A RAND NOTE

THE RAND EDITOR, e: VERSION 19

Lynn Anderson, Charles L. Batten

February 1986

N-2239-1-RCC

Prepared for

The Rand Computation Center

Rand
1700 MAIN STREET
P.O. BOX 2138
SANTA MONICA, CA 90406-2138

Published by The Rand Corporation

A RAND NOTE

THE RAND EDITOR, e: VERSION 19

Lynn Anderson, Charles L. Batten

February 1986

N-2239-1-RCC

Prepared for

The Rand Computation Center

Rand
1700 MAIN STREET
P.O. BOX 2138
SANTA MONICA, CA 90406-2138

PREFACE

This reference manual describes many of the capabilities available in Version 19 of the Rand editor, e. It supersedes Rand Note N-2239-RCC, which described Version 17 of e.

The level of description presented here assumes that the user has prior experience with an on-line text editor. Individuals without such experience should first read the *Self-Teaching Guide to Rand's Text Processor*, N-2056-1-RCC.

The authors wish to thank Susan Payne, Rosalind Chambers, and Gail Roberts for their extremely helpful technical reviews of this document.

CONTENTS

HOW TO USE THIS REFERENCE MANUAL

This manual uses different typefaces and symbols to denote the keys you must press on your keyboard.

TYPEFACES

bold Bold indicates that you must type characters exactly as they are shown. Thus, if you are told to type

exit

you must type **exit** to exit from the file.

italics Italics indicate that you must type characters like the ones that are shown. They may be the name of a file or a number. Thus, if you are told to type

e *newfilename*

you must type the letter **e** exactly as shown and a filename like *newfilename*.

<Brackets> Angled brackets indicate that you must press a key with a particular label on it. Thus, if you are told to press

<Cmd>

you must press the key with the label Cmd on it.

<CR> This indicates that you must press the return or enter key.

<Key+Key> This indicates that you must hold down the first key and then press the second one. Thus, if you are told to press

<Ctrl+b>

you must hold down the Ctrl key and then press the b key.

SYMBOLS

▮	The "block" is used in conjunction with other characters to make the printer perform special tasks when printing your text. The block symbol appears as a bright box on your screen, and it looks like a ▮ in this document. To insert a block symbol in your file, press <CntlChar> if you have an Ambassador or XL terminal or <Ctrl+\> if you have an older Ann Arbor terminal.

1. INTRODUCTION

The Rand editor, known as e, lets you type, view, and change files in an easy and natural way. Your terminal screen shows a portion of your file at a time, and you can move other portions into this "window" at will. You can type new characters and change existing ones. In addition, the editor lets you insert, delete, and move characters, lines, and rectangular portions of your file, much as you would use scissors and tape to edit a paper document.

This document serves as a reference manual for Version 19 of the Rand editor. Section 2 describes the basic concepts you should master before using the editor. Section 3 describes editor commands, and Section 4 the editor function keys. The appendixes discuss how to deal with crashes, the function of editor work files, variations of the editor's cover command, special options that can be added to UNIX[1] commands, and how to use different terminals and keyboards.

---

[1]UNIX is a trademark of Bell Laboratories.

2. CONCEPTS

To use the editor, you first need to know some basic concepts concerning files you edit, the buffers in which you store text, and the comma files which contain the previous version of the file you are editing.

WHAT ARE FILES?

A file is a collection of text saved by the Text Processor.

Naming a File

You must give each file an individual identifying filename when you create it. The filename you choose:

- Can contain up to 255 letters, numbers, dots, and dashes.[1]
- Cannot contain a blank space, because the editor will ignore all characters typed after the space.
- Should not begin with a comma, because this convention is for special kinds of files that are automatically deleted from the system 24 to 48 hours after their creation.
- Generally should not begin with a period, because files beginning with periods are "hidden" and are not listed with the standard files.
- Must not be named **core** or **a.out**, because these files are automatically deleted from the system 24 to 48 hours after their creation.

Editing a File

To edit a file, type **e** *filename* <CR> in response to the % prompt. If the file does not already exist, respond to the prompt "Do you want to create *filename*?" by typing **y** for yes.

You should limit the text in your files to 1000 lines, and in most cases you should type text and formatting macros beginning in column 1 of those lines.

The editor uses the window concept to let you view your file. The terminal screen is like a window that opens on part of the file. You

[1]The full pathname can contain up to 1024 letters, numbers, dots, and dashes.

can use editor function keys to move the window forward and backward (and right and left) to view different portions of your file.

The window normally shows columns 1 through 78 in your file. When the cursor reaches column 78, the window automatically moves 16 spaces to the right, moving any text at the left margin out of sight. To return your cursor to the left margin of your text, press <CR>.

On the screen:

- Hyphens (-) outline the top and bottom edges of the window.
- Pluses (+) indicate:
  - The four corners of the window.
  - The location of tab stops where they appear on the top edge.
  - The right margin where one appears at the right of the bottom edge.
  - The left margin where one appears at the left of the bottom edge.
- Colons (:) appear at the left and right edges of the window before you have reached the end of your file and at the right edge of the window after you have reached the end of your file.
- Semi-colons (;) appear at the left edge of the window when you have reached the end of your file.
- The greater-than character (>) appears at the right edge of the window when there is text that cannot be seen to the right of the window.
- The less-than character (<) appears at the left edge of the window when there is text that cannot be seen to the left of the window (this occurs when the window has been moved to the right).
- Periods (.) outline windows other than the window in which you are currently working.

Exiting and Saving a File

When you want to save your file and leave the editor, press <Cmd> **exit** <CR>. You should save your file at least once every half hour.

The Cursor

The cursor indicates the position where the next character typed will be placed in your file (and where it will print). Most commands take place at or following the cursor position. Depending on the kind of terminal you use, the cursor resembles either an underline character (_) or a block character (▉). In addition, bright blocks (▉) sometimes appear on the four edges of the screen to target the column and line where the cursor is located.

You can move the cursor by pressing <→>, <←>, <↑>, <↓>, <Bsp>, <Home>, <+Line>, <-Line>, <LineFeed>, <+Page>, <-Page>, <CR>, <Spacebar>, <+Tab>, <-Tab>, <+Word>, and <-Word>, and by using certain command sequences involving <+Sch> and <-Sch>.

Alternating Files

The editor allows you to alternate between several files without exiting. This enables you to move text between files and compare files. To move to an alternate file, press <Cmd> **e** *newfilename* <CR>. To move back to the original file, press <Alt> or <Ctrl+b>. (See description in Section 4.)

Using Changes Files, #o and #p, to Restore Lost Text

The editor maintains two "changes" files to help protect you from losing material that you have accidentally deleted or changed. These files are automatically deleted each time you exit from the editor:

- The #o file contains all text centered, closed, filled, joined, justified, split, and deleted with <DelWord>, <Cmd> <Del>, or <Cmd> <Bsp> during the current editing session.

- The #p file contains all text picked during the current editing session.

To use these files, press <Cmd> **e #o** <CR> or <Cmd> **e #p** <CR>. This brings the changes file into the editing window as an alternate file. You may then pick and move text to the original file. Caution: Do not use <Close> to move material from these files.

WHAT ARE BUFFERS?

Buffers are temporary storage areas that exist only while you are in the editor. The editor uses five buffers to store text for special purposes: the adjust, close, erase, pick, and search buffers.

Adjust Buffer

The adjust buffer contains the original version of the last material you centered, filled, or justified. Text is placed in the adjust buffer when you press <Cmd> **center** <CR>, <Cmd> **fill** <CR>, or <Cmd> **justify** <CR>. To retrieve text from the adjust buffer, press <Cmd> **insert adjust** <CR>. Material remains in the adjust buffer and may be retrieved repeatedly until you center, fill, or justify different material or until you exit from the file.

Close Buffer

The close buffer contains the last material you closed. Text is placed in the close buffer when you press <Close>, <Cmd> **close** <CR>, or <Cmd> <Bsp> while in the Insert Mode. To retrieve text from the close buffer, press <Cmd> <Close> or <Cmd> **-close** <CR>. Material remains in the close buffer and may be retrieved repeatedly until you close different material or until you exit from the file.

Erase Buffer

The erase buffer contains the last material you erased. Text is placed in the erase buffer when you press <Erase>, <Cmd> **erase** <CR>, or <Cmd> <DelChar>. To retrieve text from the erase buffer, press <Cmd> <Erase> or <Cmd> **-erase** <CR>. Material remains in the erase buffer and may be retrieved repeatedly until you erase different material or until you exit from the file.

Pick Buffer

The pick buffer contains the last material you picked. Text is placed in the pick buffer when you press <Pick> or <Cmd> **pick** <CR>. To retrieve text from the pick buffer, press <Cmd> <Pick> or <Cmd> **-pick** <CR>. Material remains in the pick buffer and may be retrieved repeatedly until you pick different material or until you exit from the file.

Search Buffer

The search buffer contains the last material for which you searched. Text is placed in the search buffer when you press <Cmd> *word* <+Sch> or <Cmd> *word* <-Sch>, or when you press <Cmd> <+Sch> or <Cmd> <-Sch> when the cursor is on the first character of the word being searched for. Material remains in the search buffer until you search for different material or until you exit from the file. If, however, you return to the file by typing **e** <CR> in response to the % prompt, the material you last searched for remains in the search buffer.

WHAT ARE COMMA OR BACKUP FILES?

A comma file is a temporary copy of the file as it existed before the most recent editing session. When you exit normally, the editor renames the original file by preceding the original name with a comma.

For example, the editor backs up a file named *memo* (the current file) with a file named *,memo* (the previous version of this file). The comma file is stored in the same directory as the modified file. All comma files are automatically deleted 24 to 48 hours after they are created.

Comma files protect you from inadvertent mistakes. If you have made a serious error when you worked on the file, you can either:

- Exit with the abort option (<Cmd> **exit abort** <CR>) and make the comma file into a permanent file (% **cp** *,memo memo* <CR>); or

- Without exiting from the editor, you can edit the comma file as an alternate file (<Cmd> **e** *,memo* <CR>) and use the pick key or pick command to copy into your permanent file the section of the file as it existed before the editing error.

If you exit with the abort option or if the system crashes, no comma file is created and any previous comma file is left unchanged.

3. COMMANDS ISSUED FROM WITHIN THE EDITOR

This section describes the commands you issue to the editor to perform such functions as inserting, deleting, and rearranging text.

Occasionally, you can perform the same function either by using an editor command (described in this section), or by pressing an editor function key (described in Section 4). For example, you can remove the line on which the cursor is located either by pressing <Cmd> **close** <CR> or by pressing the <Close> key.

Abbreviating Editor Commands

Most editor commands can be abbreviated to their initial one, two, or three characters as long as these characters are not identical to those of some other editor command. For example, the window command can be abbreviated as w, wi, win, or even wind and windo. The join command can be abbreviated as jo and joi, but not j, because the justify command also begins with j.

A few commands, such as tab, cannot be abbreviated.

If you type an inadequate number of characters or if you abbreviate a command that cannot be abbreviated, the editor will display the following error message: "*** Command '...' ambiguous".

Defining Areas for Editor Commands

An area is any part of a file; it can be as small as one character or as large as the entire file.

An area may be a default area, such as the default area for the fill command (one paragraph following the cursor position).

You may also define an area by using:

- A number in conjunction with the letter l (ell) to define lines or the letter p to define paragraphs.
- An explicit description of the dimensions (lines x columns).
- The mark function in conjunction with cursor movements.

For example, <Cmd> **fill** <CR> fills two lines following the cursor and <Cmd> **center 2p** <CR> centers two paragraphs following the cursor.

To close a rectangular area three lines long by four columns wide following the cursor position, explicitly describe the dimensions by pressing <Cmd> **close 3x4** <CR>.

To fill an area defined by the mark function, press <Mark>, move the cursor, and then press <Cmd> **fill** <CR>.

BOX: DRAWING A BOX

The box command draws a box around a specified area. To draw a box:

1. Place the cursor at the left edge of the area and press <Mark>.
2. Move the cursor on the same line to the right edge of the area by pressing <→>.
3. Move the cursor to the bottom of the area by pressing <↓>. (The number of lines and columns marked off is indicated in the mark message below the window, e.g., "MARK 4x32".)
4. Press <Cmd> **box** <CR>.

This command draws a box that consists of:

- Hyphens (-) on the marked horizontal lines
- Pipes (|) on the marked vertical lines
- Pluses (+) at each corner.

