Ridge Text Editor Reference Manual

RIDGE

9001

Ridge Text Editor Reference Manual

Ridge Computers
Santa Clara, California
December 1983

9001

NOTICE

TABLE OF CONTENTS

List of Figures:

List of Tables:

PREFACE

This manual documents the Ridge Text Editor by giving a general introduction to the editor (Section 1), explaining each of the commands and operations (Section 2),and offering a short tutorial (Section 3). The reader should be able to both create and edit files using the text editor after having read this document.

The reader is referred to the "Ridge Operating System Manual" for basic operating system information, such as particulars of file and directory structure, allocation information for discs, and the like. No knowledge of the Ridge System on the reader's part is assumed beyond a minimal acquaintance with the operating system.

SECTION 1

INTRODUCTION

OVERVIEW

The Ridge Text Editor allows the user to create and edit text files in a way that combines the attributes of a screen-oriented editor with the capabilities of a line-oriented editor.

Screen-oriented attributes (for example, moving or copying blocks), use the video display to provide a window on the text file (Figure 1). The displayed text can be edited using terminal features such as insert/delete line, insert/delete character, cursor movement, or scrolling. Changes or alterations are made by entering text in a strike-over manner on the display.

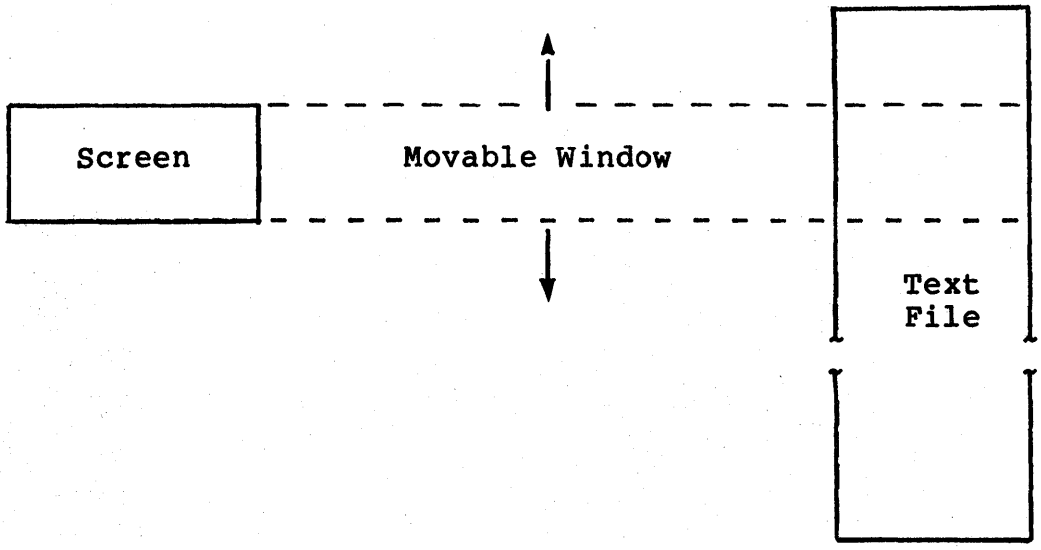


Figure 1. Movable Window on Text File

Line-oriented capabilities provide string search and replacement with ranges and repetition facilities. All editor capabilities, whether screen- or line-oriented, are controlled by user commands.

These commands are broken into two groups:

- EDITING COMMANDS. With this group, the user enters the command into the command entry area of the display.

- PREDEFINED OPERATIONS. These operations are invoked directly by function keys. The predefined operations have been implemented using various combinations of the editing comands.

GENERAL DESCRIPTION

This section describes the logical structure of text files, the display format, ASCII and work files, and the concept of selected blocks and windows.

Text File Logical Structure

A text file consists of a set of records referred to as lines. Each line has a length attribute; the length describes the position of the last non-blank character in the line. The length of a line ranges from zero (0) to 128 characters. A length of zero denotes a blank line.

Each line can be identified by its position relative to the beginning of the file. The position (line number) has a value in the range of 0 to 99,999,999, inclusive. Lines are effectively renumbered if preceding lines are inserted or deleted.

Display Format

The Ridge System supports two terminals: the Ridge Display and the Televideo TVI-950C. The Ridge Display editor format is 50 lines by 128 columns; the Televideo is 25 lines by 80 columns. Figure 2 shows the display format for the Ridge Display and Figure 3 that of the Televideo.

