

6800/6809 SEDT

V 1.3

USER'S MANUAL



SOFTWARE DYNAMICS

2111 W. Crescent, Suite G, ▲ Anaheim, CA 92801

SOFTWARE DYNAMICS © COPYRIGHT 1980

6 8 0 0 / 6 8 0 9

S E D I T

U S E R ' S

M A N U A L

SEDT V1.3

1st Printing

COPYRIGHT (C) 1980 SOFTWARE DYNAMICS

NOTICE

This manual describes Software Dynamics SEDIT Version 1.3. Software Dynamics has carefully checked the information given in this manual, and it is believed to be entirely reliable. However, no responsibility is assumed for inaccuracies.

SD reserves the right to change the specifications without notice.

```
*****
** This manual describes software which is a proprietary product **
** of Software Dynamics (SD). SD software is licensed for use on a **
** single copy per computer basis, and is covered by U.S copyright **
** laws. Unless a written exception is obtained from SD, the soft- **
** ware must be used only on the single computer whose unique, SD- **
** assigned serial number matches that for which the software was **
** purchased. Copying the software for any purpose other than **
** archival storage, or use of the software on other than the as- **
** signed serial numbered CPU is strictly prohibited. SD assumes **
** no liability regarding the use of the software. **
** Certain software programs and datafiles are delivered for use **
** in an encrypted format. The content of such programs and data **
** are considered to be a trade secret of SD. Attempts or suc- **
** cess at breaking the encryption, publication of the results of **
** such attempts or successes, or copying, storage or use of such a **
** file in clear text form will be treated as theft of a trade sec- **
** ret, and prosecuted as such. **
** POSSESSION OR USE OF THIS MANUAL OR THE SOFTWARE IT DESCRIBES **
** CONSTITUTES AGREEMENT BY THE USER TO THESE TERMS. **
*****
```

This manual and the software it describes are the copyrighted property of Software Dynamics.

TABLE OF CONTENTS

SECTION I: INTRODUCTION	1
SEDIT--AN EASY-TO-USE SCREEN EDITOR	1
OPERATION OVERVIEW	1
SECTION II: GENERAL COMMANDS	4
SEDIT COMMANDS	4
SECTION III: SINGLE KEYSTROKE COMMANDS	5
INSERT ME	5
<RETURN>	5
<TAB>	5
<CONTROL-H> (LEFT ARROW)	6
<CONTROL-L> (RIGHT ARROW)	6
<CONTROL-K> (UP ARROW)	6
<CONTROL-J> (DOWN ARROW)	6
<CONTROL-E> (ERASE TO END OF LINE)	7
<CONTROL-X> (CANCEL LINE)	7
<CONTROL-F>	7
<CONTROL-R>	7
SECTION IV: SPECIAL COMMANDS	8
<ESC><RETURN>	8
<ESC><UP ARROW>	8
<ESC><DOWN ARROW>	8
<ESC><LEFT ARROW>	8
<ESC><RIGHT ARROW>	9
<ESC><DIGIT><DIGIT>	9
<ESC>@	9
<ESC><COLON>	9
<ESC><PERIOD>	9
<ESC><MINUS>	9
<ESC>?	10
<ESC>A	11
<ESC>B	11
<ESC>C	11
<ESC>D	11
<ESC>F	11
<ESC>H	12
<ESC>J	12
<ESC>K	12
<ESC>N	13
<ESC>Q	13
<ESC>R	13
<ESC>X	14
<ESC>Z	14
<ESC><TAB>	15
<ESC> <	15
<ESC> >	15
<ESC>T<DIGIT>	16
SECTION V: DISASTERS	17

SEDIT -- An Easy-To-Use Screen Editor

SEDIT is a very easy to use, CRT screen-oriented text editor. SEDIT can be used to both create and modify arbitrary text documents, such as letters, manuscripts, input text for SD's TYPE text formatter, and even computer program source files. It is fundamentally a "what you see is what you get" editor, and, as such, is suitable for use by both novice computer users and old computer hands. It takes about an hour of training time for a novice to become reasonably proficient with SEDIT.