CAUTION: The box command erases characters on the lines where it inserts the "-", "|", and "+" characters; it does not, however, erase characters within the box.

You can also use the box command to draw horizontal and vertical lines. To draw a horizontal line:

1. Press <Mark>.
2. Move the cursor to the end of the line by pressing <→>.
3. Press <Cmd> **box** <CR>.

To draw a vertical line:

1. Press <Mark>.
2. Move the cursor one space to the right by pressing <→>.
3. Move the cursor to the point where you want the line to end by pressing <↓>.
4. Press <Cmd> **box** <CR>.

To remove a box:

1. Press <Mark>.
2. Move the cursor to mark the dimensions of the box.
3. Press <Cmd> **blot box** <CR>.

This leaves blank spaces where the box outline had been drawn.

<Cmd> **box** <CR>	Places box around marked area
<Cmd> **blot box** <CR>	Deletes box around marked area

CAPS: CHANGING ALL LETTERS TO UPPERCASE

The caps command changes all letters to uppercase in a rectangularly marked area. To use this command, place your cursor at the beginning of the area you want to change, press <Mark>, move the cursor, and press <Cmd> **caps** <CR>.

<Cmd> **caps** <CR>	Changes all letters to uppercase in a rectangularly marked area

CCASE: REVERSING LETTERS TO UPPER/LOWER CASE

The ccase command changes all uppercase letters to lowercase and all lowercase letters to uppercase in a rectangularly marked area. To use this command, place your cursor at the beginning of the area you want to change, press <Mark>, move the cursor, and press <Cmd> **ccase** <CR>.

<Cmd> **ccase** <CR>	Changes all uppercase letters to lowercase and all lowercase letters to uppercase in a rectangularly marked area

CENTER: CENTERING TEXT

The center command centers text in your file. To center the line where the cursor is located, press <Cmd> **center** <CR>. To center several lines, press <Mark>, move the cursor to the last line to be centered, press <Cmd> **center** <CR>. You can also press <Cmd> **center** *n* <CR>, where *n* is the number of lines to be centered. (You cannot center a rectangularly marked area.)

The line length of the center command is the default width of 75, the width specified by the last set right margin command (**set rmar** *n*) <CR>, or the width specified in the last center, fill, or justify command. To specify a width, press <Cmd> **center width=***n* <CR>, where *n* is the number of characters per line.

If you have made a mistake when centering text, press <Cmd> **insert adjust** <CR>. This command will insert the original uncentered version of the text at the cursor position. You must then delete the mistakenly centered text. You can retrieve this uncentered text only while you are in the current editing session and only before you issue another center, fill, or justify command.

<Cmd> **center** <CR>	Centers line
<Cmd> **center** *n* <CR>	Centers specified number of lines
<Cmd> **center width=***n* <CR>	Centers with specified width
<Cmd> **insert adjust** <CR>	Restores centered text

CLOSE, -CLOSE: REMOVING/RESTORING TEXT

The close command removes characters, lines, or rectangular areas and moves remaining text so that no gaps are left in your file. (By contrast, the erase command removes characters, lines, or rectangular areas and leaves a blank space where they were.) Text removed by the close command is placed in the close buffer.

To close the line where the cursor is located, press <Cmd> **close** <CR>. To close a number of lines or a rectangular area, press <Mark>, move the cursor, and press <Cmd> **close** <CR>. You can also close lines by pressing <Cmd> **close** *n* <CR>, where *n* is the number of lines to be closed.

To restore the last text you closed, press <Cmd> **-close** <CR>. Copies of the closed text can be inserted as many times and in as many places as you wish.

Any subsequent use of the close command will replace the text in the close buffer. Note: Pressing <Cmd> <Bsp> while in Insert Mode also replaces the contents of the close buffer with the newly deleted text.

All material removed from a file by pressing <Cmd> **close** <CR> or <Cmd> <Bsp> during your current editing session is placed in the changes file (#o).

<Cmd> **close** <CR>	Closes line
<Cmd> **close** *n* <CR>	Closes specified number of lines
<Cmd> **-close** <CR>	Restores closed text

COMMAND, -COMMAND: ISSUING A SERIES OF COMMANDS

Usually commands are issued one at a time by pressing <Cmd>, typing the command, and pressing <CR>. If you wish to issue a series of commands without pressing <Cmd> each time, press <Cmd> **command** <CR>. The message "CMDS:" will appear below the screen. The editor remains in the constant Command Mode until you type <Cmd> **-command** <CR>.

You cannot change the cursor position once you are in the constant Command Mode.

<Cmd> **command** <CR>	Places editor in constant Command Mode
<Cmd> **-command** <CR>	Takes editor out of constant Command Mode

COVER: COVERING EXISTING TEXT

If you use <Cmd> <Pick> or <Cmd> **-pick** <CR> to insert rectangularly marked material into your file (see the discussion of the pick command below), all existing material is moved to the right.

You can, however, cover or write over existing material rather than move it to the right, by using the cover command. This command is especially useful when you are changing columns in a table or moving portions of a figure.

For example, suppose you have a table like the one below, and you wish to replace the numbers in the "Percent" column, using numbers from another part of your file:

	Percent	Number
Alabama	18	12,000
California	23	16,500
Maryland	17	11,250
New York	24	9,500
Oregon	20	14,000
Virginia	43	23,500

To copy the material you want to use, place your cursor at the upper left corner of the material, press <Mark>, move the cursor, and press <Pick>. The marked area must be rectangular. The material you pick might be the following numbers which you want to insert in the "Percent" column:

23
12
16
21
19
39

Place your cursor on the first figure of the first number in the "Percent" column and press <Cmd> **cover pick** <CR>. Your file will now look like:

	Percent	Number
Alabama	23	12,000
California	12	16,500
Maryland	16	11,250
New York	21	9,500
Oregon	19	14,000
Virginia	39	23,500

Everything in your buffer, both characters and blank spaces, now covers text in your file. Nothing is moved to the right. You can also use the cover command with rectangularly marked material stored in the close and erase buffers: <Cmd> **cover close** <CR>, or <Cmd> **cover erase** <CR>.

CAUTION: When issuing the cover command, you must type the name of the buffer you use; do not press the key with the name of the buffer on it, such as <Close>, <Erase>, or <Pick>.

Appendix C describes other ways to insert rectangularly marked material from your close, erase, and pick buffers without shifting existing text to the right.

<Cmd> **cover close** <CR>	Uses contents of close buffer to cover portion of file
<Cmd> **cover erase** <CR>	Uses contents of erase buffer to cover portion of file
<Cmd> **cover pick** <CR>	Uses contents of pick buffer to cover portion of file

DELETE: DELETING THE CURRENT FILE WITHIN THE EDITOR

To delete the file you are currently editing (in both the editor and the UNIX file system), press <Cmd> **delete** <CR>. The file is deleted immediately and you are moved either to the alternate file or to a file called *scratch*. When you exit from the alternate file or from *scratch*, the system responds with "DELETE: *filename*".

CAUTION: Make absolutely certain you want to delete the file. The delete command takes effect immediately. There is no way of retrieving a file once you have deleted it in this fashion.

<Cmd> **delete** <CR>	Deletes current file

DWORD, -DWORD: DELETING/RESTORING A WORD

To delete a single word or a series of characters up to a blank space and fill in the space left by the deletion, place the cursor on any letter or on the space immediately preceding the characters and press <Cmd> **dword** <CR>.

To restore the word or characters deleted in this fashion, press <Cmd> **-dword** <CR>.

<Cmd> **dword** <CR>	Deletes word at cursor position
<Cmd> **-dword** <CR>	Restores deleted word

E: EDITING A FILE

To enter the editor and edit a file, type **e** *filename* <CR> in response to the % prompt.

To enter the editor and edit the file you most recently edited, type **e** <CR> in response to the % prompt.

If you are already editing a file, you can edit an additional file (as an alternate file) by pressing <Cmd> **e** *newfilename* <CR>.

If the file does not already exist, respond to the prompt "Do you want to create *newfilename*?" by typing **y** for yes or **n** for no.

If you are editing a file in a subdirectory and wish to edit as an alternate file a file in your main (or login) directory, press <Cmd> **e** *~/filename* <CR>. The tilde (~) serves as an abbreviation for either */a/loginname* or */b/loginname*.

<Cmd> **e** *filename* <CR>	Brings up an alternate file
% **e** *filename* <CR>	Brings up a named file
% **e** <CR>	Brings up file most recently edited

ERASE, -ERASE: ERASING/RESTORING TEXT

The erase command removes characters, lines, or rectangular areas and leaves a blank space where they were. (By contrast, the close command removes characters, lines, or rectangular areas and moves the remaining text so that no gaps are left in your file.) Text closed by the erase command is placed in the erase buffer.

To erase the line where the cursor is located, press <Cmd> **erase** <CR>. To erase a number of lines or a rectangular area, press <Mark>, move the cursor, and press <Cmd> **erase** <CR>. You can also erase lines by pressing <Cmd> **erase** *n* <CR>, where *n* is the number of lines to be erased.

To restore the last text you erased, press <Cmd> **-erase** <CR>. Copies of the erased text can be inserted as many times and in as many places as you wish.

Any subsequent use of the erase command will replace the text in the erase buffer. Note: Pressing <Cmd> <DelChar> also replaces the contents of the erase buffer with the newly deleted text.

All material removed from a file by pressing <Cmd> **erase** <CR> or <Cmd>

<DelChar> during your current editing session is placed in the changes file (#o).

<Cmd> **erase** <CR>	Erases text, leaves blank space
<Cmd> **erase** *n* <CR>	Erases specified number of lines
<Cmd> **-erase** <CR>	Restores erased text

EXIT, BYE: EXITING FROM A FILE

To exit from the editor, press <Cmd> **exit** <CR> or <Cmd> **bye** <CR>. The edited file will be saved.

You can also exit from the editor without saving the file with any of the following commands:

<Cmd> **exit nosave** <CR>
<Cmd> **exit quit** <CR>
<Cmd> **exit abort** <CR>
<Cmd> **exit dump** <CR>

These commands perform the following functions:

	Saves edited file	Updates state file (.es1)	Removes keystroke file (.ek1)	Removes change file (.ec1)
<Cmd> exit <CR>	yes	yes	yes	yes
<Cmd> exit abort <CR>	no	no	no	yes
<Cmd> exit dump <CR>	no	no	no	no
<Cmd> exit nosave <CR>	no	yes	yes	yes
<Cmd> exit quit <CR>	no	no	yes	yes

See Appendix B for descriptions of state, keystroke, and change files.

Normally, you will exit from files with either the exit, exit nosave, or exit abort commands. When your editing session contains no serious editing mistakes, exit the editor and save the file with <Cmd> **exit** <CR>.

If your editing session contains a serious mistake, ruining all or part of the file, you can negate the final portion of the editing session with exit abort:

1. Press <Cmd> **exit abort** <CR>.

2. Type **e** <CR> in response to the % prompt.

3. Choose option 2, when prompted with the crash message (see Appendix A).

4. Watch the replay of your editing session on the screen. At a point just before the mistake occurred, interrupt the replay by pressing <Int> if you have a newer Ann Arbor, Ambassador, or XL terminal, or <Ctrl+c> if you have an earlier-model Ann Arbor terminal. The replay occurs quickly, so some skill is required to interrupt at the desired point.

5. Exit from the file with <Cmd> **exit** <CR>.

Sometimes you may wish to cancel all changes made during an editing session. In that case, exit the file with <Cmd> **exit nosave** <CR>. This option deletes the entire editing session, but will save the original version of your file.

<Cmd> **exit** <CR>	Exits from the editor and saves file
<Cmd> **exit abort** <CR>	Exits from the editor without saving file
<Cmd> **exit dump** <CR>	Exits from the editor without saving file
<Cmd> **exit nosave** <CR>	Exits from the editor without saving file
<Cmd> **exit quit** <CR>	Exits from the editor without saving file

FEED: ISSUING UNIX COMMANDS FROM WITHIN THE EDITOR

You can use some UNIX commands while you are in the editor by pressing <Cmd> **feed** *UNIXcommand* <CR>.

For example, you can use the UNIX sort command to alphabetize lines in a marked area by pressing <Mark> <↓> <Cmd> **feed sort** <CR>. This command keeps the original version of the lines and inserts the alphabetized version at the cursor position. (By contrast, the run sort command deletes the original version of the lines and inserts the alphabetized version in its place.)