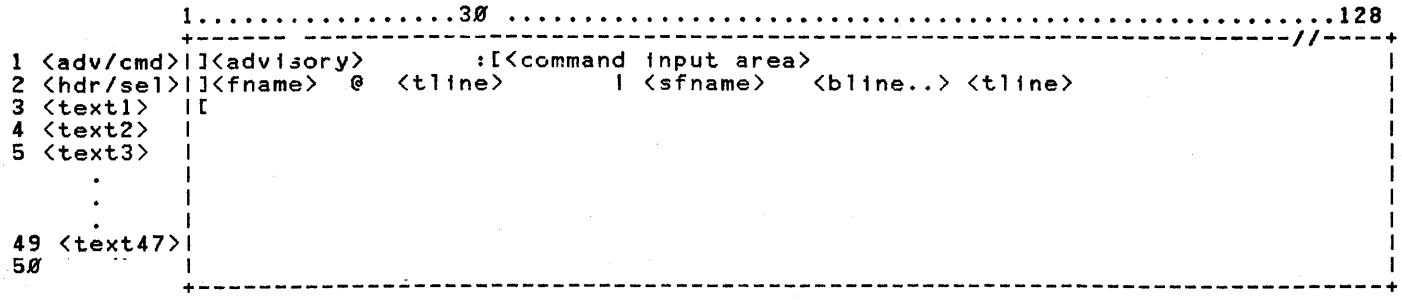


Figure 2. Ridge Display Format

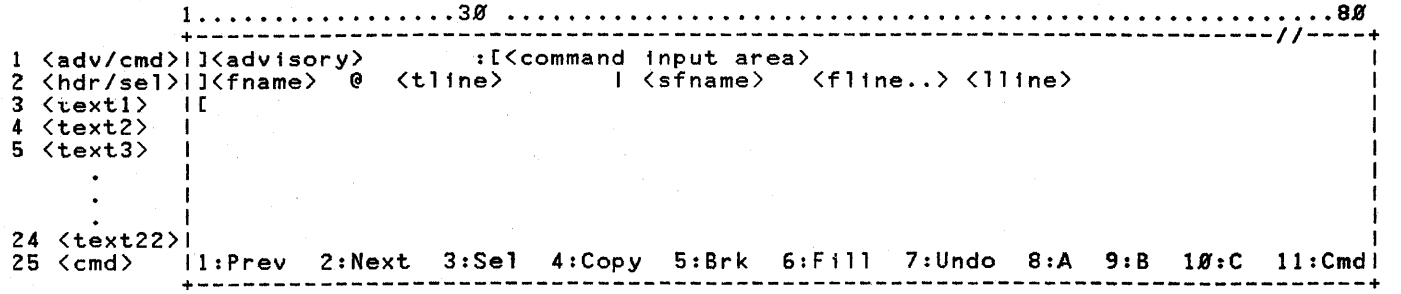


Figure 3. Televideo Display Format

In these illustrations, the following notation is used:

]	Denotes the beginning of an area into which the user cannot enter text (a protected area).
[	Denotes the beginning of an area into which the user can enter text (an unprotected area).
<adv/cmd>	Advisory/command area. <advisory> is where error messages are displayed; <command input area> is where editor commands can be entered.
<hdr/sel>	Window header and selection status area.

The window header consists of three parts:

(1) The name of the file visible in the window.
(2) The currency indicator "@" which is displayed if the window is current.
(3) The line number of the line at the top of

the window.

The selection status area displays the state of the currently selected block of text (see the SELECT command). The selection status area consists of three parts:

(1) The name of the file containing the currently selected block.
(2) The line number of the first line of the currently selected block.
(3) The line number of the last line of the currently selected block.

<textl..n> Area of the screen in which text is displayed. The cursor may be placed anywhere within this block for editing; entering text over existing text automatically replaces that text.

<cmd> Shows the editing functions of the keys above the top row of the keyboard, from left to right.

ASCII File and Work File

The term "ASCII file" refers to a file that is directly readable by such Ridge subsystems as the editor, language compilers, and utility programs. When an existing ASCII file is to be edited, the file is first copied to a "work file." The work file provides rapid, random access to any place in the text; this file is not directly readable by other Ridge subsystems. During the editing session, changes to the text are made only to the work file, not to the ASCII file. At the end of the editing session, the text part of the work file is copied to a new ASCII file.

The work file is saved by default between editing sessions, which provides three benefits: time spent copying the ASCII file to the text file at the beginning of the editing session is eliminated, some contextual information (such as current position in the file and tab settings) is maintained across editing sessions, and the work file is an additional backup to the ASCII file.

In the user's directory, the work file bears a ".e" extension to the original ASCII file name. So, for an ASCII source file named "prime.s", a work file named "prime.s.e" is created.

Selected Blocks

In the Ridge Text Editor, text blocks can be defined and manipulated using several of the predefined operations and editing

commands. The SELECT command is used to define a block--to set its bounds (first and last lines). Blocks can be copied or moved using single keystroke commands. Editing commands and predefined functions can be used locally within a block, thus limiting their actions to just a part of the text file. The CLEAR SELECT command nullifies the selection of the current block so another can be selected.

Used in conjunction with the WINDOW capabilities, blocks of text can be passed from file to file, edited, revised, reinserted, etc. Information about the currently selected block--the file name in which the current block resides, and its bounds--is displayed in the selection status area of the display. (See Figures 2 and 3.)

Windows

The Ridge Text Editor offers three windows (predefined areas of the screen) in which text can appear simultaneously. This means that up to three files of text can be displayed and edited without having to close or save any one of them. Window A occupies the entire screen until window B is created; then the screen is split horizontally between the two. If window C is created, the B window is divided horizontally between these two. Figure 3 shows the screen when all three windows have been opened. Movement from window to window is easily done via the WINDOW keys.