Operation Overview

This section gives a rough idea of what operation of the screen editor is like. The section "SEDIT Commands" gives detailed descriptions of all editor functions and some simple examples.

The operator selects a text (disk) file to be modified (the "old" file). SEDIT essentially makes a new disk file containing the text from the old file, plus whatever changes (insertions, deletions, and movements of text) that the operator specifies (creation of a totally new text file is accomplished easily if one simply treats the old file as one containing no text).

SEDIT establishes a window in which editing may occur. Text is copied from the front of the old file to fill the window. The window may be moved down the old file; as it does, new text from the old file is added to the bottom of the window. When the window becomes overly full, text is taken from the top of the window and written sequentially at the end of the new file. The window is moved down the entire old file to copy text to the new file.

All changes, insertions, deletions and text movement are performed by the operator only on the window contents. A cursor (a solid blotch, blinking character or movable underline) is used to designate a particular character in the window. At any time, SEDIT displays a CRT screenful of text around the current location of the cursor. All editing occurs at the position specified by the current cursor location. Editor commands allow the cursor to be moved up one line on the screen, down one line, to the right one character, and to the left one character. This movement of the cursor designates a different character in the window. By repeated application of commands to move the cursor, it may be positioned anywhere on the screen (a repeat key is very helpful here). Attempting to move the cursor off the bottom or top of the displayed screen will cause the CRT to roll the currently displayed text up or down one line, respectively, allowing the operator to display any block of lines that will fit on the CRT screen (note that the cursor always stays on the CRT screen). **Note: moving the cursor implicitly moves the editing context to the character shown underneath the cursor.

SEEDIT USER'S MANUAL
SECTION I: Introduction

Other SEEDIT commands allow the cursor to be moved up or down as many times as there are lines on the CRT screen; this selects and displays the next or the screen of text in the window.

To insert new text, the cursor is positioned in the window at the place where text needs insertion; then the desired text is simply typed. SEEDIT will move characters to the right as needed to make room for the newly inserted text. Keystrokes which cause insertion are all keystrokes with visible printed representations (i.e., 0 to 9, A thru Z, punctuation, etc.), spaces, and carriage returns. A carriage return will cause a new line to be formed. Note that even though a carriage return has no visible representation, a "character" slot in the window is reserved to hold it.

If a mistake is made while inserting new text, the <RUB>out (or ete) key can be used to erase it. Pressing rubout causes the character immediately to the left of the cursor location to be deleted from the text. The cursor will be moved to the location occupied by the deleted character. If the cursor is at the left side of the screen, the carriage return character of the previous line will be deleted, and the previous line and the current line will become a single line.

<RUB>out can also be used to delete unwanted text by positioning the cursor to the right of the unwanted text, and pressing <RUB>out as many times as needed. However, another method can be used. The cursor can be positioned on top of the leftmost character to be deleted, and <Control-U> depressed. This will delete the character UNDER THE CURSOR. Characters to the right will be shuffled left to fill the gap, thus leaving the cursor over the character which was immediately to the right of the character that was deleted. Repeating <Control-U> will then delete from left to right (whereas <RUB>out deleted from right to left).

Virtually all the editing required to a document can be performed via positioning the cursor, inserting, and/or deleting using <RUB>out or <Control-U> as necessary.

Entry of text for documents or letters can be easily accomplished using the margin control facilities of SEEDIT. The operator can specify, using a set of special commands, the position of left and right margins, and tab stops. Then text entry may be performed by simply typing the raw text as a continuous stream of keystrokes. SEEDIT will automatically break off lines that are longer than the right margin, and continue them on the next line starting at the specified left margin. Typing errors may be corrected without regard to margin boundaries by use of the <RUB>out key. Correction of text after it is entered is performed as described above; re-justification of the text after correction can be performed using the Justify command.

SEdit USER'S MANUAL
SECTION I: Introduction

Other editing functions, such as searching for a specific string, removing a block of text ("cutting") so it may be inserted at another point ("pasting"), making repeated identical changes to the text, etc., are accomplished with special keystroke sequences that begin with the <ESC>ape key. Depressing this key signals the editor that a special function is being requested. The operator must type another key to specify what function is desired (i.e. <ESC>ape followed by H for HELP). The operation of the special commands is described below.

