

# OS9/6809 Scred Commands

## Introduction

The OS-9 Level 2 Screen Editor (Scred) is a powerful and simple to learn screen-oriented text editor. You can use Scred to prepare text for letters and documents or text to be used by other OS-9 programs such as the assembler and high-level languages. Scred's features include:

- Adjustable screen and workspace size
- Continuously updated screen
- Cursor positioning by characters, words, and line-by-line
- Scrolling
- Cut and paste
- Change, find, and search strings
- Wild cards

## Modes of Operation

Scred has three modes of operation: Command, Edit, and Insert. The Command Mode lets you execute Scred commands that affect files or the edit buffer. Scred starts up in Command Mode. The Edit Mode lets you modify or manipulate text within the edit buffer. The Insert Mode lets you enter new text into the edit buffer.

## Starting Scred

To start Scred, type:

```
scred filename
```

If the file exists, Scred loads the file into the edit buffer, displays the beginning of the file, and enters Edit Mode.

If the file does not exist, Scred displays:

```
can't open filename  
ERROR #216
```

and enters Command Mode.

If you want to create a new file, type:

```
scred
```

This starts Scred in Command Mode, from which you can load a file or begin creating a new one by using the NEW command (see Chapter 3).

**Note:** Scred uses a special file called *termset* to describe the attributes of a particular terminal. See Chapter 2, "The Termset File," for more information on this file.

## Available Options

You can use several options on the command line when starting Scred. These options specify the terminal type, buffer size, and so on. Use the following form when starting Scred with options:

```
scred filename options
```

The available options are:

-?

Displays a list of the Scred options.

-b=*\_num\_k*

Allocates *\_num\_k* bytes of memory for Scred's working buffer. The buffer's default size is 12 kilobytes. The "=" and "kj" are optional parameters. For example, -b32 is the same as -b=32k.

-e

Configures Screeen for terminals that have embedded video attributes; that is, terminals in which the attribute start flag uses one character position.

-g

Configures Scred for special graphic-oriented terminals (terminals that do not support line feeds).

-l=*num*

Specifies the number of lines to be displayed on the terminal screen. You can also set this option in the *termset* file. See Chapter 2, "The Termset File," for more information.

-t=*term*

Specifies the terminal type. Use this option if your terminal type is different from the default terminal type as set in the *termset* file. See Chapter 2, "The Termset File," for more information.

-w=*num*

Specifies the maximum number of characters per line to be displayed on the terminal screen. You can also set this option in the *termset* file. See Chapter 2, "The Termset File," for more information.

-z=*path*

Sets the pathlist that Scred uses to find the *termset* file. See Chapter 2, "The Termset File," for more information.

**Note:** Since Scred normally checks the current window size, the -l and -w options are not often needed. If you use them, be certain you give valid values. Otherwise these options can interfere with screen formatting.

### Examples

```
scred file1 -b=32k
```

This command starts Scred with a 32k byte buffer.

```
scred file1 -l=24 -w=30
```

This command starts Scred with a screen size of 24 lines by 30 characters.

## The Termset File

To operate properly, Scred must know the type of terminal you are using. Scred finds this information in a file named Termset. The Termset is a text file containing entries

that describe a variety of terminals. The terminal types currently supported in Termset are:

- CoCo (the default for windows)
- VDG (for VDG screens)
- ABM85
- KT7
- ANSI
- ABM85H

If you are using other than the CoCo terminal, use the `-t` option and specify the terminal name when starting Scred. If your terminal type is not currently supported in the Termset file, read the rest of this chapter for instructions to add your terminal to the file. Scred looks for the Termset file in the directory `/dd/sys`, where `dd` is the default device for your system. If Scred doesn't find the file there, it looks in `/h0/sys` and then in `/d0/sys`. You can use the `-z` option of Scred to specify a different path for the Termset file.

## Modifying the Termset File

To add a new terminal type to the Termset file, you can:

- Edit the Termset file using a text editor
- Use the
- `maketerm`

supplied on the Scred distribution diskette. Because

`maketerm`

is easier to use, it is the method shown in this chapter's examples.

