Cromemco Screen Editor Instruction Manual

Five Dollars

Cromemco
Screen Editor

Cromemco
incorporated
Tomorrow's Computers Today
280 BERNARDO AVE. MOUNTAIN VIEW, CA 94043

Part No. 023-0081

August 1979

This manual was produced on a Cromemco System Three computer using the SCREEN Editor described herein. The edited text was formatted using the Cromemco Word Processing System Formatter. Final camera-ready copy was printed on a Cromemco 3355 printer.

TABLE OF CONTENTS

CHAPTER 1
Introduction

1.1 Description and File Organization

The most common way to enter text or programs into a computer is by means of an editor. The SCREEN editor described in this manual is one of the easiest to use and most powerful editors available for this function. SCREEN was designed specifically to be used with Cromemco computer systems.

The Cromemco SCREEN editor displays an entire screen of information during the editing process. A cursor in the display can be readily moved around the screen to add, delete, or change information. Special features of Cromemco CRT terminals such as cursor positioning, blinking fields, and programmable function keys are used to simplify operation to the fullest.

Cromemco software packages are provided with both this screen editor as well as a string text editor. These editors may be used interchangeably in that a file created with one editor may be edited with the other editor. The string editor is necessary for using macro instructions or conditional editing functions. The string editor must also be used if the Cromemco computer is used with other than a Cromemco CRT terminal. But for the great majority of applications the screen editor is the best choice. It is easy to use and has many features not normally found in an editor of this type.

One important feature of the SCREEN editor is that it prompts the user automatically. This is done by using the top line of the screen display as a "menu" of command choices. By referring to this menu there is no need to refer back to this instruction manual during the routine operation of the editor. Another feature of the editor is that the user is politely notified by a beeping tone produced by the terminal if an illegal command has been entered. Still another feature of SCREEN is that the disk drive motors of the computer are automatically turned off during user editing (this applies to the floppy disk drive motors in most Cromemco computers). The advantages of turning off the motors during editing are to prevent media and drive wear while providing a quieter atmosphere for the user. The motors are automatically turned on again when required.

1.2 Hardware Requirements

The SCREEN editor operates as a program under the control of the Cromemco Disk Operating System (CDOS). SCREEN requires approximately 9K of memory in the CDOS user space. CDOS itself requires approximately 1ØK. Although not absolutely necessary, a 32K system is probably the minimum acceptable for most editing.

SCREEN is supplied by Cromemco with version Ø1.Ø7 or higher of CDOS. It is not compatible with version ØØ.2Ø of CDOS.

SCREEN is licensed by Cromemco exclusively for use on Cromemco computer systems. Cursor positioning and other functions are designed specifically for use with the Cromemco CRT terminal. Cromemco cannot assure proper operation of this software package unless it is used with Cromemco computer products.

1.3 Getting Started

The SCREEN editor makes use of two files: an INPUT FILE, and an OUTPUT FILE. The input file resides on a system disk and is the file to be edited. After editing, the resultant file is called the output file. When the output file is stored on disk, it takes the name of the original input file, while the original input file is renamed as a BACKUP FILE. This means that even if you did make some error in the editing process, the original input file is still saved on the disk with the three letter extension, BAK. Of course, when you are using SCREEN to create a new file, there is no input file and therefore no backup file.

The SCREEN editor is contained in a file named SCREEN.COM on the system disk. You can call SCREEN by typing the following in response to the CDOS prompt:

```
     SCREEN [X:]filename.ext
```

where filename.ext, if an existing file, is the name of the input file. If filename.ext does not already exist, it is the name of the new file to be created. If filename.ext is not on the current drive, X is an identifier to specify the proper disk drive (e.g. A, B, C, etc.). If filename.ext is an input file residing on the current drive, then the drive identifier need not be used.

If after editing you wish the output file to be stored on a disk drive other than the current drive, this may be accomplished by calling SCREEN in the following way:

```
     SCREEN [X:]file1.ext [Y:]file2.ext
```

Here the input file is taken from drive X (A, B, C, etc.) and the output file is stored on drive Y.

Now you are ready to begin. The best way to start using SCREEN is by following an example. The next chapter takes you step-by-step through such an example that is specially designed to illustrate how to use SCREEN.