CAUTION: If you experience a crash and have used the feed command, a replay of the editing session can produce unexpected results. For example, the output from the UNIX sort command can vary if you have added or deleted lines in the list being alphabetized between the time you originally issued the feed sort command and the time you recovered from the crash.

<Cmd> **feed** *UNIXcommand* <CR>	Executes UNIXcommand and inserts output at cursor position

FILL: CONTROLLING THE RIGHT MARGIN

The fill command moves text so that each line is filled with characters up to, but not beyond, the right margin. The default line length is 75 characters.

To fill a paragraph, move the cursor to the first line of the paragraph and press <Cmd> **fill** <CR>. This command evens out the line length beginning with the line the cursor is on and ending at the first totally blank line thereafter.

CAUTION: If you use the fill command, be certain each paragraph is followed by a totally blank line. The fill command treats formatting macros (such as .pp and .ip o) as part of the text to be filled.

The line length used by the fill command is the default width of 75, the width specified by the last set right margin command (**set rmar** *n* <CR>), or the width specified in the last center, fill, or justify command. To specify a different width, press <Cmd> **fill width=***n* <CR>, where *n* is the number of characters per line.

If you have made a mistake when filling text, press <Cmd> **insert adjust** <CR>. This command will insert the original unfilled version of the text at the cursor position. You must then delete the mistakenly filled text. You can retrieve this unfilled text only while you are in the current editing session and only before you issue another center, fill, or justify command.

<Cmd> **fill** <CR>	Fills text
<Cmd> **fill width=***n* <CR>	Fills text to specified width
<Cmd> **insert adjust** <CR>	Restores unfilled text

GOTO: MOVING THE CURSOR TO A SPECIFIED LINE

You can move the cursor to a specific line number in a file by pressing either <Cmd> **goto** *n* <CR> or <Cmd> *n* <CR>, where *n* is the line number.

To go to the beginning of a file, press <Cmd> **goto** <CR>, <Cmd> **1** <CR>, or <Cmd> **goto beginning** <CR>. To go to the end of a file, press <Cmd> **goto end** <CR>.

To go to the beginning of a range that you have set, press <Cmd> **goto rbeginning** <CR>. To go to the end of a range that you have set, press <Cmd> **goto rend** <CR>.

<Cmd> *n* <CR>	Moves cursor to designated line
<Cmd> **goto** *n* <CR>	Moves cursor to designated line
<Cmd> **goto beginning** <CR>	Moves cursor to beginning of file
<Cmd> **goto** <CR>	Moves cursor to beginning of file
<Cmd> **goto end** <CR>	Moves cursor to end of file
<Cmd> **goto rbeginning** <CR>	Moves cursor to beginning of range
<Cmd> **goto rend** <CR>	Moves cursor to end of range

INSERT ADJUST: RETRIEVING CENTERED/FILLED/JUSTIFIED TEXT

You may recover the original version of lines that have been centered, filled, or justified by pressing <Cmd> **insert adjust** <CR>. These lines are inserted at the cursor position. They do not replace the centered, filled, or justified text. You can retrieve this text only while you are in the current editing session and only before you issue another center, fill, or justify command.

<Cmd> **insert adjust** <CR>	Restores centered, filled, or justified text to original state

JOIN, -JOIN: JOINING/SPLITTING LINES

The join command joins two lines. The line the cursor is on is lengthened by appending the following line of text to it. The join command deletes blank spaces or blank lines preceding the line to be joined. To issue this command, press <Cmd> **join** <CR>. The cursor can be placed anywhere on the line when using the join command.

The -join command splits a line at the cursor position. To issue this command, press <Cmd> **-join** <CR>.

<Cmd> **join** <CR>	Joins two lines of text
<Cmd> **-join** <CR>	Returns joined line to its original state

JUSTIFY: CONTROLLING THE RIGHT MARGIN

The justify command moves text and adds blank spaces where necessary so that each line ends exactly at the right margin. The default line length is 75 characters.

To justify a paragraph, move the cursor to the first line of the paragraph and press <Cmd> **justify** <CR>. This command justifies all lines beginning with the line the cursor is on and ending at the first totally blank line thereafter.

CAUTION: If you use the justify command, be certain each paragraph is followed by a totally blank line. The justify command treats formatting macros (such as .pp and .ip o) as part of the text to be justified.

The line length used by the justify command is the default width of 75, the width specified by the last set right margin command (<Cmd> **set rmar** *n* <CR>), or the width specified in the last center, fill, or justify command. To specify a different width, press <Cmd> **justify width=***n* <CR>, where *n* is the number of characters per line.

If you have made a mistake when justifying text, press <Cmd> **insert adjust** <CR>. This command will insert the original unjustified version of the text at the cursor position. You must then delete the mistakenly justified text. You can retrieve this unjustified text only while you are in the current editing session and only before you issue another center, fill, or justify command.

<Cmd> **justify** <CR>	Justifies text
<Cmd> **justify width=***n* <CR>	Justifies text to specified width
<Cmd> **insert adjust** <CR>	Restores unjustified text

NAME: RENAMING A FILE WITHIN THE EDITOR

To rename a file while you are in the editor, press <Cmd> **name** *newfilename* <CR>. The renaming takes place immediately.

<Cmd> **name** *newfilename* <CR>	Renames current file

OPEN: INSERTING BLANK LINES

The open command inserts blank lines or rectangular areas in a file. To insert a blank line where the cursor is located, press <Cmd> **open** <CR>. To insert a number of blank lines or a blank rectangular area, press <Mark>, move the cursor to define the area you wish to open, and press <Cmd> **open** <CR>.

You can also insert blank lines by pressing <Cmd> **open** *n* <CR>, where *n* is the number of blank lines to be inserted in the file.

<Cmd> **open** <CR>	Opens blank line
<Cmd> **open** *n* <CR>	Opens specified number of blank lines

PICK, -PICK: COPYING/INSERTING TEXT

The pick command copies characters, lines, or rectangular areas into your pick buffer, from which copies can be placed anywhere in the file. It does not change the material that is picked--your text will appear untouched.

To pick the line on which the cursor is located, press <Cmd> **pick** <CR>. To pick a number of lines or a rectangular area, press <Mark>, move the cursor, and press <Cmd> **pick** <CR>. You can also pick lines by pressing <Cmd> **pick** *n* <CR>, where *n* is the number of lines to be picked.

To place copies of the text stored in the pick buffer anywhere in your

file, move the cursor to the desired location and press <Cmd> **-pick** <CR>. Copies of the picked text can be inserted as many times and in as many places as you wish.

Any subsequent use of the pick command will replace the text in the pick buffer.

All material picked during the current editing session is placed in the changes file (#p).

<Cmd> **pick** <CR>	Places text in pick buffer
<Cmd> **pick** *n* <CR>	Places specified number of lines in pick buffer
<Cmd> **-pick** <CR>	Inserts contents of pick buffer

RANGE, -RANGE, ?RANGE: SPECIFYING LINES FOR SEARCH/REPLACE

The range command specifies a range of lines when using the search or replace commands. This is useful when issuing multiple commands on the same area of the file.

To issue this command, press <Mark>, move the cursor to the last line to be included in the range, and press <Cmd> **range** <CR>. You can also define a range by pressing <Cmd> **range** *n* <CR>, where *n* is the number of lines in the range following the cursor position. (You cannot set a range in a rectangularly marked area.)

When a range is specified, the search or replace commands can be issued only when the cursor is within that range. "=RANGE" at the bottom of the screen indicates that you have set a range and the cursor is within the range. "<RANGE" or ">RANGE" at the bottom of the screen indicates that the cursor is located either before or after the range.

To remove the range, press <Cmd> **-range** <CR>. To set the same range again, press <Cmd> **range** <CR>.

To go to the beginning or end of a range, press <Cmd> **goto rbeginning** <CR> or <Cmd> **goto rend** <CR>. To determine the number of lines in a range, press <Cmd> **?range** <CR>.

<Cmd> **range** <CR>	Specifies a range
<Cmd> **range** *n* <CR>	Specifies a range
<Cmd> **-range** <CR>	Removes a range
<Cmd> **?range** <CR>	Displays number of lines within range
<Cmd> **goto rbeginning** <CR>	Moves cursor to beginning of range
<Cmd> **goto rend** <CR>	Moves cursor to end of range

REDRAW: REMOVING EXTRANEOUS CHARACTERS FROM SCREEN

To redraw the screen and eliminate extraneous characters (such as system messages, noise characters, or garbage characters), press <Cmd> **redraw** <CR>. This is helpful when system messages appear in the middle of your file or when you want to determine if strange characters are actually part of your file.

<Cmd> **redraw** <CR>	Removes unwanted characters from screen

REGULAR EXPRESSION: USING SPECIAL CHARACTERS WITH SEARCH/REPLACE

The Regular Expression Mode enables you to use special characters when searching for or replacing characters or words in a file.[1]

When searching for or replacing characters in the Regular Expression Mode, the following characters have special meanings: periods, backslashes, left brackets, carets, dollar signs, and asterisks.

To enter the Regular Expression Mode, press <Cmd> **regexp** <CR>. When you are in the Regular Expression Mode, you will see "RE" displayed beneath the window.

To turn off the Regular Expression Mode, press <Cmd> **-regexp** <CR>.

---

[1]For more information on using regular expression, consult the *UNIX Programmer's Manual* by typing **man ed** <CR> in response to the % prompt.

<Cmd> **regexp** <CR>	Turns on Regular Expression Mode
<Cmd> **-regexp** <CR>	Turns off Regular Expression Mode

The following examples demonstrate how to use the Regular Expression Mode with the <+Sch> key and with the replace command. You can also use the Regular Expression Mode with the <-Sch> key and with the replace interactive, replace show, and -replace commands.

Period takes the place of any one character:

<Cmd> *J.nes* <+Sch>	Finds subsequent words where *J* and *nes* are separated by a single character. This locates occurrences of *Janes*, *Jones*, and *Junes*
<Cmd> **replace** */J.nes/Jones/* <CR>	Substitutes *Jones* for all subsequent words where *J* and *nes* are separated by a single character. This changes words like *Janes* and *Junes* to *Jones*

Brackets define a range of alternative characters:

<Cmd> *[Aa]ffect* <+Sch>	Finds subsequent occurrences of both *Affect* and *affect*
<Cmd> *[2-5]* <+Sch>	Finds subsequent occurrences of *2*, *3*, *4*, and *5*
<Cmd> *[C-E]* <+Sch>	Finds subsequent occurrences of *C*, *D*, and *E*
<Cmd> *[C-E,c-e]* <+Sch>	Finds subsequent occurrences of *C*, *D*, *E*, *c*, *d*, and *e*
<Cmd> **replace** */[C-E]/?/* <CR>	Substitutes any subsequent *C*, *D*, or *E* with *?*

Caret in conjunction with brackets excludes characters it precedes:

<Cmd> *[^C-E,c-e]* <+Sch>	Finds subsequent occurrences of all characters except *C, D, E, c, d,* and *e*
<Cmd> **replace** */[^C-E]/?/* <CR>	Substitutes all subsequent characters except *C, D,* and *E* with *?*

Caret not in conjunction with brackets indicates beginning of a line:

<Cmd> *^Dear* <+Sch>	Finds subsequent occurrences where *Dear* appears at the beginning of a line
<Cmd> **replace** */^Dear/Dearest/* <CR>	Substitutes *Dear* with *Dearest* in all subsequent places where *Dear* appears at the beginning of a line

Dollar sign indicates end of a line:

<Cmd> *1985$* <+Sch>	Finds subsequent occurrences where *1985* appears at the end of a line
<Cmd> **replace** */1985$/1986/* <CR>	Substitutes *1985* with *1986* in all subsequent places where *1985* appears at the end of a line

Asterisk used in conjunction with period or brackets takes the place of any number of characters:

<Cmd> *hyper.*ism* <+Sch>	Finds subsequent occurrences where *hyper* and *ism* are separated by any number of characters. This locates occurrences of such words as *hyperthyroidism* and *hyperparathyroidism*
<Cmd> *hyper[a-z]*ism* <+Sch>	Finds subsequent occurrences where *hyper* and *ism* are separated by any number of lowercase characters. This locates occurrences of such words as *hyperthyroidism* and *thyperparathyroidism*, but not *hyperTHYROIDism*
<Cmd> **replace** */hyper.*ism/hyopthryoidism/* <CR>	Substitutes subsequent words where *hyper* and *ism* are separated by any number of characters with *hypothryodism*

Backslash causes Regular Expression Mode to ignore the meaning of one of its special characters:

<Cmd> \. <+Sch>	Finds subsequent occurrences of all periods rather than of all single characters
<Cmd> **replace** /\./?/ <CR>	Substitutes all subsequent periods (rather than all single characters) with *?*

REPLACE, -REPLACE: REPLACING TEXT

The replace command substitutes characters or words for other characters or words.