## The Termset File Format

The Termset file contains control code definitions for one or more types of terminals. Each text line in the file is a complete description list for a particular kind of terminal. The first line of the Termset file contains the name and control code definition for the terminal type. This is the terminal type Scred uses if you do not use the `-t` option. The form is:

```
NAME:ccc:cov:dl:dc:cs:cel:il:sav:eav:sl:sw
```

Each field represents a different control code definition. Notice that each field is separated by a colon (:). Even if the terminal cannot perform a certain function, the colon must still be present to hold the function's position.

## Termset Fields

The following list defines each field in a terminal type entry:

*NAME*

**Terminal Name**

Specifies the identification name of the terminal described in the line. Use this name with the -t option to specify the terminal type for Scred to use. You must specify the name in all uppercase, although you can specify lowercase with the -t option on Scred's command line.

*ccc*

### **Cursor Control Code**

Positions the cursor to any location on the screen. This function is required. There are two parts to the Cursor Control Code: (1) one or more *position cursor* command characters, and (2) cursor coordinates. \X and \Y (or \X\X and \Y\Y) are cursor coordinates where X and Y refer to the column number and row number, respectively. The order in which you specify the cursor coordinates is dependent on your terminal's requirements.

This information should be supplied with the hardware specifications that come with your terminal.

*cov*

### **Cursor Offset Value**

Sets the offset value for the cursor coordinates. This value, specified in hexadecimal, is always added to the cursor X and Y coordinates. Many terminals use an offset of \$20.

*d/*

### **Delete Line Control Character(s)**

Deletes the character under the cursor and shifts the remaining characters on the line to the left by one character position.

*cs*

### **Clear Screen**

Erases the entire screen, and returns the cursor to the home position.

*ce/*

### **Clear to End of Line**

Erases all characters on the line from the current cursor position to the end of the line, including the character under the cursor.

*il*

### **Insert Line**

Creates a new blank line by scrolling the current and subsequent lines down one line.

*sav*

### **Start Alternate Video**

Displays all subsequent characters in reverse video, different intensity, or any similar mode that is visibly different from the normal video mode. This code is used when highlighting text.

*eav*

### **End Alternate Video**

Displays all subsequent characters in normal video mode.

You can specify 0-4 output control characters for the following fields: Delete Line, Delete Character, Clear Screen, Clear to End of Line, Insert Line, Start Alternate Video, and End Alternate Video.

*s/*

### **Screen Length**

Specifies, in hexadecimal, the number of lines to be displayed on the terminal screen. This field is optional. If you omit this value, Scred uses 24.

*sw*

### **Screen Width**

Specifies, in hexadecimal, the number of columns to be displayed on the terminal screen. This field is optional. If you omit this value, Scred uses 80.

Screen length and screen width are optional fields. If you omit them, Scred checks the size of the current screen (or part of the screen) and uses these values. For external terminals, Scred assumes a screen size of 24 lines by 80 columns. If you do specify a length and width, Scred uses these values and does not check the size of the current screen.

### **Examples**

#### **Example 1**

Create the following Termset entry:

```
ABM85:$1b$3d\eY\eX:$20:$1bR:$1bW:$1e$1bY:$1bT:$1bE:$1bj:$1bk:$18
:$50:
```

To create the above entry, type the following at the system prompt (\$):

```
maketerm
```

The

```
maketerm
```

utility prompts you to supply a value for each field in the Termset entry. If a Termset file does not exist,

```
maketerm
```

creates it. If the file does exist,

```
maketerm
```

appends the new entry to the end of the Termset file.

**Note:** If a particular terminal does not have one of the requested features, simply press `&#0091;ENTER`] at the prompt.

Following are the prompts displayed by

```
maketerm
```

and the responses needed to create the ABM85 entry:

```
terminal name: ABM85 [ENTER]
cursor positioning sequence: $1b$3d\eY\eX [ENTER]
cursor position offset: $20 [ENTER]
delete line sequence: $1bR [ENTER]
delete character sequence: $1bW [ENTER]
clear screen: $1e$1bY [ENTER]
clear to end of line: $1bT [ENTER]
insert line: $1bE [ENTER]
alternate video: $1bj [ENTER]
restore normal video: $1bk [ENTER]
screen length: $18 [ENTER]
screen width: $50 [ENTER]
```

## Example 2

To create the following Termset entry:

```
TERM:$1bR\X\Y:$00:::$0e:::$1bj:$1bl:::
```