CHAPTER 2
Using the SCREEN Editor

2.1 Creating a File

To start, call the program SCREEN from CDOS. On the same line enter the name of the file to be created. The SCREEN Editor will respond with its sign-on and then will display the command selection menu at the top of the screen. This is what it will look like (all information typed by the user is underlined):

```
A.SCREEN BUS.TXT <CR>

 CROMEMCO SCREEN Editor version xx.yy       new file

 >Edit: @ Copy Delete Exit Find Insert ... Xchng Zap
```

The SCREEN Editor has signed-on and informed the user that it could not find a file called BUS.TXT so it created a new file with that name. BUS.TXT will be the name of the output file when you are finished using the SCREEN Editor.

Note that the file name is written in two parts. The three characters following the period are called the file extension. For word processing applications the extension TXT is used to indicate a text file. When writing software program source, the following conventions are recommended:

BASIC	.BAS
COBOL	.COB
FORTRAN	.FOR
RATFOR	.RFR
Z80 assembler	.Z80

Next, enter some text into the text buffer using the I (Insert) command. We will intentionally introduce some mistakes in order to illustrate some features of the editor.

```
     >Edit: @ Copy Delete Exit Find Insert ... Xchng Zap

    I

    >Insert:  <text...> <esc>

    THE S-1ØØ MICROCONPUTERBUS CONSISTS OF A <CR>
    BANK OF 1ØØØ-CONTACT CONNECTORS WIRED IN <CR>
    PARALLEL ONA COMMON MOTHER BOARD. <CR>
    <esc>▒
```

The Insert command is initiated by typing the letter I. Notice that all command characters may be entered in either upper or lower-case. Any text (a program, a letter, a series of numbers, etc.) may then be entered. Insert is terminated by pressing the ESCape key. The SCREEN Editor then redisplays the menu.

After the completion of the insert sequence, the cursor is left after the last character which was inserted. In this case the cursor is at the end of the text buffer or just after the <CR> of the last line.

Note that if you find errors while still in the INSERT mode, you may remove them by using the DELete key, CTRL-A, or CTRL-H. The left arrow key of the Cromemco 31Ø2 terminal may also be used for this purpose.

Suppose the user wanted to take a break now and turn off the computer. If the computer was turned off before writing the newly INSERTed text to the output file, all would be lost, because the computer memory (which is where the text is stored) "forgets" when the power to it is turned off. To avoid this situation, the Exit command will cause the contents of the text buffer to be written to the output file and control to be returned to CDOS:

```
>Edit: @ Copy Delete Exit Find Insert ... Xchng Zap
```

E

```
>Exit: Quit Update <esc>
         Quit   - exit without updating the file
         Update - update the file and exit
         <exc> - return to editor
```

U

A.

The computer can safely be turned off after the disk has been removed from the drive because the text file has been stored on the disk.

2.2 Editing a File

Now to retrieve our newly created file and correct the mistakes. Call the SCREEN Editor with the name of the file, as was done when it was created. SCREEN will find the file we created last time and treat it as the input file. The SCREEN Editor signals its readiness to accept commands by displaying the commands menu at the top of the screen. Commands can be entered in upper or lower-case; for example, the Substitute command can be called with S or s. Now we will correct the mistakes.

```
>Edit: @ Copy Delete Exit Find Insert ... Xchng Zap
```

The first error is an N instead of an M in MICROCOMPUTER. Position the cursor over the N in MICROCONPUTERBUS by moving left with CTRL-A (hold down the CTRL key and press A) to the R in BOARD and up with CTRL-W to N. Type X to call the Xchng command and overwrite the N with an M. Press the ESCape key to exit the Xchng command.

```
>Edit: @ Copy Delete Exit Find Insert ... Xchng Zap


THE S-100 MICROCONPUTERBUS CONSISTS OF A BANK
OF 1000-CONTACT CONNECTORS WIRED IN PARALLEL
ONA COMMON MOTHER BOARD.

>Xchng:  <text> <esc>

THE S-100 MICROCOMPUTERBUS CONSISTS OF A BANK
OF 1000-CONTACT CONNECTORS WIRED IN PARALLEL
ONA COMMON MOTHER BOARD.
```

The second error is that MICROCOMPUTER and BUS are two words and should be separated by a space. Move the cursor right with CTRL-D to the B in MICROCOMPUTERBUS. Call the Insert command by typing I. Enter a space by pressing the space bar. Leave Insert by pressing the ESCape key and return to the command selection menu.

```
>Insert:  <text...> <esc>
```

```
THE S-1ØØ MICROCOMPUTER BUS CONSISTS OF A
BANK OF 1ØØØ-CONTACT CONNECTORS WIRED IN
PARALLEL ONA COMMON MOTHER BOARD.
```