Type to INSERT; <RUBOUT> deletes insert; <CTL-U> deletes under cursor
<UP> <DOWN> <LEFT> <RIGHT>: Move cursor in specified direction
ESC <UP> or <DOWN>: Move cursor one screen in specified direction
ESC <LEFT> or <RIGHT>: Move cursor left or right to start of word
ESC <CR>: Move cursor to start of current line
ESC <PERIOD>: Mark cursor location as start of 'Delete' or 'Justify'
ESC -: Delete text from marked point to cursor
ESC ?: Redisplay the screen, centering cursor context
ESC A: Again, repeat change and find next occurrence
ESC B: Move cursor to top of text window
ESC C <string> ESC: Change found string to <string>
ESC D: Delete word
ESC F <string> ESC: FIND <string>
ESC H ESC: Displays this text
ESC J: Force text between marked point and cursor to fit margins
ESC K: Delete (Kill) rest of line
ESC N: Find next occurrence of <ESC>F string
ESC Q: Quit editing, don't update file
ESC R: Restore deleted text here
ESC X: Exit edit mode, update file
ESC Z: Move cursor to bottom of text window
ESC < or ESC >: Set Left Margin/Right Margin
ESC <TAB>: Set/Reset Tab Stop
ESC T <digit>: Select new Ruler from SEDIT.TAB
<CTL-E> erases to end-of-line; <CTL-X> cancels line
<CTL-F> moves cursor to front of line; <CTL-R> moves cursor to end-of-line
ESC<digits>: Repeat next command <digits> times
ESC @: Moves cursor to end of line
ESC <COLON>: Swaps cursor position with marked text location

SEDIT Commands

SEDIT commands consist of following types:

- Simple Cursor Movement
- "Insert Me"
- <Tab>
- Delete Character
- Cancel Line or Erase to End of Line
- Special

Unless an <ESC>ape character is typed immediately before a keystroke, the keystroke is cursor movement, insert-me, delete character, cancel or erase to end of line, a tab character, or illegal. The arrow keys on the CRT are simple cursor movement keys, and are used to move the cursor in the direction given by the arrow. All characters with visible representations, and the space character are treated as "Insert Me" commands; carriage return is also an "Insert Me" command that has some side effects. Tabs cause conventional tabbing to occur. The <RUB>out key is a "delete character to left of cursor" command; <Control-U> is a "delete character under cursor" command. Erase to end of line deletes part of a line; cancel line deletes the entire contents of a line. An <ESC>ape character tells SEDIT that the next keystroke specifies which of the special functions is to be invoked.

All other keys except <ESC>ape are illegal, and will echo <BEEP> if depressed.

If <ESC>ape is depressed, the keystroke following the <ESC>ape is used to select from a set of special functions. The section "special commands" below lists all the legal special functions and describes their actions; any <ESC>ape keystroke sequence not listed causes <BEEP> to be echoed and the <ESC>ape keystroke sequence is ignored. The character immediately following the <ESC>ape keystroke is treated as upper case if it is a lower case letter.

Commands which are not defined (i.e., most control characters and unused letters following <ESC>ape), or which cannot be executed for some reason, cause SEDIT to echo <BEEP>. Whenever the editor echoes <BEEP> as a response to a command, the next keystroke entered by the operator will be interpreted as an editor command.

Single Keystroke Commands

"Insert Me"

The keystroke hit is inserted into the text where the cursor indicates; any text to the right is moved right to make room, and the cursor is moved right one character to prepare for another "Insert Me" command. If the cursor is at the right side of the screen, a <BEEP> will be echoed (the editor cannot insert and display the character!) and the command will be ignored (typing <RETURN> will split the text line so that more "Insert Me"s are legal). If the cursor is beyond the <RETURN> character that ends the line, blanks are automatically inserted to extend the line to the point of the insertion before insertion takes place. In any case, if the window is full, SEDIT will attempt to make more room before inserting the character by moving the first screenful of text in the window to the new file (if the operator is relatively fast, a noticeable delay may occur before typed-ahead characters start being echoed again; this is normal). If the insertion is occurring in the first screenful of text in the window then this cannot be done without losing the operator's cursor position, and so SEDIT will <BEEP> and ignore the command (this is an unusual circumstance).