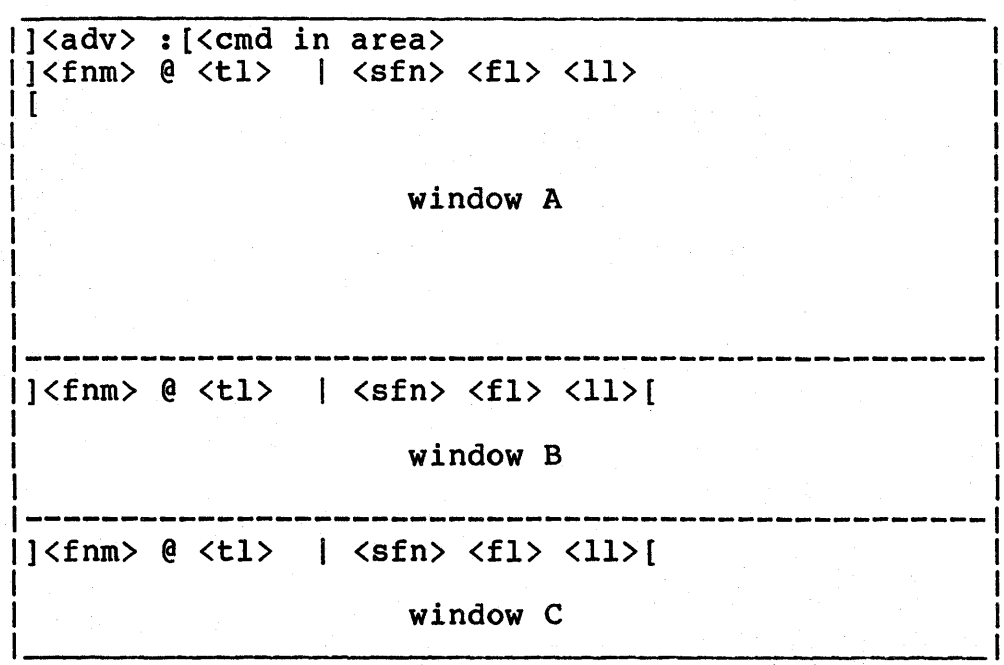


Figure 4. Screen Window Format

SECTION 2

EDITOR COMMANDS AND PREDEFINED OPERATIONS

EDITOR COMMANDS

This subsection defines the editor commands that are typed into the command entry area. The commands are used alone and in combination with one another to provide the editing capabilities. The predefined operations described in the next subsection are implemented using combinations of the commands themselves.

On the Ridge Display, the most frequently used predefined operations appear on the function keys at the top of the keyboard; less frequently used operations are accessed from the keypads to the left and right of the main keyboard. The exceptions to this are the cursor keys which are grouped together. (See Figure 5 for an illustration of the Ridge Display keyboard.)

Syntax Notation

Keyword characters in uppercase and any indicated punctuation are required. Characters shown in lowercase are optional, except that all optional characters must be supplied if any are. Metasymbols are shown surrounded by "<" and ">".

Optional components are surrounded by "[" and "]". If each component in a group is surrounded by "{" and "}", one is to be selected; where a default exists, it is underlined. Lists are indicated by an ellipsis, "...".

The text editor accepts either upper or lower case letters when commands are typed in. Each command element is separated from the other elements by one or more spaces.

Range and Line Specification

<range> is used to specify a block of text to be operated on by an editor command or sequence of commands. <line-spec> is used to specify a single line.

```
<range> = [ ( <filename> ) . ] <line-spec> .. <line-spec>

<line-spec> = { <line number> }  [ { + } <offset> ]
              { @             }    { - }
              { B             }
              { T             }
              { F             }
              { L             }
              { P             }

              { [ F ] }  { 'quoted string' }  [ I ] }
              { [ L ] }  { "quoted string" }  [ W ] }
              { [ N ] }                       [ E ] }
              { [ P ] }                             }
```

Where:

@	Indicates the current line.
B	Bottom line of selected block (towards beginning of the file).
T	Top line of selected block (towards end of file).
F	First occurrence or line.
L	Last occurrence or line.
N	Next occurrence or line.
P	Previous occurrence or line.
B	Both upper and lower case are found.
W	Word only is found (not embedded string).
E	Exact character string is found.
<offset>	May be an integer, or the keyword, PAGELEN. "Pagelen" is the number of lines that can be displayed at once.