Type

```
maketerm
```

. The prompts and responses look like this:

```
terminal name: TERM [ENTER]
cursor positioning sequence: $1bR\X\Y [ENTER]
cursor position offset: $00 [ENTER]
delete line sequence: [ENTER]
delete character sequence: [ENTER]
clear screen: $0e [ENTER]
clear to end of line: [ENTER]
insert line: [ENTER]
alternate video: $1bj [ENTER]
restore normal video: $1bl [ENTER]
screen length: [ENTER]
screen width: [ENTER]
```

## Command Mode

The Command Mode lets you invoke commands that affect files or manipulate the entire edit buffer. Scred starts up in Command Mode if you do not specify a file on the command line. When you are in the Command Mode, Scred displays the > prompt in the lower left corner of the display screen.

Command Mode commands (except the GOTO command) are at least two characters long to distinguish them from the Edit and Insert Mode commands. You can use either the full name for the command, such as **edit**, or Scred's shortened form, **ed**. Commands that have short forms are shown as follows:

```
ed[it]
```

This means you can type either **ed** or **edit** for the EDIT command. Do not type the square brackets.

When entering commands in Command Mode, you can use the standard OS-9 control keys to backspace, delete lines and characters, and so on. Press [ENTER] after typing each command.

### Changing to the Edit Mode

There are two methods in which you can enter Edit Mode from Command Mode:

1. Edit an existing file by typing at the > prompt:
2. ol[d] filename  
If Scred can open the file, it then enters the Edit Mode.
3. If you have a file open and want to enter the Edit Mode, type:
4. ed[it]

You can also press [CTRL]E] to enter Edit Mode.

From the Edit Mode, you can change to the Command Mode by pressing [CTRL]BREAK].

## Changing to the Insert Mode

You can enter Insert Mode from Command Mode by typing:

```
in[sert]
```

Create a new file typing at the > prompt:

```
ne[w] filename
```

If Scred can create the file, it loads the file into the edit buffer and then enters the Edit Mode.

You can enter the Insert Mode from the Edit Mode by: (1) pressing [ENTER] to insert text before the cursor position, and (2) pressing the down arrow to insert a new line before the current line. You can then begin typing the new line.

**Note:** You cannot enter the Insert or Edit mode if no file exists in the edit buffer.

## Manipulating the Edit Buffer

Scred's edit buffer size is 12k bytes unless you use the -b option to specify a different value. If your file is larger than the edit buffer, Scred loads as much of the file as it can, while leaving approximately 2k free for changes and additions. With the 12k buffer size, Scred loads 10k of the file. The following commands show how to write, read, and insert files or sections of files.

## Saving Text

The WRITE command writes the contents of the edit buffer and the remainder of the input file (if any) to the output file. WRITE then closes the file and clears the edit buffer.

To write a file, type:

```
wr[ite]
```

When Scred saves a file, it creates an output file called Ed.tmp.xxx, where xxx is the process id number. If Scred can successfully create and write the entire output file, it deletes the current input file and renames the output file to the old name.

The UPDATE command writes the changes you made to the edit buffer and re-enters the Edit Mode. To update a file, type:

```
up[date]
```

The ADD command lets you insert a specified file within the text of the edit buffer. Scred inserts the file directly before the current line. To add a file before the current line, type:

```
ad[d] filename
```

**Note:** There must be enough free space in the edit buffer for the extra text. If Scred runs out of space, it terminates with the message **file too large to add** and does not load any of the file.

The MORE command lets you read in the next section of the input file. Use this command when the file you are editing is too large to entirely fit in the edit buffer. The MORE command causes Scred to write the contents of the edit buffer between the top of the buffer and the current cursor position to the output file and read the next section of the input file into the edit buffer. To read the next section of a file, type:

```
mo[re]
```

## Removing Text

Scred lets you delete specified lines of text from the edit buffer or delete the entire buffer.

The DELETE command lets you delete specified lines from the edit buffer. To delete lines, type:

```
de[lete] start-line end-line
```

This command deletes text from *start-line* to *end-line*, inclusive.

The ABORT command erases the entire contents of the edit buffer and closes the file.