<RETURN>

This command causes a <RETURN> character to be inserted into the text at the point of the cursor. Characters to the right of the cursor are moved down to the next screen line, and the cursor is positioned at the left of the characters moved down, so that more "Insert Me" commands may be entered. This command is used at the end of a screen line, or to break a line into two parts. It may be entered even when the cursor is at the right side of the screen, unlike an "Insert Me" command. Note that even though the <RETURN> character is invisible, it is present at the end of every line on the screen and can be deleted. If the window is full, SEDIT attempts to make more room as described in "Insert Me".

<TAB>

The <TAB> key is used to effectively insert one or more spaces in the buffer from the current cursor location until the next tab stop, as indicated by the current ruler, is reached. SEDIT 1.2 actually places spaces in the text; SEDIT 1.3 and above place actual <TAB> characters in the text, so that displaying the text causes actual tabbing to occur. This allows the tab boundaries to be changed, and have the text display automatically adjusted, without changing the text.

SEDIT USER'S MANUAL
SECTION III: Single Keystroke Commands

<Control-H> (Left Arrow)

This causes the cursor to be moved left one character position on the CRT screen. If the cursor is past the <RETURN> on a line, it is moved left until it is over the <RETURN>. If the cursor is at the left hand side of the screen, it is placed at the right hand end of the line above; if the line above is wider than the CRT, the cursor will be placed at the rightmost position of that line on the CRT, and <BEEP> will be echoed. The screen will scroll down a line if the cursor was at the top left hand corner. If the cursor selects the character at the very top of the window, this command is illegal.

<Control-L> (Right Arrow)

This causes the cursor to be moved right one character position on the CRT screen. If the cursor is at the right hand side of the screen, a <BEEP> will result and the cursor will not be moved.

<Control-K> (Up Arrow)

This causes the cursor to be moved up the screen by one CRT line. If the cursor is at the top of the screen, the screen is rolled down one line and the previous line in the window is displayed as the top line of the screen. If the cursor is logically at the top of the window, it is moved to the top left hand corner. If the cursor is in the top left hand corner, SEDIT echos a <BEEP> and the cursor is not moved.

<Control-J> (Down Arrow)

This cause the cursor to be moved down the screen by one CRT line. If the cursor is at the bottom of the screen, the screen is rolled up one line and the next line from the text window is displayed as the bottom screen line. If the bottom of the text window is reached, another line is fetched from the old file and appended to the bottom of the window. SEDIT will make room for a new line by moving a screenful of text from the top of the window to the end of the new file; once this happens, the operator may not go back and edit the screenful that was moved without exiting SEDIT and re-entering. If no more lines of text are available from the old file, SEDIT <BEEP>s but does not move the cursor.

SEDIT USER'S MANUAL
SECTION III: Single Keystroke Commands

<Control-E> (Erase to End of line)

This key causes the characters under the cursor and to the right of the cursor on the same screen line to be deleted. The cursor is not moved. Text deleted by this command cannot be recovered; see <ESC>K for a similar command whose effects are reversible.

<Control-X> (Cancel Line)

This key causes all the characters on the same line as the cursor to be deleted; the <RETURN> character at the end of the line is not deleted. The cursor is placed at the left edge of the screen on the same line. Text deleted by this command cannot be recovered; see <ESC>K for a similar command whose effects are reversible.

<Control-F>

This key causes the cursor to be moved to the currently selected left margin position on the same line.

<Control-R>

This key causes the cursor to move to the right end of the line currently holding the cursor.

Special commands

Special commands are those used for less common functions, such as moving the cursor long distances, deleting large blocks of text, moving text blocks around, and other miscellaneous functions. All special commands are invoked by typing <ESC>ape key, followed by a keystroke to select the desired special function. The special commands are listed below.