To replace all occurrences of *text* with *newtext* following the cursor position, press <Cmd> **replace** */text/newtext/* <CR>. To replace all occurrences of *text* with *newtext* in the opposite direction, press <Cmd> **-replace** */text/newtext/* <CR>.

The text that is searched for and the text that replaces it can be separated in the command by a variety of characters including /, ', ", and +. Use these characters only if they are not part of the original or replacement text.

The replace show option allows you to see each replacement as it occurs on the screen. To use this option, press <Cmd> **replace show** */text/newtext/* <CR>. You can interrupt the replacements at any time by pressing <Int> if you have a newer Ann Arbor, Ambassador, or XL terminal, or <Ctrl+c> if you have an earlier-model Ann Arbor terminal. The replace is performed quickly, so some skill is required to interrupt it at the desired point.

The replace interactive option allows you to replace only selected occurrences of *text* with *newtext*. To use this option, press <Cmd> **replace interactive** */text/newtext/* <CR>. This moves the cursor to the first occurrence of *text*. Pressing <Repl> substitutes *newtext* for *text*. To find the next occurrence of *text*, press <+Sch>. If you do not wish to replace *text* with *newtext*, press <+Sch> rather than <Repl>.

The replace command will also work with a marked area of full lines; however, it will not work on a rectangularly marked area.

For additional ways to replace text, see the regular expression and range commands in this section.

<Cmd> **replace** */a/b/* <CR>	Replaces *a* with *b* from cursor to end of file
<Cmd> **-replace** */a/b/* <CR>	Replaces *a* with *b* from cursor to beginning of file
<Cmd> **replace show** */a/b/* <CR>	Replaces *a* with *b*, showing each time replace occurs
<Cmd> **replace interactive** */a/b/* <CR>	Replaces *a* with *b* only if <Repl> is pressed

RUN: ISSUING UNIX COMMANDS FROM WITHIN THE EDITOR

You can use some UNIX commands while you are in the editor by pressing <Cmd> **run** *UNIXcommand* <CR>.

For example, you can use the UNIX sort command to alphabetize lines in a marked area by pressing <Mark> <↓> <Cmd> **run sort** <CR>. This command deletes the original version of the lines and inserts the alphabetized version at the cursor position. (By contrast, the feed command keeps the original version of the lines and inserts the alphabetized version at the cursor position.)

CAUTION: If you experience a crash and have used the run command, a replay of the editing session can produce unexpected results. For example, the output from the UNIX sort command can vary if you have added or deleted lines in the list being alphabetized between the time you originally issued the run sort command and the time you recovered from the crash.

<Cmd> **run** *UNIXcommand* <CR>	Executes UNIXcommand and inserts output at cursor position

SAVE: SAVING CURRENT FILE WITHOUT EXITING FROM THE EDITOR

The save command is a convenient way of backing up a modified file before attempting some new, dramatic editing. To save a copy of the file being edited, press <Cmd> **save** *newfilename* <CR>. The *newfilename* cannot be the name of the file being edited. The save command takes effect immediately.

<Cmd> **save** *newfilename* <CR>	Makes copy of current file giving it new filename

SET: CHANGING DEFAULTS FOR EDITOR KEYS AND COMMANDS

The set option allows you to change defaults for a number of editor keys and editor commands. To change a default, press <Cmd>, type in the appropriate set command (shown below), and press <CR>. The following options are now available with the set command:

Line Keys (default is one-fourth screen)

<Cmd> **set +line** *n* <CR>	Sets <+Line> to move file *n* lines forward
<Cmd> **set -line** *n* <CR>	Sets <-Line> to move file *n* lines backward
<Cmd> **set line** *n* <CR>	Sets <+Line> and <-Line> to move your file *n* lines

Page Keys (default is 1 screen)

<Cmd> **set +page** *n* <CR>	Sets <+Page> to move file *n* screens forward
<Cmd> **set -page** *n* <CR>	Sets <-Page> to move file *n* screens backward
<Cmd> **set page** *n* <CR>	Sets <+Page> and <-Page> to move your file *n* screens

Window Keys (default is 16 columns)

<Cmd> **set left** *n* <CR>	Sets <Left> and <Ctrl+a> to move window *n* columns to the left
<Cmd> **set right** *n* <CR>	Sets <Right> and <Ctrl+s> to move window *n* columns to the right
<Cmd> **set window** *n* <CR>	Sets <Left>, <Ctrl+a>, <Right>, and <Ctrl+s> to move window *n* columns

Word Keys (default is whitespace)

<Cmd> **set worddelim alphanum** <CR>	Sets <+Word> and <-Word> to move cursor to first character of following or preceding word or to first character following or preceding a nonalphanumeric character (e.g., #)
<Cmd> **set worddelim whitespace** <CR>	Sets <+Word> and <-Word> to move cursor to first character of following or preceding word

Line Width (default is 75 columns)

<Cmd> **set width** *n* <CR>	Sets line width to *n* columns

Margins (default: left margin column 1; right margin column 75; stick)

<Cmd> **set lmar** *n* <CR>	Sets left margin at the *n*th column
<Cmd> **set rmar** *n* <CR>	Sets right margin at the *n*th column
<Cmd> **set stick** <CR>	When word wrap is turned off, causes cursor to stop at right margin
<Cmd> **set nostick** <CR>	When word wrap is turned off, causes cursor not to stick at right margin

Determining Current Set Options

<Cmd> **set ?** <CR>	Shows values of set options at bottom of screen

SPLIT, -SPLIT: SPLITTING/RESTORING LINES

The split command divides one line into two at the cursor position. The right-hand side of the line, including the character at the cursor position, is inserted as a new line on the line below, and the window is redrawn. To issue this command, press <Cmd> **split** <CR>.

The -split command restores the line to its original version. To issue this command, press <Cmd> **-split** <CR>.

<Cmd> **split** <CR>	Splits line into two lines
<Cmd> **-split** <CR>	Returns split line to its original state

STOP: EXITING QUICKLY FROM A FILE

The stop command exits quickly from a file, returning you to the % prompt. This does not save your file, which requires extra time. To issue this command, press <Cmd> **stop** <CR>.

You may not edit another file at this time. To resume editing this file, type **fg** <CR> in response to the % prompt. This returns you to your file.

If you have issued the stop command, you must bring your file into the "foreground" before logging out of the Text Processor. Otherwise, you will receive a message saying "There are stopped jobs" and you will not be allowed to logout.

<Cmd> **stop** <CR>	Exits quickly from file
% **fg** <CR>	Returns you to file exited with stop command

TAB, -TAB: SETTING/REMOVING TABS

To set tab stops in addition to those already displayed at the top of the screen (every 8 columns), press <Cmd> **tab** *n* <CR>, where *n* is one or more column numbers separated by blanks. If *n* is not specified, the tab is set where the cursor is positioned.

To clear a tab stop at the cursor position, press <Cmd> **-tab** <CR>.

<Cmd> **tab** *n* <CR>	Sets tab(s) in designated column(s)
<Cmd> **-tab** <CR>	Removes tab at cursor position

TABFILE: CREATING A SEPARATE FILE WITH TAB STOPS

Tabfile sets tabs according to a file containing a list of column numbers separated by blanks. To set tabs using this command, press <Cmd> **tabfile** *filename* <CR>, where *filename* is the name of a file containing the column numbers of the tab stops. The file with the tabs must be created and saved before editing the text file. Note: This command does not erase any existing tabs; it merely inserts additional tabs.

<Cmd> **tabfile** *filename* <CR>	Sets tabs from a file containing column numbers

TABS, -TABS: SETTING/REMOVING MULTIPLE TABS

To set multiple tabs in addition to those already set (every 8 columns), press <Cmd> **tabs** *n* <CR>, where *n* is every nth column beginning with column 1. For example, pressing <Cmd> **tabs** *5* <CR> sets tabs in columns 5, 10, 15, 20, etc.

To remove all tabs press <Cmd> **-tabs** <CR>. This command can also be used to clear tabs in every nth column by pressing <Cmd> **-tabs** *n* <CR>, where *n* is every nth column beginning with column 1.

<Cmd> **tabs** *n* <CR>	Sets tabs every *n*th column
<Cmd> **-tabs** <CR>	Removes all tabs
<Cmd> **-tabs** *n* <CR>	Removes tabs in every *n*th column

TRACK, -TRACK: TRACKING CURRENT AND ALTERNATE FILES

The track command causes the current file and the alternate file to track each other. Thus, if you move the current file's window two screens forward, the alternate file's window also moves two screens forward. This command is useful when comparing the contents of two files.

To issue this command, press <Cmd> **track** <CR>. "TRACK" at the bottom of the screen indicates the track command is in effect.

To turn off tracking, press <Cmd> **-track** <CR>.

<Cmd> **track** <CR>	Tracks current and alternate files
<Cmd> **-track** <CR>	Removes tracking with alternate file

-UPDATE, UPDATE: EXITING FROM A FILE WITHOUT SAVING CHANGES

If you are editing a file and do not want to save the changes you have made, press <Cmd> **-update** <CR>. This command causes the editor to ignore all changes made during the current editing session, both before and after you issue the -update command. This command takes effect when you exit from the file.

The -update command is especially useful when you are editing several files as alternate files and do not wish to save the changes you have made in one or more (but not all) of them. Bring onto the screen each of the files with changes you do not want to save and then press <Cmd> **-update** <CR>.

If you then change your mind while still in the current editing session, you can override the -update command by pressing <Cmd> **update** <CR>; the changes made throughout the editing session will be saved.

<Cmd> **-update** <CR>	Does not save changes during current editing session
<Cmd> **update** <CR>	Cancels -update command

WINDOW, -WINDOW: OPENING/REMOVING AN ALTERNATE WINDOW

To create a window, position the cursor either on the left or the top edge of the screen, but not in the top left-hand corner. Press <Cmd> **window** *newfilename* <CR>. A second window will be drawn on the screen, and *newfilename* will become the current file.

Windows are numbered in the order in which they were created. To move the editor from window to window, press <ChgWin> or <Ctrl+z>. To move to a particular window, press <Cmd> *n* <ChgWin> or <Cmd> *n* <Ctrl+z>, where *n* is the number of the window.

You can create a new window with the current file in it by pressing <Cmd> **window** <CR>.

To remove the last window created, press <Cmd> **-window** <CR>.

<Cmd> **window** <CR>	Creates window with current file in it
<Cmd> **window** *filename* <CR>	Creates window with named file in it
<Cmd> **-window** <CR>	Removes last window created

WP, -WP: TURNING ON/OFF WORD WRAP

The Word Wrap Mode causes the cursor to return automatically to the following line when it reaches the right margin. The default right margin is 75. It can be changed by a **set rmar** *n* command or by a **width=***n* option added to an editor command.

Word wrap is the default mode for the editor. When you are in the Word Wrap Mode, you will see "WP" displayed beneath the window. You can turn off the Word Wrap Mode by pressing <Cmd> **-wp** <CR>. To turn it back on, press <Cmd> **wp** <CR>.

If you mark an area and then turn on word wrap, the boundaries of the marked area become your left and right margin settings.