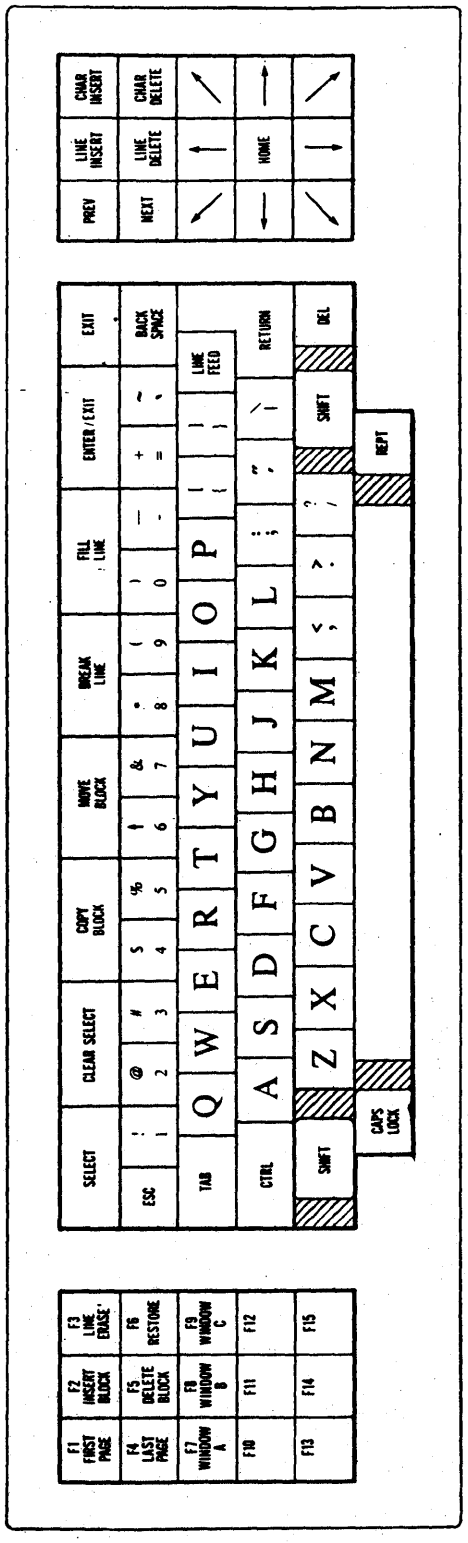


Figure 5. Ridge Display Keyboard

ATTACH Command

```
ATTACH <filename> [, NEW] [, { A } ]
                             -
                           { B }
                           { C }
```

This command places a file in one of the editing windows (A, B, or C). If a file does not currently exist, the NEW option creates one. Normally, this is the first command executed after entering the editor; ATTACH is automatically invoked if a filename is used in calling up the editor itself. (See Section 3 for details about this.)

Example: The following command opens an existing file and attaches it to window A.

```
ATTACH myfile.s
```

Example: The following command opens a new file and attaches it to window B.

```
ATTACH newfile.s, NEW,B
```

CLEAR Command

```
CLEAR { SELECT }
      {  TEMP  }
```

CLEAR SELECT clears the currently selected block. CLEAR TEMP clears the temporary text stack.

CLOSE Command

```
CLOSE <filename> [, [ NO ] ASCII ]
                           -----
                 [, [ NO ] SAVE  ]
                           ----
```

This command detaches the specified file from any window(s) to which it is attached, and closes the file, but leaves the editor still active. If the ASCII option is specified (by default), and the text has been modified since the last time the text file was copied, the work file (".e") is copied to the ASCII file. If NO

ASCII is selected, the work file is not copied to the ASCII file. If the SAVE option is specified (again, by default), the work file is closed. For the NO SAVE option, the work file is purged after its contents are copied to the ASCII file. NO ASCII and NO SAVE cannot be specified simultaneously.

COPY Command

COPY <range>

This command inserts a copy of the specified line(s) just before the current line (the line the cursor is positioned on).

Example: The following command copies lines 1 through 50 from an open file back into itself.

```
COPY 1..50
```

Example: The following command positions a copy of lines 1 through 50 of an open file (named "kit") before the current line of the file in the current window.

```
COPY (kit). 1..50
```

DELETE Command

DELETE

This command deletes the current line. (The RESTORE key does not restore a line so eliminated.)

DETACH Command

```
DETACH { A }
       { B }
       { C }
```

DETACH removes from a specified window, but does not close, a file.

EXIT

EXIT

This command terminates the editor. The equivalent of a CLOSE <filename>, ASCII, SAVE is done for all currently open files.

FIND Command

{ FIND } <line-spec>
{ F }

This command locates a line or string within the text file and makes that the location of the current line. In general, <line-spec> can be a line number (e.g., 10), a quoted string (e.g., "String"), or an occurrence (e.g., N = next, P = previous). If a line number is specified, the cursor is placed within the specified line at the same column position it previously occupied. If, however, a string is specified, the cursor is positioned on the first character of the string. Specifying an occurrence by itself is equivalent to specifying the line number of that occurrence. Specifying an occurrence in combination with a string locates that string relative to the start (F), end (L), or current position (P or N) in the file.

Example: The following command makes line 50 the current line.

FIND 50

Example: The following command makes the previous line the current line.

FIND p

Example: The following command finds the next occurrence of the string "abc".