<ESC><RETURN>

This command causes the cursor to be moved to the currently selected left margin position on the same line. The <Control-F> command should be used.

<ESC><Up Arrow>

This command causes SEDIT to display the screenful of text preceding the text currently displayed, and to position the cursor roughly halfway down the screen at the currently selected left margin. Attempting to go up beyond the beginning of the window will cause a <BEEP> and the cursor will not move.

<ESC><Down Arrow>

This command causes display of the screenful of text following the text currently being displayed; the cursor is positioned roughly halfway down the screen at the currently selected left margin. When the screen being displayed is at the end of the window (<ESC>Z can do this), <ESC><Down Arrow> will cause the next screenful to be taken from the old file. If SEDIT does not have enough room in the window to hold the new text, text at the beginning of the window is moved to the end of the new file to make space. If only a partial screenful of text is displayed when the command is executed, the screenful will be filled out instead of displaying the next screenful.

<ESC><Left Arrow>

This command causes the cursor to be moved to the beginning of the word to the left of the cursor (a "word" begins with a letter or a digit). If the cursor is at the left margin position, and the contents of the left margin are blank, this command will position the cursor at the left side of the screen. If the cursor is at the left side of the CRT screen, it is moved to the end of the preceding line. Should this require the cursor to move off the top of screen, the screen will be re-displayed so that the line on which the cursor will be placed will roughly be centered.

SEDIT USER'S MANUAL
SECTION IV: Special Commands

<ESC><Right Arrow>

This command causes the cursor to be moved to the beginning of the word following the current cursor location. If the cursor is anywhere within the last word of a line, it is moved to the end of the line. If the cursor is at the end of a line, it is moved to the left side of the CRT screen on the next line. SEDIT will <BEEP> if the cursor is at the end of the text window and will not move the cursor.

<ESC><Digit><Digit>...

This command specifies a "repeat" count used to repeat the following command (some commands ignore the repeat count; in particular, single character inserts/deletes do not repeat). For example, to move the cursor right 13 places, type <ESC>13<Right arrow>.

<ESC>@

This command causes the cursor to move to the right end of the line on which the cursor is currently positioned. The <CONTROL-R> key should be used instead.

<ESC><Colon>

This command causes the cursor position and the location of the distinguished point to be exchanged; the new cursor location is then displayed. This is useful when one needs to remember a particular place in the text, go inspect something else, and then move the editor's attention back to the particular place.

<ESC><Period>

This command is used in conjunction with <ESC><Minus> and <ESC>R to effect a textual "cut and paste" function. It simply marks a point in the text window which is used later by <ESC><Minus>. If the window is moved down the old file, causing the marked place to move out of the window, the marked place is lost. Moving or changing text in the window above the marked point causes the marked point to be forgotten. The marked place can also be used to specify the beginning of a region of text to be justified; see <ESC>J.

<ESC><Minus>

This command is used with <ESC><Period> to effect a textual "cut" or deletion of a large block of text. First, <ESC><Period> is used to mark a point in the text buffer, which is the beginning of the region to be deleted or moved; then the cursor is positioned to the character beyond the end of the text region desired. At this point, <ESC><Minus> is depressed. SEDIT will remove the text between the selected limits from the file and the CRT display. The deleted text is saved for use by <ESC>R. If the text block deleted is large, SEDIT will use a disk file, DELETED.TMP, to store the deleted text. The the last block of deleted text is always placed in DELETED.TMP when SEDIT is exited via <ESC>Q or <ESC>X; this allows blocks of text to be

shuffled arbitrarily between different files. This command ignores any repeat specification.

<ESC>?

This command causes SEDIT to re-display the text on the CRT such that the line on which the cursor is positioned is displayed roughly halfway down the CRT screen. The logical position of the cursor with respect to the text is not changed. This is useful when the text surrounding the cursor in both directions needs to be examined, or if (rarely!) the CRT fails to display the text properly.