<Cmd> **wp** <CR>	Turns on Word Wrap Mode
<Cmd> **-wp** <CR>	Turns off Word Wrap Mode

SUMMARY OF EDITOR COMMANDS

Command	What the Command Does
blot box	Deletes box around marked area
box	Places box around marked area
caps	Changes all letters to uppercase in a rectangularly marked area
ccase	Changes all uppercase letters to lowercase and all lowercase letters to uppercase in a rectangularly marked area
center	Centers line
close	Removes line
-close	Restores closed text
command	Places editor in constant Command Mode
-command	Takes editor out of constant Command Mode
cover close	Uses contents of close buffer to cover portion of file
cover erase	Uses contents of erase buffer to cover portion of file
cover pick	Uses contents of pick buffer to cover portion of file
delete	Deletes current file
dword	Deletes word at cursor position
-dword	Restores deleted word
e filename	Brings up an alternate file
erase	Erases text; leaves blank space
-erase	Restores erased text

Command	What the Command Does
exit	Exits from the editor and saves file
exit abort	Exits from the editor without saving file
exit dump	Exits from the editor without saving file
exit nosave	Exits from the editor without saving file
exit quit	Exits from the editor without saving file
feed UNIXcommand	Executes UNIXcommand and inserts output at cursor position
fill	Fills text
goto beginning	Moves cursor to beginning of file
goto end	Moves cursor to end of file
goto n	Moves cursor to designated line
goto rbeginning	Moves cursor to beginning of range
goto rend	Moves cursor to end of range
insert adjust	Restores centered, filled, or justified text to original state
join	Joins two lines of text
-join	Returns joined line to its original state
justify	Justifies text
n	Moves cursor to designated line
name filename	Renames current file
open	Inserts blank line
pick	Places text in pick buffer
-pick	Inserts contents of pick buffer into file

Command	What the Command Does
range	Specifies a range
-range	Removes a range
?range	Displays number of lines within range
redraw	Removes unwanted characters from screen
regexp	Turns on Regular Expression Mode
-regexp	Turns off Regular Expression Mode
replace /a/b/	Replaces a with b from cursor position to end of file
-replace /a/b/	Replaces a with b from cursor position to beginning of file
replace interactive /a/b/	Replaces a with b when <Repl> is pressed
replace show /a/b/	Replaces a with b showing each time replace occurs
run UNIXcommand	Executes UNIXcommand and inserts output at cursor position
save newfilename	Makes copy of current file, giving it new filename
set left n	Sets <Left> and <Ctrl+a> to move window n columns to the left
set line n	Sets <+Line> and <-Line> to move file n lines
set +line n	Sets <+Line> to move file n lines forward
set -line n	Sets <-Line> to move file n lines backward
set lmar n	Sets left margin at the nth column
set page n	Sets <+Page> and <-Page> to move file n screens
set +page n	Sets <+Page> to move file n screens forward

Command	What the Command Does
set -page n	Sets <-Page> to move file n screens backward
set right n	Sets <Right> and <Ctrl+s> to move window n columns to the right
set rmar n	Sets right margin at the nth column
set nostick	When word wrap is turned off, causes cursor not to stick at right margin
set stick	When word wrap is turned off, causes cursor to stick at right margin
set width n	Sets line width to n columns
set window n	Sets <Left>, <Ctrl+a>, <Right>, and <Ctrl+s> to move window n columns
set worddelim alphanum	Sets <+Word> and <-Word> to move cursor to first character of following or preceding word or to first character following or preceding a nonalphanumeric character (e.g., #)
set worddelim whitespace	Sets <+Word> and <-Word> to move cursor to first character of following or preceding word
set ?	Shows values of set options at bottom of screen
split	Splits line into two lines
-split	Returns split line to its original state
stop	Exits quickly from file, without saving file
tab n	Sets tab(s) in designated column(s)
-tab	Removes tab at cursor position
tabfile filename	Sets tabs from a file containing column numbers
tabs n	Sets tabs in every nth column
-tabs	Removes all tabs
-tabs n	Removes tabs in every nth column

Command	What the Command Does
track	Tracks current and alternate files
-track	Removes tracking with alternate file
update	Cancels -update command
-update	Does not save changes during current editing session
window	Creates window with current file in it
window file	Creates window with named file in it
-window	Removes last window created
wp	Turns on Word Wrap Mode
-wp	Turns off Word Wrap Mode

4. EDITOR FUNCTION KEYS

This section describes the keys on your terminal that are used to perform special functions when you are using the editor.

Different terminals may label the same function key in different ways. For example, the key used to issue a command to the editor is labeled <Arg>, <Brk>, <Brk/Cmd>, or <Cmd> depending on what kind of terminal you use.

The three most common terminals currently in use at Rand are Ann Arbor, Ann Arbor Ambassador, and Ann Arbor XL terminals. (The label on the front of the terminal will tell you what kind you are using.) Newer Ann Arbor, Ambassador, and XL terminals have keys (e.g., <Int>) for functions that require a keystroke sequence (e.g., <Ctrl+c>) on earlier-model Ann Arbor terminals. Ambassador and XL terminals also have keys (e.g., <Erase>) for functions that require an editor command (e.g., <Cmd> **erase** <CR>) on other Ann Arbor terminals.

<ALT>: MOVING TO AN ALTERNATE FILE

Pressing <Alt> moves the cursor to an alternate file. If you have no alternate file, the editor will display the following error message: "*** No alternate file". (Press <Ctrl+b> on earlier-model Ann Arbor terminals.)

<Alt>	Moves cursor to alternate file

<ARG>: ISSUING A COMMAND

See <Cmd>.

<ARROW KEYS>: MOVING THE CURSOR

Your terminal has four arrow keys: <→>, <←>, <↑>, and <↓>. These keys move the cursor without erasing any text (unlike <Bsp> and <Spacebar>).

When the cursor is at the right side of the window and you press <→>, the window moves so that the cursor remains visible. This also occurs when the cursor is at the top, bottom, or left edge of the window. However, the cursor (and the window) cannot go to the left of column 1, nor can it go higher than line 1.

You can use the command key in conjunction with the arrow keys.

<→>	Moves cursor to right
<←>	Moves cursor to left
<↑>	Moves cursor up
<↓>	Moves cursor down
<Cmd> <→>	Moves cursor to end of text on line; if sequence is repeated, cursor moves to right edge of window
<Cmd> <←>	Moves cursor to left edge of window
<Cmd> <↑>	Moves cursor to top edge of window
<Cmd> <↓>	Moves cursor to bottom line of text in window

<BRK>, <BRK/CMD>: ISSUING A COMMAND

See <Cmd>.

<BSP>: BACKSPACING THE CURSOR

To backspace the cursor and erase a character, press <Bsp>. To backspace the cursor without erasing a character, press <←>.

The backspace key has different functions, depending on whether the Insert Mode is off or on. The <Bsp> key is sometimes labeled <BS> or <Back Space>.

<Bsp>	Moves cursor left, deleting characters
<Cmd> <Bsp>	Leaves cursor in place; deletes text to left of cursor
<Ins> <Bsp>	Moves cursor to the left one column and deletes any character there. The text to the right of the cursor is moved one character to the left (following the cursor
<Ins> <Cmd> <Bsp>	Deletes all characters between the cursor and the left margin. The cursor is moved to the left margin and text to the right of it shifts left

<CHG WIN>: WORKING IN A DIFFERENT WINDOW

Pressing <ChgWin> moves the cursor to the next window. If there are only two windows, this alternates windows. If there are more than two windows, this moves the editor from window to window in the order in which the windows were created. To move to a specific window, press <Cmd> *n* <ChgWin>, where *n* is the window number. (Press <Ctrl+z> on earlier-model Ann Arbor terminals.)

<ChgWin>	Moves cursor from one windowed file to another
<Cmd> *n* <ChgWin>	Moves to specified window

<CLOSE>: REMOVING/RESTORING TEXT

Pressing <Close> removes characters, lines, or rectangular areas and moves the remaining text so that no gaps are left in your file. (By contrast, pressing <Erase> removes characters, lines, or rectangular areas and leaves a blank space where they were.) Text removed by pressing <Close> is placed in the close buffer.

To delete the line where the cursor is located, press <Close>. To delete a number of lines or a rectangular area, press <Mark>, move the cursor, and press <Close>. You can also delete lines by pressing <Cmd> *n* <Close>, where *n* is the number of lines to be closed.

To restore the last text you closed, press <Cmd> <Close>. Copies of the closed text can be inserted as many times and in as many places as you wish.

Any subsequent close will replace the text in the close buffer. Note: Pressing <Cmd> <Bsp> while in Insert Mode also replaces the contents of the close buffer with the newly deleted text.

All material removed from a file by pressing <Close> or <Cmd> <Bsp> during your current editing session is placed in the changes file (#o).

<Close>	Closes cursor line
<Cmd> *n* <Close>	Closes specified number of lines
<Cmd> <Close>	Restores closed line

<CMD>: ISSUING A COMMAND

Pressing <Cmd> tells the editor that you want to issue a command (see Section 3 for commands that may be issued). Pressing <CR> tells the editor to perform the command you have typed.

To clear an unwanted command from the command line, backspace over it and press <CR>.

The last command executed can be returned to the command line by pressing <Cmd> <Cmd> <Alt>. This line can be edited and then executed again by pressing <CR>.

The <Cmd> key is sometimes labeled <Arg>, <Brk>, or <Brk/Cmd>.

<Cmd>	Allows you to issue a command
<Cmd> <Cmd> <Alt>	Repeats last command on command line

<CNTL CHAR>: INSERTING A CONTROL CHARACTER

Pressing <CntlChar> produces a small, bright block (▮) on the screen. The ▮ tells the editor that the next character typed is to be treated as a control character to produce special codes, such as for page breaks and underlining. (Press <Ctrl+\> on Ann Arbor terminals.)

<CntlChar>	Makes block symbol on Ambassador and XL terminals

<CR>: MOVING CURSOR AND PERFORMING COMMANDS

In the Edit Mode, pressing <CR> moves the cursor to left margin of the next line. If the cursor is at the bottom of the window, pressing <CR> also advances the window one-fourth of a screen.

In the Command Mode, pressing <CR> tells the editor to perform the command you have typed.

<CR>	Moves cursor to left margin of next line

<CTRL>: USING CONTROL FUNCTIONS

Holding down the <Ctrl> key shifts the keyboard into a third case, in which many keys perform special editor functions. Pressing <Ctrl> by itself does not perform a function; you must press another key in conjunction with the <Ctrl> key. The most commonly used keystrokes are described below.

Earlier-model Ann Arbor terminals, which require the use of the <Ctrl> key, have the function written on the key. For example, the <a> key has LEFT printed in small blue letters on its front side. On the newer Ann Arbor, Ambassador, and XL terminals, many of these editor features have been replaced by function keys (the following control key functions will not work on Ann Arbor Ambassador or XL terminals).

<Ctrl+a>. Pressing <Ctrl+a> moves the window 16 columns to the left. (The window cannot move to the left of column 1.) Pressing <Cmd> <Ctrl+a> moves the window to the left so that it displays columns 1-78 of the file. (Press <Left> on newer Ann Arbor, Ambassador, and XL terminals.)

To change the number of columns moved by pressing <Ctrl+a>, see the set left and set window commands described in Section 3.

<Ctrl+b>. Pressing <Ctrl+b> moves the cursor to an alternate file. If you have no alternate file, the editor will display the following error message: "*** No alternate file". (Press <Alt> on newer Ann Arbor, Ambassador, and XL terminals.)

<Ctrl+c>. Pressing <Ctrl+c> interrupts a command. If there is no command to interrupt or if you do not press <Ctrl+c> fast enough to interrupt the command, the editor will display the following error message: "*** No operation to interrupt". (Press <Int> on newer Ann Arbor, Ambassador, and XL terminals.)

<Ctrl+s>. Pressing <Ctrl+s> moves the window 16 columns to the right. Pressing <Cmd> <Ctrl+s> moves the window to the right so

that the cursor is located in column 1 of the moved window. (Press <Right> on newer Ann Arbor, Ambassador, and XL terminals.)

To change the number of columns moved by pressing <Ctrl+s>, see the set right and set window commands described in Section 3.

<Ctrl+z>. Pressing <Ctrl+z> moves the cursor to the next window. If there are only two windows, this alternates windows. If there are more than two windows, this moves the editor from window to window in the order in which the windows were created. To move to a specific window, press <Cmd> *n* <Ctrl+z>, where *n* is the window number. (Press <ChgWin> on newer Ann Arbor, Ambassador, and XL terminals.)

<Ctrl+[>. Pressing <Ctrl+[> sets and removes tabs. To set a tab, move the cursor to the column where a tab is desired and press <Ctrl+[>. To remove a tab, move the cursor to the column where the tab is to be removed and press <Cmd> <Ctrl+[>. (Press <S/R Tab> on newer Ann Arbor, Ambassador, and XL terminals.)