To erase and close a file, type:

```
ab[ort]
```

The CLEAR command also erases the entire contents of the edit buffer but the file remains open. To clear the edit buffer, type:

```
cl[ear]
```

## Searching for Strings

The FIND command prompts you to enter a *search mask* and then searches for that string. If Scred finds the string, it positions the cursor at the beginning of the first occurrence of the string and then enters Edit Mode. To find a string, type:

```
fi[nd]
```

The SEARCH command prompts you to enter a search mask and then searches for that string. In addition, SEARCH lets you search for that string between specified lines instead of through the entire file. If Scred finds the string, it displays the lines, including the line number, in which the string was found. To search for a string, type:

```
se[arch] start-line end-line
```

This command searches for the string beginning at *start-line* through *end-line*, inclusive. If you omit *start-line* and *end-line*, Scred searches the entire edit buffer.

**Note:** The SEARCH and FIND commands accept a *match first word only* character. By placing a ^ as the first character in the search string, Scred finds a match only if it finds the string at the beginning of the line.

## Changing Strings

The CHANGE command replaces all occurrences of a string within the specified range of lines or over the entire edit buffer. To use the CHANGE command, type:

```
ch[ange] start-line end-line
```

If you omit *start-line* and *end-line*, Scred searches the entire edit buffer.

When you invoke the CHANGE command, Scred prompts you to enter a **Search mask**:. Enter the string you want to change. Scred then prompts you to enter a **Change mask**:. Enter the new string.

If Scred finds the search string, it displays the lines, including the line numbers, in which the changes occurred.

**Note:** The CHANGE command accepts a *match first word only* character. By placing a ^ as the first character in the search string, Scred finds a match only if it finds the string at the beginning of the line.



## Using Wild Cards

When entering the search string for the FIND, SEARCH, and CHANGE commands, you can optionally use the wild card character "?". The wild card character matches any one character in the specified location. For example:

```
m?????
```

Scrd matches all strings that begin with the letter "m" and are followed by five characters. Sample strings that would match are: "millio", "mister", and "my dog".

```
??_??
```

In this example, Scrd matches all five character strings with the underscore character (\_) in the third character position. Some sample strings that match this string are: "SS\_ID", "WA\_86", and "\_dj".

**Note:** Scrd matches spaces between words when searching for a wild card string.

## Miscellaneous Commands

The GOTO command positions the cursor on a specified line and enters Edit Mode. To position the cursor, type:

```
go[to] line-number
```

The CHD command changes the current working directory to the specified directory. You can specify either a relative or absolute path to the new directory. To change directories, type:

```
chd pathname
```

The DIR command displays the directory listing for the current directory. To obtain a listing, type:

```
dir
```

Scrd can handle files with tabs in them. However, tabs are not a function of Scrd. The TABS command lets you set tab stops at each *n* characters. To set the tab stops, type:

```
ta[bs] n
```

Scrd sets tabs at every four characters by default.

Another feature of Scrd is auto-indent. If you enter an indented line, Scrd automatically aligns the next line with it.

The NOTAB command turns off the auto-indent function. To disable the auto-indent feature, type:

```
not[ab]
```

The AUTO INDENT command turns the feature back on. To enable the auto-indent feature, type:

```
au[to indent]
```

The \$ command lets you execute a shell command line from within Scrd. To execute an OS-9 command, type:

```
$command-line
```

For example, to list the contents of a file, type:

```
$list filename
```

When you use the SHELL command (\$), OS-9 starts a new shell (if your computer has enough free memory). In this way it can process several OS-9 commands. To return to the Scrd > prompt, press [CTRL][BREAK].

## Exiting Scred

The EXIT command ends the current editing session. If a file exists, Scred saves the file to disk and returns to the OS-9 system. To exit Scred, type:

```
ex[it]
```

## Edit Mode

The Edit Mode lets you control and modify text in the edit buffer and on the screen display. You can enter Edit Mode from Command Mode by typing **ed** **&#0091;ENTER]** or by pressing **&#0091;CTRL]&#0091;E]**. You can enter Edit Mode from Insert Mode by pressing **&#0091;CTRL]&#0091;BREAK]**. When you enter Edit Mode, Scred displays the text of the file being edited.

Commands in this chapter appear in uppercase as they appear on your keyboard. Unless specifically noted, you do not have to press **&#0091;SHIFT]** to invoke the commands.

## Getting Help

You can display help information at any time while in Edit Mode. To do so, press **?**. Scred displays a list of commands at the top of the screen. The commands are divided into four groups:

- Cursor control keys
- Edit buffer controls
- Cut and Paste commands
- Miscellaneous commands

Press the spacebar to review the display for each group. Press **q** to exit the help function.