SEDIT USER'S MANUAL
SECTION IV: Special Commands

<ESC>A

This command changes the text under the cursor, if it matches the last specified "find string" (See <ESC>F), to the last specified "change string" (See <ESC>C), and then causes SEDIT to locate the next occurrence of the "find string". The newly found occurrence is then displayed, with the cursor on the first character of the found sequence. The command is logically identical in function to the commands <ESC>C(string)<ESC><ESC>N, and is best thought of as "again, change it and find the next occurrence".

<ESC>B

This command moves the cursor to the first character in the window, and displays the first screen of text in the window on the CRT. The cursor is positioned to the upper left hand corner. This command is always legal.

<ESC>C

This command is used to change a string found by using <ESC>F, <ESC>N or <ESC>A to a string of characters specified in the command. After the user types <ESC>C, the line currently occupied by the cursor is blanked on the screen; the last "change string" specified is displayed, and the cursor is moved to the left side of the screen. The user may edit the Change string using <Left Arrow>, <Right Arrow>, <Control-U>, <RUB>out, Insert-Me, <Control-E> or <Control-X> characters (if the last "change string" is not what is desired, the operator may strike <Control-X> to erase it completely). The operator indicates the desired "change string" is complete by typing <ESC>; SEDIT will re-display the text that was erased when <ESC>C was typed, and then it will change the occurrence of the "find string" under the cursor to the specified "change string". The "find string" is NOT saved for <ESC>R. Note: if the cursor is not positioned at the first character of a string that matches the "find string", the editor will <BEEP> and no change will be made.

<ESC>D

This command causes the word whose first letter is under the cursor to be deleted from the text buffer. A "word" is defined as any sequence of letters or digits that are not separated by some other character. If the cursor is not over a letter or digit, the editor will <BEEP> and the command will not be executed. This command can be invoked when the cursor is in the middle of a word, in which case the editor only deletes the part of the word under and to the right of the cursor. The deleted word can be restored using <ESC>R.

<ESC>F

This command is used to make the editor search for a sequence of characters specified by the command. The string specified by the user is referred to as the "find string" throughout the rest of this document. After the user types

SEDIT USER'S MANUAL
SECTION IV: Special Commands

<ESC>F, the line currently occupied by the cursor is blanked on the screen; the last "find string" searched for is displayed, and the cursor is moved to the left side of the screen. The user may edit the "find string" using <Left Arrow>, <Right Arrow>, <Control-u>, <RUB>out, Insert-Me, <Control-E> or <Control-X> characters (usually, the last string searched for is not what is desired, and the operator strikes <Control-X> to erase it completely). A "find string" may not include the <RETURN> character. The operator indicates the desired "find string" is complete by typing <ESC>; SEDIT will re-display the text that was erased when <ESC>F was typed, and then it will search for the next occurrence of the "find string" past (to the right or below) the cursor in the text window (note that trailing blanks in the "find string" ARE significant). An instance of the "find string" exactly under the cursor is ignored. If the desired string is found, the page of text around the string is displayed, and the cursor positioned to the first character of the found occurrence. If the desired string cannot be found in the text window, the editor <BEEPS> and the cursor is not moved (searching past the end of the text window is accomplished by use of <ESC>N). Note that when searching for a "find string", character cases must match exactly. If the repeat count is 1 (see <ESC><digit>), an "anchored" search is performed; SEDIT will only look for the "find string" at the beginning of a line (<ESC>A and <ESC>N will perform an anchored search only if the <ESC>F command was specified as an anchored search).

<ESC>H

This command causes the editor to display a short command summary. The operator must hit <ESC> to continue editing.

<ESC>J

This command justifies text between the selected distinguished point (see <ESC><Period>) and the cursor location (between the cursor location and the next paragraph break if no distinguished point has been established) to force it to fit within the specified margin boundaries. Lines that are too long are broken off at the rightmost blank that does not exceed the margin; lines that are too short are filled to the maximum that does not exceed the right margin.

<ESC>K

This command is used to delete one or more lines of text. <ESC>K by itself will delete text starting at, and to the right of, the current cursor location, up to and including the <RETURN> character that ends the current line. If the cursor is at the left side of the screen, then the entire line is deleted. Specifying a repeat count (see <ESC><Digit>...) causes the specified number of lines to the right of, as well as below, the position of the cursor, to be deleted. Lines below the line(s) deleted are moved up to

SEDIT USER'S MANUAL
SECTION IV: Special Commands

fill the gap. The deleted text is saved for use by <ESC>R; for more detail on where the deleted text is stored see <ESC><Minus>.