F n "abc"

INSERT Command

INSERT [<quoted-string>]

INSERT by itself inserts a blank line prior to the current line. With the <quoted- string> option, the specified string is inserted as a new line prior to the current line. The newly inserted line becomes the current line.

Example: The following command inserts the words "Follow these directions." on a line before the present current line.

```
INSERT "Follow these directions."
```

MOVE Command

MOVE <range>

This command inserts a copy of the line(s) specified by <range> just before the current line (the line the cursor is positioned on), and then deletes the original lines.

Example: The following command puts a copy of lines 1 through 50 before the current line and eliminates the original of the lines.

```
MOVE 1..50
```

Example: The following command uses the first occurrence of a string to locate a 10-line block, which is then moved from an open file named "kit," to before the current line in the current window. The original lines are deleted from "kit."

```
MOVE (kit). f "Sec.1" .. f "Sec.1" +10
```

POP Command

POP

This command inserts the top element of the temporary data stack before the current line; the top element is then deleted from the temporary data stack. If the stack is empty, no action takes place.

PUSH Command

PUSH

This command places a copy of the current line on the top of the temporary data stack.

REPEAT Command

```
REPEAT ( <command> [;<command> ]...) [,{ IN <range> [,QUIET]   } ]
                                       { COUNT <num> [,QUIET] }
                                       { QUIET                }
```

REPEAT is used to execute one or more commands repeatedly within an implied or designated range, or until a specified number of iterations are completed.

IN <range> allows iterations to occur within a defined range (a specified group of text lines). If IN <range> is omitted, a range of "1" is implied. If IN <range> is specified, the first line of the range is made the current line prior to the first execution of the command list. If a range of one (e.g., a range of 10..10) is defined, the command list executes one time. If there are no commands that cause the current line to exceed the bounds of the implied or specified range, the REPEAT sequence will continue indefinitely.

The COUNT <num> parameter specifies a limit on the number of times the command list can be repeated.

The QUIET parameter suppresses display update until the REPEAT command has finished iterating.

REPLACE Command

```
REPLACE <quoted-string>
```

This command replaces the currently selected text with <quoted-string>. The text can be explicitly replaced by the execution of a FIND command used in conjunction with REPLACE. Or text can be implicitly replaced based on the current position of the cursor: if the cursor is currently positioned on a string that has just been located by a FIND string search, that string is replaced by the quoted string. Otherwise, <quoted-string> is inserted into the current line beginning at the cursor position.

Example: The following command list replaces all occurrences of the string "abc" with the string "xyz". (Note that the EXECUTE/ ENTER key must be pressed to go from one "FIND-REPLACE" sequence to the next.)

```
REPEAT (FIND N 'abc' ; REPLACE 'xyz')
```

SELECT Command

SELECT

This command uses the current line (where the cursor is positioned) to define the bounds (the first and last lines) of a text block, which becomes the currently selected block.

If no block is currently defined, SELECT defines a block of one line (the current line becomes both the first and last bound). If the cursor is moved to a line outside the current bounds, SELECT redefines the block (its first or last bound is changed). Scrolling up (towards the beginning of the file) causes the last line to be changed; scrolling down (towards the end of the file) causes the first line to be changed. If the cursor is moved to a line within the defined block, pressing SELECT does not redefine the current block bounds.

If the current line is not in the same window as the currently selected block, SELECT nullifies the current selection and sets the bounds for a one-line block.

SET TAB Command

SET TAB <col> [<col>] ...

This command can be used to override the default display tabs (which are set every eight spaces). A maximum of 15 tabs can be defined. Since this is a two word command, and since a series of numbers may be given, spacing is significant.

Example: The following command sets up three tabs. The first ranges from column 1 to 10, the next from 10 to 20, the last from 20 to 40.

```
SET TAB 10 20 40
```

WINDOW Command

```
WINDOW { A }
       { B }
       { C }
```

The WINDOW command selects the current window for editing. The selected window must have a file attached to it.

XEQ Command

XEQ <filename>

This command is used to execute a series of editor commands that are contained in an ASCII file.

Example: Assume a file ("tree") in window A is being edited. Then, to create and edit an XEQ file named "sap" that will open an existing file "leaf" in window B, insert a text string at a specified line in "leaf," and then go back to a specified line in "tree," the following sequence could be used. First, the file "sap" is created. It contains:

```
ATTACH leaf,b
FIND 4
INSERT 'maple sugar'
WINDOW A
FIND 1
```

Next, "tree" is attached to window A. Then the command sequence:

```
XEQ sap
```

causes the desired search, string insertion, and return to line 1 in "tree."

PREDEFINED OPERATIONS

This subsection describes the predefined operations for the Ridge Display and the Televideo TVI-950C. These operations are invoked by pressing one of the terminal editing keys or one of the functions keys, either unshifted or in combination with the SHIFT key.

On the Ridge display, function keys F1 through F9 have been assigned editor operations; keys F10 through F15 are unassigned. On the Televideo, all available function keys (F1 through F11) have been assigned editor operations.