## Controlling the Cursor

The following table lists the keys Scred uses to position the cursor. When looking at this table, notice that the location of each key on the keyboard is related to the movement it performs.

Key Action

I

moves the cursor up one line

, (comma)

moves the cursor down one line

J

moves the cursor left one character

L

moves the cursor right one character

K

moves the cursor alternately to the beginning or end of the current line

H

moves the cursor one word to the left

;  
moves the cursor one word to the right

## Scrolling the Screen

Scrd uses four keys to scroll the screen. The table below lists the keys and their descriptions. As before, notice the location of the keys on your keyboard.

Key Action

U  
scrolls the screen up continuously

M  
scrolls the screen down continuously

O  
scrolls the screen up

.  
scrolls the screen down

The continuous scroll feature is useful when you want to quickly scan through a file. Use the space bar to pause and restart scrolling. Type any other character to terminate scrolling.

When scrolling down one screenful, the line at the bottom of the screen scrolls to the top of the screen. When scrolling up one screenful, the line at the top of the screen scrolls to the bottom of the screen.

## Moving to a Specific Line

The GOTO command moves the cursor to the specified line within the edit buffer. To move the cursor to a specific line, press **G**. Scrd prompts you to enter the line number with the prompt **goto:**. Enter the line number to which you want to move the cursor. Scrd positions the cursor at the beginning of the specified line and positions that line on the third line of the screen.

Line 1 is the first line of the edit buffer. Any number higher than the last line number causes the last line to be selected.

## Finding a String

The FIND command searches for a specified string and positions the cursor on the first character of that string. To invoke FIND, press **F**. Scrd prompts you to enter a **Search mask:**. Type the string you want to find. If Scrd finds the string, it positions the cursor on the first character of the string and positions the line in which the string occurred on the third line of the screen. If Scrd cannot find the string, it displays the message, **find: no match**.

To find another occurrence of the same string, press **F** and press **&#0091;ENTER]** for the search mask. Scrd moves the cursor to the next occurrence of the previously entered string.

## Replacing Strings

The REPLACE command lets you substitute one string for another. To replace a string, press **R &#0091;ENTER]** and Scrd prompts you to enter a **Search string:**. Enter the

string you want to replace. Scrd then prompts you to enter the **Change string:**. Enter the new string.

To replace the next occurrence of the search string with the same string, press **R** and press **&#0091;ENTER]** for both prompts.

## Deleting Text

Scrd offers a variety of ways to delete text. You can delete characters, words, and lines. The following table summarizes the key commands and their definitions.

Key Action

backspace

deletes the character to the left of the cursor

**&#0091;CTRL]&#0091;;]**

dletes the character under the cursor

**&#0091;CTRL]&#0091;A]**

deletes one word to the left of the cursor

**&#0091;CTRL]&#0091;D]**

deletes one word to the right of the cursor

**&#0091;CTRL]&#0091;C]**

deletes from the current cursor position to the end of the line

**&#0091;CTRL]&#0091;Z]**

deletes from the current cursor position to the beginning of the line

**&#0091;CTRL]&#0091;X]**

deletes the current line

**Note:** If you accidentally delete text, you can recover by pressing **&#0091;CTRL]&#0091;F]**. The **&#0091;CTRL]&#0091;F]** command restores the current line to its original state.

## Inserting or Replacing a Single Character

Scrd easily lets you insert one character or substitute one character with another without having to enter Insert Mode.

The REPLACE CHARACTER command replaces the character under the cursor. To replace a character, type **X\_character\_**. For example, typing **Xz** replaces the character under the cursor with a "z."

The INSERT CHARACTER command inserts a character in front of the character under the cursor. To insert a character, type **B\_character\_**. For example, typing **Ba** inserts an "a" in front of the character under the cursor.

## Cutting and Pasting

Scrd's *cut and paste* feature lets you move a block of text and insert it at another location. Scrd lets you move, delete, or duplicate blocks of text.

Before you move a block of text, you must mark the beginning point of the block. The SET command marks the starting line. To mark a line, move the cursor to the first line of the block of text you want to move, and press **S**. To mark in the middle of a line, first break the line into two lines, and then mark it. Scrd displays the marked line in reverse video if your terminal has the capability.