<Ctrl+\>. Pressing <Ctrl+\> produces a small, bright block (▮) on the screen. The ▮ tells the editor that the next character typed is to be treated as a control character to produce special codes, such as for page breaks and underlining. (Press <CntlChar> on Ambassador and XL terminals.)

<Ctrl+a>	Moves screen 16 characters to left
<Ctrl+b>	Moves cursor to alternate file
<Ctrl+c>	Interrupts editor command
<Ctrl+s>	Moves screen 16 characters to right
<Ctrl+z>	Moves cursor from one windowed file to another
<Ctrl+[>	Sets tab at cursor position
<Cmd><Ctrl+[>	Removes tab at cursor position
<Ctrl+\>	Makes a block symbol on Ann Arbor terminals

<DEL CHAR>: DELETING CHARACTERS

Pressing <DelChar> deletes the character the cursor is on and moves all characters that are to the right of the cursor one column to the left. Characters deleted in this manner cannot be recovered.

To delete the character at the cursor and all characters to the right of the cursor on the same line, press <Cmd> <DelChar>. Characters deleted in this manner enter the erase buffer and can be restored by pressing <Cmd> <Erase> or <Cmd> **-erase** <CR>.

<DelChar>	Deletes character
<Cmd> <DelChar>	Deletes all characters to right of cursor
<Cmd> <Erase>	Restores erased text

<DEL WORD>: DELETING/RESTORING A WORD

Pressing <DelWord> on Ambassador or XL terminals deletes a single word or a series of characters up to a blank space and fills in the space left by the deletion.

To delete a word or series of characters, place the cursor on any letter or on the space immediately preceding the characters and press <DelWord>. To restore the word or characters deleted in this fashion, press <Cmd> <DelWord>.

<DelWord>	Deletes word
<Cmd> <DelWord>	Restores deleted word

<ERASE>: ERASING/RESTORING TEXT

Pressing <Erase> on Ambassador or XL terminals removes characters, lines, or rectangular areas and leaves a blank space where they were. (By contrast, pressing <Close> removes characters, lines, or rectangular areas and moves the remaining text so that no gaps are left in your file.) Text closed by pressing <Erase> is placed in the erase buffer.

To erase the line on which the cursor is located, press <Erase>. To erase a number of lines or a rectangular area, press <Mark>, move the cursor, and press <Erase>. You can also erase lines by pressing <Cmd> *n* <Erase>, where *n* is the number of lines to be erased.

To restore the last text you erased, press <Cmd> <Erase>. Copies of the erased text can be inserted as many times and in as many places as you wish.

Any subsequent pressing of <Erase> will replace the text in the erase buffer. Note: Pressing <Cmd> <DelChar> also replaces the contents of the erase buffer with the newly deleted text.

All material removed from a file by pressing <Erase> or <Cmd> <DelChar> during your current editing session is placed in the changes file (#o).

<Erase>	Erases text; leaves blank space
<Cmd> *n* <Erase>	Erases specified number of lines
<Cmd> <Erase>	Restores erased text

<HOME>: MOVING CURSOR TO UPPER LEFT CORNER OF WINDOW

Pressing <Home> moves the cursor to the upper left corner of the current window. Pressing <Cmd> <Home> moves the cursor to the lower left corner.

<Home>	Moves cursor to top left corner of window
<Cmd> <Home>	Moves cursor to bottom left corner of window

<INS>: INSERTING TEXT WHILE TYPING

Pressing <Ins> puts the editor into the Insert Mode. Pressing it again turns off the Insert Mode. When the editor is in Insert Mode, the word "INSERT" appears below the window.

Text typed while the Insert Mode is turned off replaces existing text. Text typed while the Insert Mode is turned on is inserted into the line, moving existing text to the right. In addition, backspace functions behave differently while Insert Mode is turned on.

<Ins>	Moves into and out of Insert Mode

<INT>: INTERRUPTING A COMMAND

Pressing <Int> interrupts an editor command. If there is no command to interrupt or if you do not press <Int> fast enough to interrupt the command, the editor will display the error message: "*** No operation to interrupt". (Press <Ctrl+c> on earlier-model Ann Arbor terminals.)

<Int>	Interrupts editor command

<JOIN>: JOINING LINES

Pressing <Join> on XL terminals joins two lines. The line the cursor is on is lengthened by appending the following line of text to it. This deletes blank spaces or blank lines preceding the line to be joined.

There is no comparable key on other terminal models.

<Join>	Joins two lines of text

<LEFT>: MOVING WINDOW TO THE LEFT

Pressing <Left> moves the window 16 characters to the left. (The window cannot move to the left of column 1.) Pressing <Cmd> <Left> moves the window to the left so that it displays columns 1-78 of the file. (Press <Ctrl+a> on earlier-model Ann Arbor terminals.)

To change the number of columns moved by pressing <Left>, see the set left and set window commands described in Section 3.

<Left>	Moves screen 16 characters to left
<Cmd> <Left>	Moves screen back to original position

<+LINE>: MOVING WINDOW FORWARD

Pressing <+Line> moves the top fourth of what is in the window out of sight and redraws the screen to bring into sight the same number of lines at the bottom. The cursor stays at the same line number, or if that line is moved off the screen, the cursor goes to the top of the window.

Pressing <Cmd> <+Line> moves the screen so that the line the cursor is on is positioned at the top of the window.

Pressing <Cmd> *n* <+Line> moves the screen *n* lines forward. (To change the number of lines moved by pressing <+Line>, see the set line and the set +line commands described in Section 3.)

<+Line>	Moves top fourth of screen out of sight; brings equal number of lines into sight at bottom of screen
<Cmd> <+Line>	Moves screen so cursor line is at top of screen
<Cmd> *n* <+Line>	Moves screen n lines forward

<-LINE>: MOVING WINDOW BACKWARD

Pressing <-Line> moves the bottom fourth of what is in the window out of sight and redraws the screen to bring into sight the same number of lines at the top. The cursor stays at the same line number, or if that line is moved off the screen, the cursor goes to the bottom of the window. Pressing <-Line> has no effect if the top of the window is at line 1.

Pressing <Cmd> <-Line> moves the screen so that the line the cursor is on is positioned at the bottom of the window.

Pressing <Cmd> *n* <-Line> moves the screen *n* lines backward. (To change the number of lines moved by pressing <-Line>, see the set line and the set -line commands described in Section 3.)

<-Line>	Moves bottom fourth of screen out of sight; brings equal number of lines into sight at top of screen
<Cmd> <-Line>	Moves screen so cursor line is at bottom of screen
<Cmd> *n* <-Line>	Moves screen n lines backward

<LINE FEED>: MOVING THE CURSOR

Pressing <LineFeed> moves the cursor down one line in the same column without erasing any text. Pressing <Cmd> <LineFeed> moves the cursor to the last line of text in the window, in the same column; if this sequence is repeated, the cursor moves to the bottom edge of the window.

<LineFeed>	Moves cursor down one line
<Cmd> <LineFeed>	Moves cursor to bottom of window

<MARK>: DEFINING A BLOCK OF TEXT

Pressing <Mark> allows you to indicate a block of text to be used in conjunction with various editor commands and key functions.

To mark several whole lines, move the cursor to the first line and press <Mark>. Then move the cursor to the last line by pressing <↓>, <+Line>, or <+Page>. (The number of lines marked off is indicated in the message below the window, e.g., "MARK 4".)

To mark a rectangular area:

1. Move the cursor to the upper left corner of the rectangle and press <Mark>.

2. Move the cursor to the right one column beyond the text to be marked by pressing <→>; this defines the width of the rectangle.

3. Finally, move the cursor to the bottom by pressing <↓>, <+Line>, or <+Page>; this defines the length of the rectangle. (The number of lines and columns marked off is indicated in the message below the window, e.g., "MARK 2x4".)

To mark a portion of a line:

1. Move the cursor to the first character of the text to be marked and press <Mark>.

2. Move the cursor to the right one column beyond the text to be marked by pressing <→>. (The number of characters marked off is indicated in the message below the window, e.g., "MARK 1x12".)

After marking an area in this fashion, you can use a key function or editor command to change the marked area. For example, you can press <Pick> or <Cmd> **pick** <CR> to place a copy of the marked area in the pick buffer, or <Close> or <Cmd> **close** <CR> to remove the area from the file and place it in the close buffer, or <Open> or <Cmd> **open** <CR> to open the area.

Some commands, such as center, fill, and justify, can be used only with marked whole lines. Other commands, such as caps, ccase, and cover, can be used only with marked rectangular areas.

The mark function is normally turned off when you press a function key or issue an editor command. You can also turn off an unwanted mark function by pressing <Cmd> <Mark>.

<Mark>	Defines a marked area
<Cmd> <Mark>	Turns off Mark Mode

<OPEN>: OPENING BLANK LINES

Pressing <Open> inserts blank lines or rectangular areas in a file. To insert a blank line where the cursor is located, press <Open>. To insert a number of blank lines or a blank rectangular area, press <Mark>, move the cursor to define the area you wish to open, and press <Open>.

You can also insert blank lines by pressing <Cmd> *n* <Open>, where *n* is the number of blank lines to be inserted in the file.

Note that the dimensions of the opened area are not saved by the editor. Thus a subsequent press of the <Cmd> <Open> key does not return the area to its former state.

<Open>	Opens blank line
<Cmd> *n* <Open>	Opens specified number of blank lines

<+PAGE>: MOVING WINDOW FORWARD ONE SCREEN

Pressing <+Page> moves the window toward the end of the file one screen at a time. You can move the window more than one screen by pressing <Cmd> *n* <+Page>, where *n* is the number of screens to be moved toward the end of the file. (To change the number of screens moved by pressing <+Page>, see the set page and the set +page commands described in Section 3.)

Pressing <Cmd> <+Page> moves the cursor to the last line of the file.

<+Page>	Moves cursor forward one window
<Cmd> *n* <+Page>	Moves cursor forward specified number of windows
<Cmd> <+Page>	Moves cursor to end of file

<-PAGE>: MOVING WINDOW BACKWARD ONE SCREEN

Pressing <-Page> moves the window toward the beginning of the file one screen at a time. You can move the window more than one screen by pressing <Cmd> *n* <-Page>, where *n* is the number of screens to be moved toward the beginning of the file. The window never moves beyond line 1. (To change the number of screens moved by pressing <-Page>, see the set page and the set -page commands described in Section 3.)

Pressing <Cmd> <-Page> moves the cursor to line 1 of the file.

<-Page>	Moves cursor backward one window
<Cmd> *n* <-Page>	Moves cursor backward specified number of windows
<Cmd> <-Page>	Moves cursor to beginning of file

<PICK>: COPYING/INSERTING TEXT

Pressing <Pick> copies characters, lines, or marked rectangular areas into your pick buffer, from which copies can be placed anywhere in the file. It does not change the material that is picked; your text will appear untouched.

To pick the line on which the cursor is located, press <Pick>. To pick a number of lines or a rectangular area, press <Mark>, move the cursor, and press <Pick>. You can also pick lines by pressing <Cmd> *n* <Pick>, where *n* is the number of lines to be picked.

To place copies of the text stored in the pick buffer anywhere in your file, move the cursor to the desired location and press <Cmd> <Pick>. Copies of the picked text can be inserted as many times and in as many places as you wish.

Any subsequent pick will replace the text in the pick buffer.

All material picked during the current editing session is placed in the changes file (#p).

<Pick>	Places text in pick buffer
<Cmd> *n* <Pick>	Places specified number of lines in pick buffer
<Cmd> <Pick>	Inserts contents of pick buffer

<REPL>: REPLACING WORD OR SERIES OF WORDS

Pressing <Repl> in conjunction with the replace interactive command (described in Section 3) allows you selectively to replace text with different text.

To use this option, press <Cmd> **replace interactive** */text/newtext/* <CR>. This moves the cursor to the first occurrence of *text*. Pressing <Repl> substitutes *newtext* for *text*. To find the next occurrence of *text*, press <+Sch>. If you do not wish to replace *text* with *newtext*, press <+Sch> rather than <Repl>.

<Repl>	Replaces characters after issuing interactive replace command

<RIGHT>: MOVING WINDOW TO THE RIGHT

Pressing <Right> moves the window 16 columns to the right. Pressing <Cmd> <Right> moves the window to the right so that the cursor is located in column 1 of the moved window. (Press <Ctrl+s> on earlier-model Ann Arbor terminals.)

To change the number of columns moved by pressing <Right>, see the set right and set window commands described in Section 3.