The following descriptions of the predefined operations are arranged alphabetically. Figure 5 represents the keyboard for the Ridge Display, and Table 1 shows which keys on the Ridge Display and the Televideo correspond with which operations, as well as providing the key sequences the terminals will also accept for some operations.

Table 1. Predefined Operations

PREDEFINED OPERATION	TELEVIDEO KEY	RIDGE DISPLAY KEY	ALTERNATE KEY SEQUENCE
Backspace	Backspace	Backspace	ctrl-H
Break Line	F5 Unshifted	Break Line	
Char Delete	Char Delete	Char Delete	
Char Insert	Char Insert	Char Insert	
Clear Select	F3 Shifted	Clear Select	
Copy Block	F4 Unshifted	Copy Block	
Cursor Diag LD		↙	
Cursor Diag LU		↖	
Cursor Diag RD		↘	
Cursor Diag RU		↗	
Cursor Down	↓	↓	
Cursor Left	←	←	
Cursor Right	→	→	
Cursor Up	↑	↑	
Delete Block	Line Del Shft	Delete Block (F5)	
Enter/Execute	F11 Unshifted	Enter/Execute	
Exit	F11 Shifted	Exit	
Fill Line	F6 Unshifted	Fill	
First Page	F1 Shifted	First Page (F1)	
Home	Home	Home	
Insert Block	Line Ins Shft	Insert Block (F2)	
Line Delete	Line Delete	Line Delete	
Line Erase	Line Erase	Line Erase(F3)	
Line Insert	Line Insert	Line Insert	
Last Page	F2 Shifted	Last Page (F4)	
Move Block	F4 Shifted	Move Block	
Next Page	F2 Unshifted	Next	
Previous Page	F1 Unshifted	Prev	
Restore (Undo)	F7 Unshifted	Restore (F6)	
Return	Return	Return	ctrl-M
Select	F3 Unshifted	Select	
Tab	Tab	Tab	ctrl-I
Window A	F8 Unshifted	Window A (F7)	
Window B	F9 Unshifted	Window B (F8)	
Window C	F10 Unshifted	Window C (F9)	

BACKSPACE

This operation moves the cursor back one position. If the cursor is in column 1, it is moved to the last column of the previous line.

BREAK LINE

This operation splits the current line at the cursor position, creating a new line containing the text trailling the cursor.

CHARACTER DELETE

This operation deletes the character at the cursor position. Text to the right of the cursor on the same line is moved to the left.

CHARACTER INSERT

This operation inserts a space just before the cursor position. Text to the right of the cursor on the same line is moved to the right.

CLEAR SELECT

This operation performs the same function as the editing command, CLEAR SELECT. It nullifies the currently selected block.

COPY BLOCK

This operation inserts a copy of the currently selected block just before the current line, and then nullifies the current selection.

CURSOR DIAGONAL LEFT, DOWN

This operation moves the cursor diagonally, down to the next line and left one column.

CURSOR DIAGONAL LEFT, UP

This operation moves the cursor diagonally, up one row and left one column.

CURSOR DIAGONAL RIGHT, DOWN

This operation moves the cursor diagonally, down one row and right one column.

CURSOR DIAGONAL RIGHT, UP

This operation moves the cursor diagonally, up one row and right one column.

CURSOR DOWN

This operation moves the cursor down one line.

CURSOR LEFT

This operation is the same as a backspace.

CURSOR RIGHT

This operation moves the cursor to the right one column. If the cursor is in the last screen column, it is moved to column 1 of the next line.

CURSOR UP

This operation moves the cursor up one column.

DELETE BLOCK

This operation deletes the lines in the currently selected block, then nullifies the current selection.

ENTER/EXECUTE

This operation puts the cursor in the command input area (if it is not already so positioned). If the cursor is in the command input area, pressing the key becomes a signal to interpret the command line for execution.

To NOT execute a command sequence that has been entered into the command input area, and assuming that the cursor is positioned in the area, press the appropriate window key to move the cursor to the desired current window.

EXIT

This operation performs the same function as the EXIT command: files that are currently open are copied to disc (ASCII files), work files are saved (".e" files), and the editing session is terminated.

FILL LINE

This operation attempts to place as many words as possible on a 70 character line. For short lines, words are taken from the next text line. To truncate a too lengthy line, words are moved to a new line (which becomes the current line).

FIRST PAGE

This operation moves the window back to the first page in the text file. A page represents the number of lines that constitute a "screenful": 47 lines for the Ridge Display, and 22 for the Televideo. The first text line becomes the current line.

HOME

This operation positions the cursor in the upper left hand corner of the text entry area (column 1, row 1) of the current window.

INSERT BLOCK

This operation effectively pushes the current line to the bottom of the window by inserting a block of blank lines before it.

LAST PAGE

This operation moves the window forward to the last page in the text file. The last text line becomes the current line.

LINE DELETE

This operation deletes the current line. (RESTORE restores a line deleted by this operation.)

LINE ERASE

This operation replaces the characters from the cursor position to the end of the line with blanks.

LINE INSERT

This operation inserts a blank line before the current line. The blank line becomes the new current line.