Next, move the cursor to the last line of the text block you want to move. Use the CUT command to remove the text from the edit buffer. Scred places the text in its *paste* buffer.

You can add more text to the past buffer by using the APPEND command. To use the APPEND command, mark the beginning of the text block using SET, and move the cursor to the end of the block. Press **A**, and Scred appends the text block to the text already in the paste buffer.

Use the PASTE command to return the contents of the past buffer to the edit buffer. Scred pastes text on the line above the current line. Therefore, to paste the text, position the cursor one line below the line on which you want the text inserted, and press **P**.

You can also duplicate text by using the NON-DESTRUCTIVE CUT command. To do so, mark the beginning of the text block using the SET command and move the cursor to the last line of the text to be duplicated. Press **N** and Scred copies the text block into the paste buffer. The text in the edit buffer is untouched.

Scred also offers a NON-DESTRUCTIVE APPEND command. Mark the beginning of the text block (SET), and move the cursor to the last line of the text to duplicate. Press **V**, and Scred appends a copy of the text to the end of the paste buffer. The text in the edit buffer is untouched.

The ERASE command clears the paste buffer and returns its memory. Press **E** to erase. Scred also lets you write sections of text to a file using the WRITE command. To do so, mark the beginning of the block (SET), and move the cursor to the last line of the block. Press **P**, and Scred prompts you to enter an output filename. If you invoke the WRITE command without marking a text block, Scred writes the paste buffer to the output file. If Scred cannot create the file, it issues an error message.

## Editing Lines

Scred allows you to use lines of up to 256 characters in length. However, because Scred does not wrap lines, you can see only a portion of the line if it is longer than the width of your screen. Scred offers an easy method of breaking and joining lines.

The BREAK command splits the line at the current cursor position. Scred inserts the break before the cursor. To break a line, press **&#0091;CTRL]&#0091;B]**.

The JOIN command joins the current line with the one above. To join two lines, press **&#0091;CTRL]&#0091;P]**.

## Displaying the Status Line

The status line displays the line number, column number, amount of free space in the edit buffer, paste buffer size, current filename, and the current mode (Command, Edit, or Insert). To display the status line, press **&#0091;CTRL]&#0091;G]**. Press the space bar to remove the status line from the screen.

The following sample status line shows the current cursor position to be Line 50, Column 0. There is more than 14k bytes free in the edit buffer and 51 bytes of text stored in the paste buffer. The filename is *Example*, and Scred is in the Edit Mode.

```
L50 C:0 MB:14526 CB:51 F:Example edit:
```

## Insert Mode

The Insert Mode lets you enter new text into the edit buffer. To enter the Insert Mode from Command Mode, type **[ENTER]**. To enter the Insert Mode from the Edit Mode, press **[ENTER]** or **[UP ARROW]**.

Scred inserts the new text before the current cursor position and stores it exactly as you type it. You can enter control characters. To enter control characters, press **[CTRL][V]** followed by the character you wish to enter. For example, to enter a Control-L into the edit buffer, press **[CTRL][V]**, then **[L]**.

## Quick Reference

The following tables provide a quick reference to the commands for the Command, Edit, and Insert Modes.

### Command Mode

Command Description

**ab**;

Cancels all changes made to the current file, erases the entire edit buffer, and closes the current file.

**ad**; *filename*

Adds the text of the specified file to the edit buffer, starting at the line above the current cursor position.

**au**;to indent

Tells Scred to automatically indent the next line after a carriage return in the previous line begun with a tab or space(s). Scred indents the new line to the same column position as the previous line. Scred starts up in auto-indent mode.

**ch**; *string*; *start-line*; *end-line*;

Replaces all occurrences of a string within the specified range of lines. Omitting a range value causes Scred to search the entire edit buffer.

**chd** *pathname*

Changes the current working directory.

**cl**;

Erases all text in the edit buffer. Scred does not close the file.

**de**; *start-line*; *end-line*;

Erases the specified range of lines from the edit buffer.

**dir**

Displays the directory listing for the current working directory.