<Right>	Moves screen 16 characters to right
<Cmd> <Right>	Moves screen so that column on which cursor is positioned is on left edge of window

<+SCH>: SEARCHING FORWARD FOR TEXT

Pressing <+Sch> enables you to search toward the end of a file for words or characters. To search for specific words or characters, press <Cmd> *word* <+Sch> or <Cmd> *phrase*<+Sch>. All characters of the word or phrase must appear on the same line in the file. If the editor finds an occurrence of *word* or *phrase*, it moves the cursor to the first letter of *word* or *phrase*. To find additional occurrences, continue pressing <+Sch> until "***Search key not found" appears beneath the window.

To search for text that appears only at the left margin of the window, press <Cmd> ▌J*word* <+Sch>.

If the text being searched for is only one word, you can place it in the search buffer by positioning the cursor on its first character and pressing <Cmd> <+Sch>. The editor will search for the next occurrence of that word. Subsequent occurrences can be searched for by pressing <+Sch>.

Text searched for remains in the search buffer until you place other text in the buffer.

For additional capabilities of <+Sch>, see the regular expression command described in Section 3.

<+Sch>	Searches from cursor position forward
<Cmd> <+Sch>	Places word at cursor position into search buffer and searches for it from cursor position forward

<-SCH>: SEARCHING BACKWARD FOR TEXT

Pressing <-Sch> enables you to search backward toward the beginning of a file for words or characters. In all other respects, it functions the same as the <+Sch> key.

For additional capabilities of <-Sch>, see the regular expression command described in Section 3.

<-Sch>	Searches from cursor position backward
<Cmd> <-Sch>	Places word at cursor position into search buffer and searches for it from cursor position backward

<SPACE>: MOVING CURSOR TO THE RIGHT

In the Edit Mode, pressing <Spacebar> moves the cursor one column to the right and erases existing text. In the Insert Mode, pressing <Spacebar> inserts blank spaces and moves existing characters to the right.

<Spacebar>	Moves cursor one column to right

<SPLIT>: SPLITTING LINES

Pressing <Split> on XL terminals splits one line into two lines at the cursor position. The right-hand side of the line, including the character at the cursor position, is inserted as a new line on the line below, and the window is redrawn.

Thre is no comparable key on other terminal models.

<Split>	Splits line into two lines

<S/R TAB>: SETTING/REMOVING TABS

Pressing <S/R Tab> sets and removes tabs. To set a tab, move the cursor to the column where a tab is desired and press <S/R Tab>. To remove a tab, move the cursor to the column where the tab is to be deleted and press <Cmd> <S/R Tab>. (Press <Ctrl+[> on earlier-model Ann Arbor terminals.)

<S/R Tab>	Sets tab
<Cmd> <S/R Tab>	Removes tab

<+TAB> OR <TAB>: MOVING FORWARD ONE TAB STOP

Pressing <+Tab> on Ann Arbor terminals or <Tab> on Ambassador and XL terminals moves the cursor to the next tab position to the right. Default tabs are set every 8 columns beginning in column 1.

<+Tab> or <Tab>	Moves cursor right to next tab stop

<-TAB> OR |SHIFT+TAB|: MOVING BACKWARD ONE TAB STOP

Pressing <-Tab> on Ann Arbor Terminals or <Shift+Tab> on Ambassador and XL terminals moves the cursor to the next tab position to the left. When the cursor is at or before the leftmost tab, pressing <-Tab> or <Shift+Tab> has no effect.

<-Tab> or <Shift+Tab>	Moves cursor left to next tab stop

<+WORD>: MOVING FORWARD ONE WORD

Pressing <+Word> on Ambassador and XL terminals moves the cursor forward one word.

<+Word>	Move cursor to next word

<-WORD>: MOVING BACKWARD ONE WORD

Pressing <-Word> on Ambassador and XL terminals moves the cursor backward one word.

<-Word>	Move cursor to previous word

SUMMARY OF EDITOR FUNCTION KEYS

Key Name	What Pressing the Key Does	What Pressing <Cmd>+<Key> Does
<Alt>	Moves cursor to alternate file	Pressing <Cmd> <Cmd> and then <Alt> repeats last command on command line
<Arg>	Allows you to issue a command	--
<Brk>	Allows you to issue a command	--
<Brk/Cmd>	Allows you to issue a command	--
<Bsp>	Moves cursor left, deleting characters	Leaves cursor in place, deletes text to left of cursor
<ChgWin>	Moves cursor from one windowed file to another	--
<Close>	Closes line	Restores closed text
<Cmd>	Allows you to issue a command	--
<CntlChar>	Makes block symbol on Ambassador and XL terminals	--
<CR>	Moves cursor to left margin of next line	Takes out of Command Mode
<Ctrl>	Shifts the keyboard into a third case; must be used with another key	--
<Ctrl+a>	Moves screen 16 characters to left	--
<Ctrl+b>	Moves cursor to alternate file	--
<Ctrl+c>	Interrupts editor command	
<Ctrl+s>	Moves screen 16 characters to right	--
<Ctrl+z>	Moves cursor from one windowed file to another	--

Key Name	What Pressing the Key Does	What Pressing <Cmd>+<Key> Does
|Ctrl+[|	Sets tab at cursor position	Removes tab at cursor position
|Ctrl+\\\|	Makes block symbol on Ann Arbor terminals	--
<DelChar>	Deletes character	Deletes all characters to right of cursor
<DelWord>	Deletes word	Restores deleted word
<Erase>	Erases text; leaves blank space	Restores erased text
<Home>	Moves cursor to top left corner of window	Moves cursor to bottom left corner of window
<Ins>	Moves into and out of Insert Mode	--
<Int>	Interrupts editor command	--
<Join>	Joins two lines of text	--
<Left>	Moves screen 16 characters to left	Moves screen back to original position
<+Line>	Moves top fourth of screen out of sight; brings equal number of lines into sight at bottom of screen	Moves screen so cursor is at top of screen
<-Line>	Moves bottom fourth of screen out of sight; brings equal number of lines into sight at top of screen	Moves screen so cursor is at bottom of screen
<LineFeed>	Moves cursor down one line	Moves cursor to bottom of window
<Mark>	Defines marked area of text	Turns off Mark Mode
<Open>	Opens blank line	--
<+Page>	Moves cursor forward one window	Moves cursor to end of file

Key Name	What Pressing the Key Does	What Pressing <Cmd>+<Key> Does
<-Page>	Moves cursor backward one window	Moves cursor to beginning of file
<Pick>	Replaces text in pick buffer	Inserts contents of pick buffer
<Repl>	Replaces characters after issuing interactive replace command	--
<Right>	Moves screen 16 characters to right	Moves screen so that column on which cursor is positioned is on left edge of window
<+Sch>	Searches from cursor position forward	Places word at cursor position into search buffer and searches for it from cursor position forward
<-Sch>	Searches from cursor position backward	Places word at cursor position into search buffer and searches for it from cursor position backward
<Spacebar>	Moves cursor one column to right	--
<Split>	Splits line into two lines	--
<S/R Tab>	Sets tab at cursor position	Removes tab
<+Tab>	Moves cursor right to next tab stop	--
<-Tab>	Moves cursor left to next tab stop	--
<+Word>	Moves cursor to next word	--
<-Word>	Moves cursor to previous word	--

Key Name	What Pressing the Key Does	What Pressing <Cmd>+<Key> Does
<→>	Moves cursor to right	Moves cursor to end of text on line; if sequence is repeated, cursor moves to right edge of window
<←>	Moves cursor to left	Moves cursor to left edge of window
<↑>	Moves cursor up	Moves cursor to top edge of window
<↓>	Moves cursor down	Moves cursor to bottom line of text in window

Appendix A

DEALING WITH CRASHES

There are two kinds of crashes you may experience when using e: individual crashes (which affect only you) and system crashes (which affect all users).

Individual crashes are caused when you:

- Edit in the same directory as another user.
- Fail to press a key for at least 30 minutes while in the editor.
- Run out of your allocated "memory space."
- Exit from a file using the exit abort command.

System crashes are caused by:

- Machine (hardware) problems.
- Program (software) problems.
- External events such as power failures and surges.

If you experience an individual or system crash, you must recover from that crash using the same kind of terminal (Ann Arbor, Ambassador, or XL) you were using when the crash occurred.

To recover from a crash, first type **e** <CR> in response to the % prompt. The editor will prompt you with the crash message:

```
The last time you used the Editor in this directory, you crashed or
  aborted.  You have these choices:

 1.  e will silently recover the last session and then update the screen;
     then you should exit before you continue editing.
       (Normally, select this option.)

 2.  e will replay the last session on the screen; you should exit before
     you continue editing.
       (Select this option if e continues to crash in response to Option 1;
       press the interrupt key just before e completes the replay.  The
       interrupt key on the e standard keyboard is control-\).

 3.  e will ignore the crashed or aborted session and forge ahead as per
     the arguments with which it was invoked.
       (Select this option only if you do NOT wish to recover the last
       session's work.)

 4.  e will return you to the shell, having done nothing.
      (Select this option if you do not know what to do and need
      assistance from your system administrators.)
 Type the number of the option you want, then hit <CR>:
```

In most instances, you should choose the first option.

If you run out of memory space or if you used the exit abort command (described in Section 3), choose the second option. In this case, watch the replay of your editing session on the screen and interrupt it by pressing <Int> or <Ctrl+c> just before you reach the place at which the crash occurred. The replay is performed quickly, so some skill is required to interrupt at the desired point. Finally, exit from the file in order to save it.

Appendix B

EDITOR WORK FILES

The editor maintains four hidden "work files" (.ec1, .ek1, .ek1b, and .es1). Since these files are created solely for the sake of the editor's internal mechanism, you cannot read and use them the way you can the #o and #p "changes files," which are created to protect you against accidental mistakes.

To see the names of these hidden "work files," type **ls -a** <CR> in response to the % prompt.

The .ec1 file is your change file. It is

- Used by the editor to update your file after a crash has occurred.
- Named .ec1 if you are working in your own directory or .ec1.*loginname* if you are working in someone else's directory.

The .ek1 file is your keystroke file. It contains a history of all your keystrokes and is

- Used by the editor for recovery purposes.
- Named .ek1 if you are working in your own directory or .ek1.*loginname* if you are working in someone else's directory.

The .ek1b file is your backup keystroke file. It is

- Used by the editor for recovery purposes.
- Created when have trouble recovering from a crash.
- Named .ek1b if you are working in your own directory or .ek1b.*loginname* if you are working in someone else's directory.

The .es1 file is your state file. It contains information about the last file edited in each directory, including filename, cursor position, line length, state of Insert Mode, tab positions, and contents of the search buffer. Thus, it allows you to edit the last file you worked on by typing **e** <CR> rather than **e** *filename* <CR> in response to the % prompt. It is

- Created by the editor every time you exit from a file.
- Named .es1 if you are working in your own directory or .es1.*loginname* if you are working in someone else's directory.

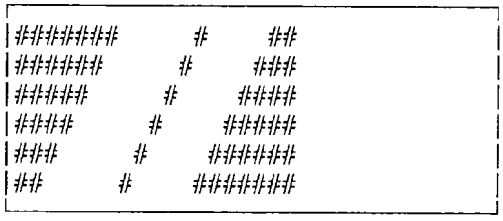
Appendix C

VARIATIONS OF THE COVER COMMAND

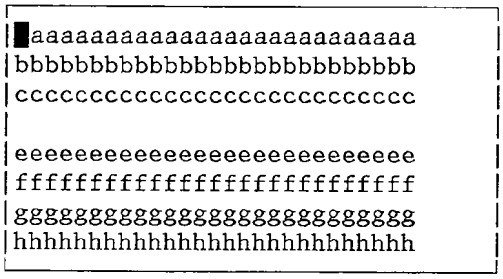
Section 3 describes how to use the cover command to insert rectangularly marked material from your close, erase, and pick buffers without shifting existing text to the right.

This appendix describes four similar commands: blot, -blot, overlay, and underlay.