<ESC>N

This command causes the editor to locate the Next occurrence of the "find string" (an occurrence of the "find string" exactly under the cursor when the command is invoked is ignored). This command will cause the editor to move the text window down the file if the "find string" cannot be found in the current window. If the desired string is eventually located, the editor will display the page of text containing the string, and position the cursor to the first character of the found occurrence. If no occurrence of the "find string" is located, the editor will display a page of text near the end of the file and <BEEP> to indicate failure. Repeated use of <ESC>N provides a convenient way of finding the desired instance of the "find string".

<ESC>Q

This command allows the operator to abort an edit session WITHOUT perserving the changes made. The editor will clear the screen, and ask, "Quit without update? ". Any answer that starts with the letter Y (ignoring case) will cause the editor to simply abandon all changes made; The file specified for editing is left intact. Any other answer causes the editor to assume the operator is confused; an automatic <ESC>H is performed. The operator must type <ESC> to continue editing.

<ESC>R

This command is used to "restore" deleted text into the text window. The last block of text deleted by <ESC><Minus>, <ESC>D or <ESC>K is inserted into the text window at the location of the cursor on the screen. The screen is suitably updated to reflect the change. In order to make room in the text window for the restored text, some text at the top of the window may be moved to the end of the new file, and so cannot be re-edited without re-invoking SEDIT. It is possible that SEDIT cannot make enough room in the text window to insert the deleted text; if this occurs, SEDIT will echo <BEEP> and the restore will not be performed. This command ignores a repeat specification.

The restore command is used primarily for two purposes: moving a block of text, and replicating a block of text. To move a block of text, the block is first located and deleted (using <ESC>K or <ESC><Minus>); the cursor is moved to the desired location for the text block, and <ESC>R performed. If the insertion point is below the deletion point, the operator can then move down the file until the insertion point is located, and restore the text. Since the text window moves down through the old file, the process of deleting the desired block may leave the cursor sufficiently

SEDIT USER'S MANUAL
SECTION IV: Special Commands

far past the desired insertion point that the insertion point has been moved out of the text window, and into the new file. In this case, exiting (<ESC>X) will leave the block of deleted text in the file DELETED.TMP, so that upon re-entering SEDIT the block of text may be restored anywhere within the file. The operator can then easily go forward to the insertion point.

The second use is replicating a text block. To do this, the desired block is deleted, and then immediately restored. This leaves the original text intact; then the deleted text can be inserted as many times as desired by simply positioning the cursor properly and doing a restore.

A rare, third use is the insertion of text from an already existing file. To do this, the text to be inserted must be copied into DISK:DELETED.TMP before SEDIT is invoked. The first <ESC>R executed will cause the contents of DELETED.TMP to be inserted into the text window.

<ESC>X

This command terminates the editing process. All changes made are moved to the new file; this process may take a while if a large file was being edited. The original text file is renamed to XXX.BAK, where XXX.YYY was its original name, and the changed text is placed in the specified file (Note: an intermediate file, EDITOR.TMP, is used by the editor during the editor process).

<ESC>Z

This command causes the screenful of text at the end of the text window (NOT necessarily the End of the File) to be displayed, and the cursor positioned at the very end of the displayed text. This command is used generally with small documents (those that fit entirely in the text window) to get to End of File. An easy way to get to the actual end of the text file in a large document is to search for (<ESC>F) a string known not to be in the text; the string ENDFILE generally works well for this purpose.

SEDIT USER'S MANUAL
SECTION IV: Special Commands

<ESC><TAB>

This command causes a new tab setting to be set at the current cursor column. If a tab setting already exists at that column, it is made to go away. The screen will be re-displayed if it has a tab in it; thus, changing the tab settings will change the appearance of text containing <TAB> characters. Note: The current setting of tab stops, left margins and right margins within the editor are only valid for the editing session in which they are made; they cannot be saved permanently. However, the <ESC>T command can be used to specify a set of tab stops (and margin selections) from a permanent file.