**ed**;it

Enters Edit Mode. You can also use **[CTRL][E]**;

**ex**;it

Writes the edit buffer to the output file and exits Scred.

fi&#91;nd&#93;

Searches for the first occurrence of a string. Enters the Edit Mode.

g&#91;oto&#93; *line*

Moves the cursor to the specified line number. Enters the Edit Mode.

in&#91;sert&#93;

Enters Insert Mode.

mo&#91;re&#93;

Saves the text in the edit buffer to the output file and reads in the next section of the input file.

ne&#91;w&#93; *filename*

Creates a new file with the specified filename and enters Insert Mode.

not&#91;ab&#93;

Turns off the auto-indent mode.

ol&#91;d&#93;

Clears the edit buffer, opens an existing file, and enters Edit Mode.

se&#91;arch&#93; &#91;*start-line&#91;end-line&#93;&#93;*

Searches for a string within the specified lines. If you omit the line numbers, Scred searches the entire edit buffer.

ta&#91;ns&#93; *n*

Sets the tab stops to every *n* characters.

up&#91;date&#93;

Writes changes to the output file and re-enters Edit Mode.

wr&#91;ite&#93;

Writes the contents of the edit buffer and the remainder of the input file, if any, to the output file.

\$ &#91;*command&#93;*

Executes a shell command line.

**&#91;CTRL&#93;&#91;G&#93;**

Displays the status line.

## Edit Mode

### Cursor Movement Commands

Command Description

I

Moves the cursor up one line.

, (comma)

Moves the cursor down one line.

J

Moves the cursor left one character.

H

Moves the cursor left one word.

L

Moves the cursor right one character.

;

Moves the cursor right one word.

K

Moves the cursor to the beginning or end of the line.

R

Replaces a string.

U

Scrolls the text up. Press the space bar to stop and start. Press any other key to abandon.

M

Scrolls the text down. Press the space bar to stop and start. Press any other key to abandon.

O

Scrolls text up one page.

.

Scrolls text down one page.

G

Moves the cursor to the specified line.

F

Finds the first occurrence of a string.

*X char*

Replaces the character under the cursor with the specified character.

*B char*

Inserts the specified character before the cursor and advances the cursor.

**&#91;Backspace&#93;**

Deletes the character to the left of the cursor.

**&#91;CTRL&#93;&#91;;&#93;**

Deletes the character under the cursor.

**&#91;ENTER&#93;**

Enters Insert Mode.

**&#91;Down Arrow&#93;**

Moves the text in the edit buffer down one line and enters Insert Mode with the cursor on the new line.

**&#91;CTRL&#93;&#91;BREAK&#93;**

Returns to Command Mode.

?

Displays help information.

**&#91;CTRL&#93;&#91;A&#93;**

Erases one word to the left of the cursor.

**&#91;CTRL&#93;&#91;D&#93;**

Erases one word to the right of the cursor.

**&#91;CTRL&#93;&#91;F&#93;**

Cancels any changes made to the current line.

**&#91;CTRL&#93;&#91;C&#93;**

Erases text from the cursor to the end of the line.

**&#91;CTRL&#93;&#91;Z&#93;**

Erases text from the cursor to the beginning of the line.



**&#91;CTRL&#93;&#91;X&#93;**

Erases the entire line.

**&#91;CTRL&#93;&#91;B&#93;**

Splits the current line into two lines at the cursor position.

**&#91;CTRL&#93;&#91;P&#93;**

Joins the current line with the line above.

**&#91;CTRL&#93;&#91;G&#93;**

Displays the status line.

## Cut and Paste Commands

Command Description

S

**Set.** Marks the first line of a text block to be deleted, duplicated, or moved. If the starting mark is already set, **S** removes the mark.

C

**Cut.** Deletes the selected block of text from the edit buffer and stores it in the paste buffer.

N

**Non-destructive Cut.** Places the selected block of text in the paste buffer without altering the edit buffer.

P

**Paste.** Inserts the contents of the paste buffer at the line above the cursor.

A

**Append.** Deletes the specified block of text from the edit buffer and adds it to the end of the paste buffer.

V

**Non-destructive Append.** Appends the specified block of text to the paste buffer without altering the edit buffer.

E

**Erase.** Erases the content of the paste buffer and releases its memory space to the edit buffer.

W

**Write.** Writes the specified lines to the output file. If no lines are marked, Scrd writes the paste buffer to the output file.

## Insert Mode

Command Description

**&#0091;CTRL&#0091;V] char**

Inserts the specified control character into the edit buffer.

**&#0091;CTRL&#0091;BREAK]**

Returns to Edit Mode.