MOVE BLOCK

This operation inserts a copy of the currently selected block before the current line, deletes the lines in the currently selected block, and then nullifies the current selection.

NEXT PAGE

This operation moves the window forward to the next page in the text file. The new current line occupies the same relative position in the window as did the old current line.

PREVIOUS PAGE

This operation moves the window back to the preceding page in the text file. The new current line occupies the same relative position in the window as did the old current line.

RESTORE (UNDO)

This operation restores the last line deleted by the LINE DELETE key by inserting the line just before the current line. The restored line becomes the new current line.

RETURN

This operation moves the cursor to the beginning of the next line. If the next line is not displayed, the window is moved forward by one line.

SELECT

This operation and the SELECT command perform identically. See the Editing Command subsection for a full description; briefly, SELECT identifies the current line by line number, and defines upper and lower bounds for a block of text.

TAB

This operation moves the cursor to the next predefined tab position. If TAB is executed past the last identified tab, the cursor is moved to column 1 of the next line. If the next line is not displayed, the window is moved forward in the text file one line.

WINDOW A
WINDOW B
WINDOW C

These operations select the specified window as the current window. They are identical to the WINDOW editing commands.

SECTION 3

TUTORIAL

INTRODUCTION

This tutorial takes the user through the basics of creating and editing a text file. No attempt is made to illustrate all the features of the text editor, simply some basics. While the Monochrome Display and the Text Terminal support the text editor in the same way, display format differences were shown in Chapter 1; and keyboard and functional differences were noted in Chapter 3. In this tutorial a few further differences are encountered and explained.

INVOKING THE EDITOR

To invoke the text editor on either the Monochrome Display or Text Terminal, type after the operating system prompt "$":

```
$ redit
```

The screen now displays in the first line (the advisory and command area line) the following:

```
Ridge Text Editor (07-Sep-83):
```

The date indicates the version of the editor. The cursor appears after the colon in the command input area.

ATTACHING A FILE

ATTACH brings an existing file to the screen--or it can be used to create a new file. To edit an existing file ("stengel"), the command line should look like this:

```
Ridge Text Editor (07-Sep-83): ATTACH stengel
```

As is true for all commands entered into the command status area, you must press the ENTER/EXECUTE key when you want the command to be executed.

The file now appears on the screen, in window A by default. Window A occupies the entire screen unless windows B and/or C are attached; Figure 4 in Chapter 2 illustrates how the screen is subdivided.

To position the file in one of the other windows, specify ",b" or ",c" after the file name:

```
Ridge Text Editor (07-Sep-83): ATTACH stengel,b
```

To create a new file (named "gold"), ",new" must be added after the file name:

```
Ridge Text Editor (07-Sep-83): ATTACH gold,new
```

However, the quickest way to start the editor and get going on a file is to use a file name when invoking the editor. This causes ATTACH to be executed automatically. To create a new file, type:

```
$ redit gold,new
```

Windows can, of course, be specified. To call "stengel" into window C, you would type:

```
$ redit stengel.c
```

After the ATTACH command has been executed, a banner appears in the window header and selection status line (the second screen line) of the display. For the new file "gold", this line shows the file name, the currency indicator ("@") since the window is the current one, and the current line number:

```
Ridge Text Editor (07-Sep-83):ATTACH gold,new
gold              @         1
```

Text can now be entered in typewriter fashion. Assume the following three line file is entered:

```
Ridge Text Editor (07-Sep-83):ATTACH gold,new
gold              @         1
The words of Samual Goldwyn himself, best illustrate the
goldwynism, a type of mixed metaphor: "No oral contract is
worth the paper it's written on."
```

As a strike-over editor, corrections can be made by simply writing over existing text. On the Monochrome Display, pressing the REPT key simultaneously with a cursor key, speeds up cursor movement.

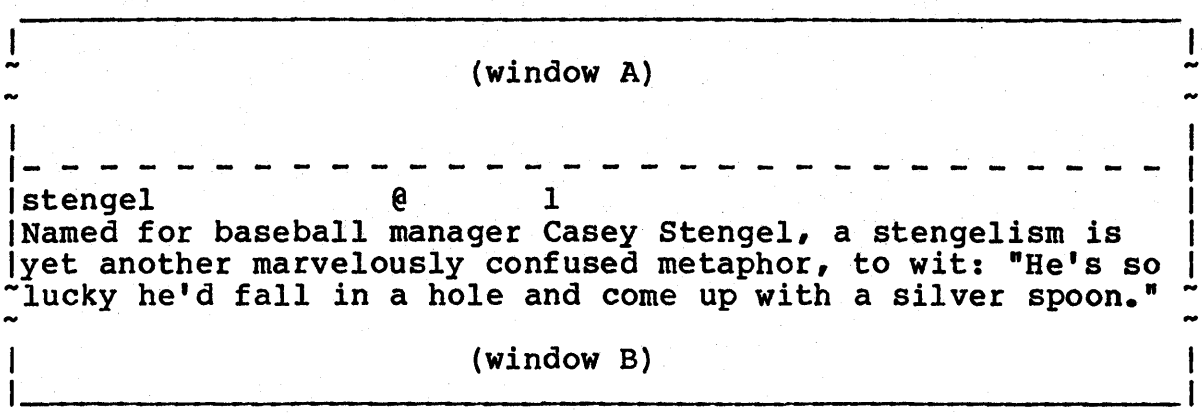
BLOCK MANIPULATION

Assume now that the existing "stengel" file should be appended to "gold". Use the ENTER/EXIT key to enter the command input area and then type:

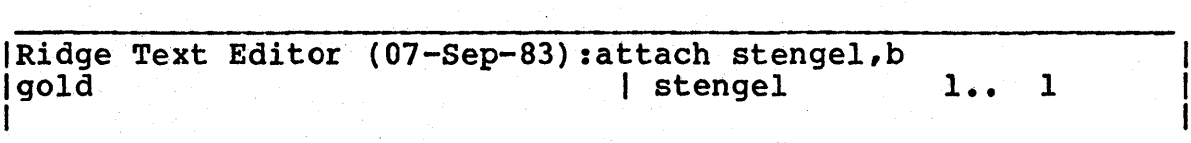
```
Ridge Text Editor (07-Sep-83):attach stengel,b
```

After pressing the ENTER/EXIT key again, the following appears

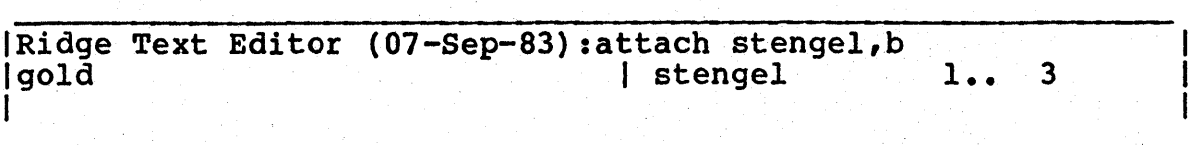
in window B:



The window's status line gives the file name, currency status indicator (B is now the current window), and the current line number ("1"). Editing is performed in the B and C windows in the same way as it is in window A. So the file could be added to, deleted from, etc. The easiest way to copy the file to "gold" is to use the COPY command. First, the line numbers of the text to be copied must be determined. Since window B is the current window, position the cursor on the first line of the file to be copied (line 1), and press SELECT. The window A status lines looks like this:



Note that the currency indicator is absent from window A and that no top line is indicated for that window. "stengel" is shown to have a current block defined in it (a block can be one line). The "1.." represents the top line of the block, and "1" the bottom line of the block. Move the cursor to the last text line to be copied (line 3). Press SELECT. The window A status line now appears as:



The bottom line is now seen to be 3. Next, return to window A

by pressing the WINDOW A key, and locating the cursor on the line below which the copied text should appear (here, it will be line 4). Next, simply press the COPY key. The following now appears in window A:

```
|Ridge Text Editor (07-Sep-83):                                    |
|gold          @       1                                           |
|The words of Samual Goldwyn himself, best illustrate the          |
|goldwynism, a type of mixed metaphor: "No oral contract is        |
|worth the paper it's written on."                                 |
|Named for baseball manager Casey Stengel, a stengelism is         |
|yet another marvelously confused metaphor, to wit: "He's so       |
|lucky he'd fall in a hole and come up with a silver spoon."       |
```

If the "stengel" portion of the text should appear before the "gold" portion of the file, the MOVE <range> command can be used in just the same way as the COPY <range> was used. The difference between the two is that MOVE deletes the original lines. MOVE BLOCK could also be used to reorganize the text. It is similar to COPY BLOCK except that it deletes the original lines. If this were a longer file and the word "stengelism" were misspelled throughout, the REPEAT command could be used to perform a global search and replace. Enter the command input area and type:

```
repeat (find "stengalism"; replace "stengelism") count -1
```

Also, ", quiet" can be added onto the command sequence so that the text manipulation is done without updating the screen after each change.

LEAVING THE EDITOR

A file can be closed and the editor terminated by using the EXIT key. All open files are saved, updated, and control is returned to the operating system.

COMMENTS ON ASCII AND WORK FILES

If an ASCII file is lost (for example, moved to another directory), the file may still be edited in the current directory as long as the work file exists. Do not invoke the file by its ".e" name, however, since the work file is not readable by the editor; rather use the ASCII file name ("prime.s"); make some sort

of change to the file; then save the file. Now, both an ASCII and a work file should appear in the directory.

A work file appears not to be updated if it is manipulated by non-redit commands (e.g., with CAT or vi). For instance, a file "red", initially created using redit, is modified using vi. If the file "red" is concatenated out onto the screen, changes made using vi appear. However, if file "red" is invoked with redit, the original version appears. Therefore, to call up the most recent version, first remove the file "red.e" from the directory.

Ridge Computers
Corporate Headquarters

2451 Mission College Blvd.
Santa Clara, California 95054
Phone: (408) 986-8500
Telex: 176956