The next error is that the 1ØØØ should have been 1ØØ. To correct this, first position the cursor over the last Ø in 1ØØØ by pressing carriage RETURN to move to the beginning of the second line and move to the right with CTRL-D. Call the Delete command by typing D. Eliminate the Ø by pressing the space bar once. Leave Delete by pressing the ESCape key.

```
THE S-1ØØ MICROCOMPUTER BUS CONSISTS OF A
BANK OF 1ØØØ-CONTACT CONNECTORS WIRED IN
PARALLEL ONA COMMON MOTHER BOARD.
```

```
>Delete:  <esc>
```

```
THE S-1ØØ MICROCOMPUTER BUS CONSISTS OF A
BANK OF 1ØØ-CONTACT CONNECTORS WIRED IN
PARALLEL ONA COMMON MOTHER BOARD.
```

Next ONA will be replaced by ON A using the Substitute command. Call Substitute by typing S. The computer then queries the user for the characters to be replaced (string1) and for the new characters (string2). In response to this query type "ONA" for string1. After the computer inserts a comma type "ON A" for string2. The substitution will be made automatically.

```
 >   1 Subs ... <string1> <string2> "ONA", "ON A"
```

```
THE S-1ØØ MICROCOMPUTER BUS CONSISTS OF A
BANK OF 1ØØ-CONTACT CONNECTORS WIRED IN
PARALLEL ON A COMMON MOTHER BOARD.
```

To add another sentence to follow the one corrected it is necessary to move the cursor to the end of the existing text so the new text will follow it. Type J to call the Jump command and enter E to designate the end of the file.

```
>Jump:  Begin End <esc> E
```

Then Insert the new text and display it to check for errors.

```
I

>Insert:  <text...> <esc>

THE 100 LINES OF THE BUS CARRY ADDRESS, <CR>
DATA, AND CONTROL SIGNAL INFORMATION. <CR>
<esc>
```

The edited text file would appear as follows:

```
 THE S-100 MICROCOMPUTER BUS CONSISTS OF A
 BANK OF 100-CONTACT CONNECTORS WIRED IN
 PARALLEL ON A COMMON MOTHER BOARD.
 THE 100 LINES OF THE BUS CARRY ADDRESS,
 DATA, AND CONTROL SIGNAL INFORMATION.
```

Finally, we will Exit from the SCREEN Editor leaving a backup file, BUS.BAK, and an edited file, BUS.TXT, on the disk.

```
E

 >Exit: Quit Update <esc>
          Quit   - exit without updating the file
          Update - update the file and exit
          <exc> - return to editor


U

A.
```

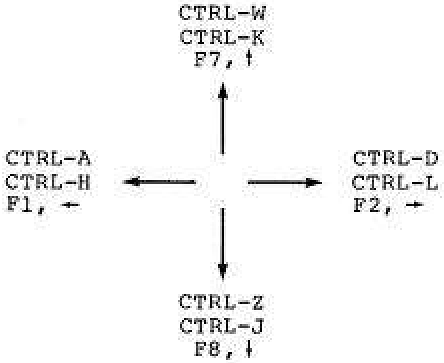
2.3 Conclusion

Having gone through this step-by-step example you should now be able to use SCREEN as a powerful editing tool with your computer. The following chapters describe SCREEN commands in more detail and describe additional features of the commands already discussed.

CHAPTER 3
Cursor Positioning Instructions

3.1 Cursor movement via CTRL characters and F# keys

The cursor may be positioned with control keys and the function keys of a Cromemco CRT terminal. This method of cursor positioning may be used both at the command selection level and without the repeat factor in the Xchng command.



CTRL-W
CTRL-K
F7, ↑

CTRL-A
CTRL-H
F1, ←

CTRL-D
CTRL-L
F2, →

CTRL-Z
CTRL-J
F8, ↓

```
[n] {CTRL-{H or A}, F1 or ← } moves left n spaces
[n] {CTRL-{L or D}, F2 or → } moves right n spaces
[n] {CTRL-{K or W}, F7 or ↑ } moves up n lines
[n] {CTRL-{J or Z}, F8 or ↓ } moves down n lines
```

where:

n	is an optional repeat factor
{}	indicate a choice from the enclosed elements.
CTRL-	is the CTRL key
F1,...	are the function keys of a Cromemco CRT terminal.
←,...	are the arrow keys of the Cromemco 3102 terminal.