Whenever a tab stop, or left or right margin are adjusted, or a new ruler is selected, the ruler is displayed. The line containing the cursor is erased, and the rightmost digit of the column number of each column is displayed in that column. Columns whose column number is a multiple of 10 have the 10's digit displayed in that column instead of a zero. The current setting of the left margin is shown by a column containing the "<" character instead of a digit; the right margin (if not at infinity) is shown by a ">" character, and all tab stops are shown by "+" characters. Columns beyond the right margin are not numbered. The ruler below has a left margin at column 15, tabs at 23 and 45, and a right margin at 65. The operator may now make further adjustments to the ruler, using the displayed digits to simplify counting. The operator signals completion of adjustments by executing any SEDIT command other than a ruler adjustment. When all adjustments to the ruler are complete, the erased line of text is re-displayed in place of the ruler.

Sample ruler display:

```
12345678911234<6789212+456789312345678941234+6789512345678961234>
```

<ESC> <

This command sets the left margin to the column in which the cursor is currently located (the left margin is where the first character inserted in a line will be placed). The old left margin setting is lost. SEDIT will automatically cancel all tab settings to the left of the newly specified Leftmargin; this may change the appearance of the text. A left margin causes SEDIT to automatically insert left margin spaces after an inserted CR or automatic wrap due to line overflow.

<ESC> >

This command sets the right margin to the column in which the cursor is currently located (the right margin is the rightmost place where a character is allowed to be). Tab settings beyond this point will be lost; this may change the appearance of the text. Selecting a right margin will cause

SEDIT USER'S MANUAL
SECTION IV: Special Commands

SEDIT to automatically move words at the right margin to the beginning (the left margin) of the next line and to re-justify until the next paragraph break when inserting text in that screen line. Setting the right margin to the rightmost column of the screen causes the right margin to be set to infinity; line "wrap" cannot then occur.

<ESC>T<Digit>

This command is used to select a new Ruler from the file DISK:SEDIT.TAB. The <Digit> specified selects a line from the SEDIT.TABS file if the digit is greater than zero; the digit "1" selects the 1st line of the file, etc. (a maximum of 9 rulers may be stored in SEDIT.TAB file). SEDIT will <BEEP> if SEDIT.TAB does not exist, or does not have enough rulers in it to satisfy the request. <ESC>T0 removes all tab stops and resets the left margin to 0, and the right margin to infinity; this may be performed even if SEDIT.TAB does not exist. Selecting a new ruler causes any previous tab stops, left or right margin specifications to be lost, and replaced by the newly selected settings. Modifications to the newly selected ruler may be made immediately if desired.

When SEDIT is invoked, an automatic <ESC>T1 is executed, so ruler "1" is the default ruler. If the file SEDIT.TAB does not exist, an automatic <ESC>T0 is performed instead. Since the rulers in SEDIT.TAB are simply lines of text, SEDIT may be used to add or modify the rulers in that file. When constructing a ruler, the character "<" indicates the desired left margin; the character ">" the (optional) right margin, and a "+" indicates a tab stop. All other characters in the ruler should be blanks; presence of a tab character in a ruler will invalidate it.

DISASTERS

A Disaster can occur while using SEDIT for either of the following reasons: a system or disk error occurred which SEDIT is not prepared to handle, or a fatal bug (Heaven forbid) in SEDIT is uncovered. In either circumstance, SEDIT displays the message:

```
*** Editor Failure ***  
Error xxxx @ nnnn/:hhhh
```

SEDIT attempts to complete copying the old file into <FILENAME>.TMP before giving up entirely. Generally, the edited text is preserved intact, although it is not saved in the proper file. Simply renaming files can solve this problem. The operator should check error code xxxx against standard SDOS error messages; Error 1015 (Disk Full) is the most common error encountered not caused by SEDIT. Other errors should be examined by knowledgeable people; those that cannot be easily understood should be treated as BUGs and reported to Software Dynamics, along with the information printed at the Editor failure message.