The examples given in this appendix use material stored in the pick buffer. You can also use these commands with material stored in the close and erase buffers. In all cases, you must use rectangularly marked material and you must type the name of the buffer you use (do not press the key with the name of the buffer on it). Suppose you have placed the following material in your pick buffer by pressing <Mark>, moving the cursor, and pressing <Pick>:

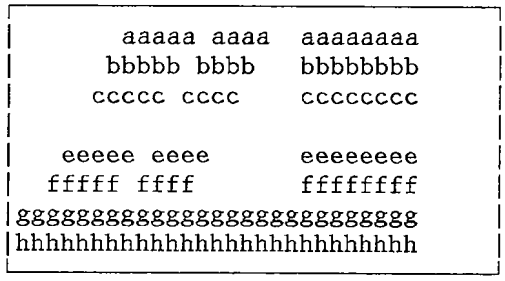


Place your cursor at the left margin of your file:



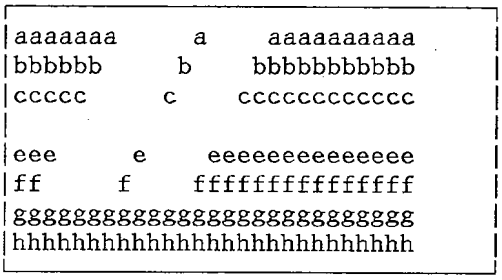
BLOT

If you press <Cmd> **blot pick** <CR>, everywhere a character appeared in your buffer, there will be a blank space in your file.



-BLOT

If you press <Cmd> **-blot pick** <CR>, everywhere a blank appeared in your buffer, there will be a blank space in your file.

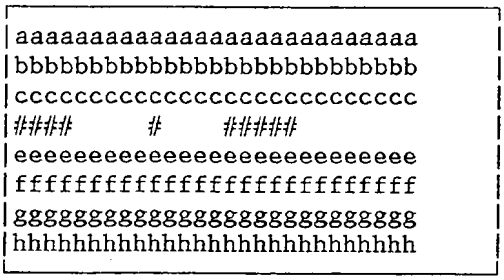


OVERLAY

If you press <Cmd> **overlay pick** <CR>, all characters in your buffer will replace characters in your text; blank spaces in the buffer will not replace characters in your file.

UNDERLAY

If you press <Cmd> **underlay pick** <CR>, characters in your buffer will replace only corresponding blank spaces in your file.



SUMMARY OF VARIATIONS OF THE COVER COMMAND

Command	What the Command Does
blot close	Places a blank space in your file everywhere a character appeared in the close buffer
blot erase	Places a blank space in your file everywhere a character appeared in the erase buffer
blot pick	Places a blank space in your file everywhere a character appeared in the pick buffer
-blot close	Places a blank space in your file everywhere a character appeared in the close buffer
-blot erase	Places a blank space in your file everywhere a character appeared in the erase buffer
-blot pick	Places a blank space in your file everywhere a character appeared in the pick buffer
overlay close	Replaces characters in text with corresponding characters from the close buffer
overlay erase	Replaces characters in text with corresponding characters from the erase buffer
overlay pick	Replaces characters in text with corresponding characters from the pick buffer
underlay close	Replaces blank spaces in your file with corresponding characters in the close buffer
underlay erase	Replaces blank spaces in your file with corresponding characters in the erase buffer
underlay pick	Replaces blank spaces in your file with corresponding characters in the pick buffer

Appendix D

UNIX COMMAND OPTIONS

You can add the following UNIX command options to the edit command when you first enter the editor. Many of these options change the way the editor works or the way your screen displays a file.

-BULLETS, -NOBULLETS: ADDING/REMOVING CURSOR BLOCK MARKINGS

Sometimes when you use the editor, bright blocks at the edges of the window indicate the row and column position of the cursor. If your window lacks these blocks or if you have removed them by issuing the -nobullets command option, you can produce them on your screen during the current editing session by issuing the following command:

% **e -bullets** *filename* <CR>

The editor is less expensive and faster to use when these bright blocks do not appear at the edges of your window. To remove them during the current editing session, issue the following command:

% **e -nobullets** *filename* <CR>

-HELP: ON-LINE EDITOR ASSISTANCE

This command option displays the currently available options and their values along with the version number of the editor.

% **e -help** <CR>

-INPLACE, INPLACE: EDITING A LINKED FILE

This command option allows you to edit and modify a linked file without breaking the link:

% **e -inplace** *filename* <CR>

Without this option, files that were originally linked to the edited file become linked to the comma file. The link is then broken when the comma file is deleted.

You can override the -inplace command by pressing <Cmd> **inplace** <CR>.

-NOCMDCMD: WASHINGTON OFFICE USERS

This command option enables personnel with some Ann Arbor terminals in Rand's Washington, D.C., office to solve certain communications problems:

% **e -nocmdcmd** *filename* <CR>

If you use this option and if you wish to return the last command executed to the command line, press <Cmd> <Spacebar> <Cmd> <Ctrl+b>. This line can be edited and then executed again by pressing <CR>.

-NORECOVER: RESTORING PREVIOUS VERSION OF FILE

This command option begins your current editing session as defined by the .es1 state file and ignores any .ec1 or .ek1 files:

% **e -norecover** *filename* <CR>

If the editor has crashed or if you have exited from a file by using the exit abort option, this option enables you to avoid having to respond to the crash message. Do not use this command unless you do not want to recover from that crash.

-NOSTICK, -STICK: RIGHT MARGIN CONTROL

If you want to type further than column 75 without having the cursor either return automatically to the next line (as it does when you are in the Word Wrap Mode) or stick on the right margin (as it does when you have turned off the Word Wrap Mode), first use the -nostick command option when you enter your file:

% **e -nostick** *filename* <CR>

Then turn off the Word Wrap Mode with:

<Cmd> **-wp** <CR>

You can turn the stick option back on by using the -stick command option when you enter your file:

% **e -stick** *filename* <CR>

You can turn the Word Wrap Mode back with:

<Cmd> **wp** <CR>

-NOTRACKS: EDITING A FILE AND RETAINING CURRENT WORK FILE

This command option begins your current editing session without using or replacing the work files (.ec1, ek1, and .es1) from your previous session:

```
% e -notracks filename <CR>
```

-REPLAY: RECOVERING FROM CRASH

This command option allows you to replay an editing session using your keystroke file (.ek1) if you cannot otherwise recover from a crash.

To use this option, you must first delete your .ec1 file if it exists:

```
% del .ec1 <CR>
```

Then rename your .ek1 and/or .ek1b files:

```
% mv .ek1 EK1 <CR>
% mv .ek1b EK1B <CR>
```

Finally, use the -replay command option:

```
% e -replay=EK1 <CR>
```

You can interrupt the replay by pressing <Int> or <Ctrl+c>. If you do not wish to watch the replay on your screen, use the following option:

```
% e -replay=EK1 -silent <CR>
```

SUMMARY OF UNIX COMMAND OPTIONS

Command	What the Command Does
e -bullets filename	Produces bright blocks at edges of window
e -help	Displays on-line help file
e -inplace filename	Allows you to modify file without breaking link
e -nobullets filename	Removes bright blocks from edges of window
e -nocmdcmd filename	Solves communications problems for some Ann Arbor terminals in D.C. office
e -norecover filename	Ignores any recovery processing of a prior crashed or aborted session
e -nostick filename	Causes cursor not to stick on right margin
e -notracks filename	Allows you to edit a file without disturbing the work files
e -replay=EK1	Uses keystroke file (renamed EK1) to replay editing session
e -replay=EK1 -silent	Uses keystroke file (renamed EK1) to replay editing session silently
e -stick filename	Causes cursor to stick on right margin

Appendix E

TERMINALS AND KEYBOARDS

The editor must be told the kind of terminal monitor and keyboard you are using so that it can interpret your keystrokes and place text properly on your screen.

The editor usually finds this information in the TERM environment variable that is defined in your .login file. For example, if you regularly use an Ann Arbor terminal, your .login file probably contains the following line:

```
setenv TERM aa; tset -e
```

If you regularly use an Ambassador or XL terminal, your .login file probably contains the following line:

```
setenv TERM aaa; tset -eH
```

DEFINING YOUR TERMINAL MONITOR

If you wish to use a terminal monitor that is different from the one defined in your .login file, use the UNIX command option:

% **e -terminal=***xxxx filename* <CR>

For *xxxx*, substitute the appropriate abbreviation from Table E.1. The UNIX command option -dtermcap tells the editor not to use its compiled description of your terminal. Instead, the editor will use your TERMCAP environment variable, or, if that does not exist, the description found in /etc/termcap.

% **e -dtermcap** *filename* <CR>

DEFINING YOUR KEYBOARD

The TERMCAP environment variable and /etc/termcap contain descriptions of the terminal monitor, but not of the keyboard. Thus if you wish to use a terminal that is different from the one you usually use and if the editor does not have a compiled description for your terminal type, the editor will assume you are using a standard keyboard unless you use the -keyboard UNIX command option:

% **e -keyboard=***xxxx filename* <CR>

For *xxxx*, substitute the appropriate abbreviation from Table E.1. If you wish to override the compiled description to indicate you are using a standard keyboard, substitute *standard* for *xxxx*. The standard editor keyboard is designed to be used on terminals with no function keys and can be used with all terminal monitors. See Table E.2 for a description of how to use a standard keyboard when performing editor functions.

Table E.1

ABBREVIATIONS FOR TERMINAL AND KEYBOARD TYPES

Terminal and Keyboard Types	Abbreviations
Ann Arbor Ambassador and XL Series	aaa, ambas, ambassador, aaa*N*, aaas, aaa-*N* (*N* is 18, 20, 22, 24, 26, 28, 30, 36, 40, 48, or 60)
Ann Arbor S001901/2	aa, aa0, annarbor, default
Ann Arbor (Model Q2878)	aa1
DEC VT100	vt100, vt100w,cit101, cit101w, tab132, tab132w, dmx14, dmx14w
DEC VT52	vt52
Heathkit H19 & H89	h19, k1
Lear Siegler ADM3a	3a, adm3a, la (ell)
Standard keyboard	standard
Sun workstation	sun

You can also tell the editor you are using a different kind of keyboard by using the following UNIX command:

% **setenv EKBD** *xxxx* <CR>

For *xxxx*, substitute the appropriate abbreviation from Table E.1. Note, however, that the -keyboard option overrides this command.

Finally, you can tailor your own keyboard specifications by using the following UNIX command option:

% **e -kbfile=***specialfile filename* <CR>

For *specialfile*, insert the name of a file that contains a description of your keyboard's characteristics.

Table E.2

USING A STANDARD KEYBOARD WHEN PERFORMING EDITOR FUNCTIONS

Editor Function Keys	Comparable Keystrokes on a Standard Keyboard
<→>	<Ctrl+l>
<←>	<Ctrl+h>
<↑>	<Ctrl+k>
<↓>	<Ctrl+j>
<Alt>	<Ctrl+-> or <Ctrl+x+a>
<Bsp>	<Ctrl+c>
<ChgWin>	<Ctrl+x+w>
<Close>	<Ctrl+v>
<Cmd>	<Ctrl+a>
<CntlChar>	<Ctrl+x+c>
<DelChar>	<Ctrl+w>
<Erase>	<Ctrl+x+e>
<Home>	<Ctrl+g>
<Ins>	<Ctrl+z>
<Int>	<Ctrl+\>
<Join>	<Ctrl+x+j>
<Left>	<Ctrl+x+h>
<+Line>	<Ctrl+f>
<-Line>	<Ctrl+d>
<Mark>	<Ctrl+u>
<Open>	<Ctrl+o>
<+Page>	<Ctrl+r>
<-Page>	<Ctrl+e>
<Pick>	<Ctrl+p>
<Repl>	<Ctrl+]> or <Ctrl+x+r>
<CR>	<Ctrl+m>
<Right>	<Ctrl+x+l>
<+Sch>	<Ctrl+y>
<-Sch>	<Ctrl+t>
<Split>	<Ctrl+x+b>
<+Tab>	<Ctrl+i>
<-Tab>	<Ctrl+x+u>
<Tabs>	<Ctrl+x+t>
<+Word>	<Ctrl+n>
<-Word>	<Ctrl+b>

NOTE: The notation <Ctrl+x+key> instructs you to hold down <Ctrl> while pressing the <x> and then the indicated second key.

SUMMARY OF TERMINAL AND KEYBOARD OPTIONS

Command	What the Command Does
e -dtermcap filename	Overrides compiled code for terminal type
e -keyboard=xxxx filename	Tells editor you are using keyboard defined by xxxx
e -terminal=xxxx filename	Tells editor you are using terminal defined by xxxx

INDEX