL + RETURN 2-12

CTRL + Y 1-4

---

editing text wider than the screen 5-32/33  
functions described for 1-7/13  
HOME 1-8, 4-4  
LINE DEL 1-11  
NEXT PAGE 3-12, 4-9, 5-6, 6-4/6, 6-16  
PREV PAGE 3-12, 6-4/6, 6-16  
RETURN 1-8  
ROLL UP and ROLL DOWN 5-29/30  
SHIFT 1-11, 1-13/14  
SPACE bar 1-8  
TAB 1-8, 3-5/6, 4-2, 4-14, 4-19

#### Text

adding white space to 9-26/27  
adjusting line lengths 9-4/6  
bolding 11-3/6  
centering 9-24/25  
creating uniform lines 9-6/7  
creating uniform lines in paragraphs  
indenting 9-13/23  
joining 9-4/9  
justifying 9-10  
keeping together 8-27/28  
offsetting on the page 12-7/9  
shifting on the page 12-7/9  
styling 11-3/6  
    location within edit file 11-4/6  
    using multiple commands 11-4/6

---

subscripts and superscripts 11-3/6  
underlining 11-3/6  
uneven lines in 9-4/6  
using TFORM with an edit file 9-2/3  
Text adjusting commands  
    *See* text formatting  
Text command 2-15  
Text editor 1-1, 7-2  
Text formatting commands  
    illustrated 9-1  
    summary 9-29  
Text item  
    defined 2-15  
    end of file and 6-8  
    End-of-line (EOL) 2-11  
    LINE 2-4  
    PARAGRAPH 2-21  
    REGION 5-7  
    SECTION 6-5  
    SENTENCE 2-11  
    WINDOW 6-8  
    WORD 2-13  
TFORM  
    capabilities of 7-4  
    command syntax 7-5/6  
    prompt 7-9

- RUN command
  - printing documents with 12-2/6
  - printing form letters with 11-24/25
  - syntax of 12-2
  - using interactively 7-9
- TFORM RUN command
  - for printing documents 12-2
  - for printing form letters 11-24/25
  - options 12-2
- TI command 9-15/16
- Time
  - including in document text 10-14/15
  - including in headers and footers 10-14/15
- TIME command 10-14/15
- Titles
  - changing 10-8/9
  - defined 10-3
  - defining fields in 10-3
  - entering 10-3/6
  - including page numbers in 10-10
  - replacing fields 10-8
  - syntax of 10-3/6
  - using in level 0 headings 11-12/13
  - using outside of headers and footers 10-4
- TODAY command 10-12/13
- Topic (TEDIT online help) 1-15, 3-11
- Transferring text between files
  - using READ 6-2/4, 6-20
  - using windows 4-1/4, 4-20, 5-2/7, 5-21, 6-15
  - using WRITE 6-2, 6-14, 6-18/19



---

## U

UNDERLINE command 11-3/6  
UNDO command 2-23, 3-9, 5-14, 5-16, 5-21, 6-8, 6-13  
Undoing mistakes 2-23  
    scratchpad and 3-9, 5-16  
    when deleting columns 5-14  
    when renumbering 6-13  
Uneven text  
    *See* Joining text  
Units of measurement (table) 7-6/7  
Unmarking positions 6-18/19  
UNMARKPOSITION command 6-18/19  
UP command 11-3/6  
UPPERCASE command 3-14, 5-18, 5-21  
USEPROFILE command 3-17

---

## V

Values 7-6  
Variables  
    address 11-22  
    in form letters 11-21  
    level headings as 11-12/13  
    page numbers as 11-18  
    system date 10-12  
    SYS\_DATE 10-12/13  
    SYS\_TIME 10-14/15  
Variables file  
    *See* List file  
Vertical rules 5-28, 5-30/32

**Volume and subvolume**

- file location in 1-3
  - profile file location in 3-16
  - using ! to create files and 5-2
- 

**W**

When-found option for SEARCH 4-22

WHO command (TACL) 5-2

Wide files, working with 5-22/25

**Widow**

- controlling number of lines in 8-29

- defined 211 8-29

- STYLE PARAGRAPHS BOTTOM command 8-29

**Wild card**

- EXTENDED-SEARCH used with 5-13

**Wildcard**

- EXTENDED-SEARCH used with 5-14

- patterns for searching 4-24, 5-14

**Windows**

- changing size of 5-6

- opening

- into two different files 4-1/3, 4-20, 5-2/3, 5-21, 6-15/16

- OPENWINDOW options 4-3

- two into same file 5-8

- specifying the number of 4-2

- WINDOW, text item 6-8

WORD-DELIMITED option in SEARCH, REPLACE 4-22

---

**Words**

- deleting 2-16
- moving 2-13/14, 2-16
- replacing 4-25/26
- WORD, text item 2-13

**Wrap**

- auto word 4-7/11
- cursor movement 1-8
- WRITE command 6-2, 6-14, 6-18/19
- Written commands 2-13/17

---

**Special Characters**

- ! (COMMENT command) 11-27
- # (page number) 10-10
- # (repeating commands with) 5-18
- \* (default profile name) 3-18
- \* (to specify the current window) 4-2, 5-7
- \*\* (to specify a second window) 4-2, 5-7
- \*C command 1-14
- ?COMMENT line in OBEY file 5-21
- \ (TFORM command indicator) 7-5