In the course of up and down movement a position may be reached that is to the right of the RETURN character. When this happens the cursor will move left to the RETURN character but will return to the original columnar position if another line is encountered that reaches as far to the right as the cursor was on the original line.

The control keys will repeat their function if held down. Function keys will not repeat if held down.

3.2 Rubouts

In the Insert, Find, Substitute, Read, and Write commands, text may be rubbed out using the DELete, underscore, or any of the "move left" control keys.

3.3 Cursor Direction

Cursor direction is determined by the < or > flag in the upper left corner of the screen. The cursor direction is set to forward by typing

```
> or . or +
```

The cursor direction is set to reverse by typing

```
< or , or -
```

Example:

```
>Edit: Copy Delete Exit Find Insert Jump Move ...
```

To change the cursor direction to reverse, enter < or , or -.

```
<Edit: Copy Delete Exit Find Insert Jump Move ...
```

To change the direction to forward, enter > or . or +.

```
>Edit: Copy Delete Exit Find Insert Jump Move ...
```

3.4 Cursor movement via space bar, TAB, and RETURN

The cursor may also be positioned by using the space bar, TAB key (or CTRL-I), and the RETURN key. These modes of positioning the cursor may all be done at the command selection level.

Mode: space bar

Format: [n]S

where:

n is an optional repeat factor.

S is an entry of a space with the space bar.

In the forward mode, after a RETURN character is encountered, the cursor will move to the beginning of the next line. In the reverse mode, after the first character of a line is encountered the cursor will move to the unprinted carriage RETURN character of the next line in that direction.

The cursor may be spaced forwards in the Delete, Insert, Tab, and Xchng commands.

The cursor may be spaced backwards in the Delete and Insert commands.

Mode: tab

Format: [n]T

where:

n is an optional repeat factor.

T is an entry of a tab with CTRL-I or the TAB key.

The directional comments of space bar apply here also.

The cursor may be TABed forwards in the Delete and Insert commands.

The cursor may be TABed backwards in the Delete and Insert commands.

Mode: carriage RETURN

Format: [n]<CR>

where:

n is an optional repeat factor.

<CR> is a carriage RETURN.

The carriage RETURN will move the cursor to the nearest beginning of a line in the selected direction.

The cursor may be carriage RETURNed forwards in the Delete and Xchng commands.

The cursor may be carriage RETURNed backwards in the Delete command.

CHAPTER 4
Command Instructions

instruction: Copy

format: Cm1,m2

where:

C — is the only character necessary to call the Copy command.

m1 — is the marker that designates the beginning of the text to be copied. Refer to the Set Marker command for further information.

m2 — is the marker that designates the end of the text to be copied.

Copy is used for duplicating part or all of a file into or onto itself.

Steps:

1. Set a marker at the beginning of the portion of the file to be duplicated.

2. Set a marker at the character following the end of the portion of the file to be duplicated.

3. Position the cursor at the character following the location to which the selected text is to be copied.

4. Press C to call the Copy command; the computer will respond:

   ```
   Copy:  Begin End [1 2 3 4 5 6 7 8] <esc>
   ```

   where only the numbered markers that the user has set will appear.

5. Enter the beginning marker and the ending marker. After the second marker has been entered, the text will be copied.

Notes:

If the first marker is in the middle of a line when copied, that line will be left justified.

Example:

```
Numeric variables may be assigned numeric values.
The range and accuracy of these depends
on their type.
```

To copy the word "variables" to a position before the word "depends", first set a marker at the beginning of "variables" and at the beginning of the following word. Position the cursor at the beginning of the word "depends". Call the Copy command and enter the two markers used to delimit "variables".

```
Numeric variables may be assigned numeric values.
The range and accuracy of these variables depends
on their type.
```

DELETE

instruction: Delete

format: D

where:

D — is the only character necessary to call the Delete command.

Delete will eliminate text from a file.

Steps:

1. Position the cursor at one end of the text to be deleted.

2. Verify that the cursor direction is toward the other end of the text to be deleted. If it is not, change the cursor direction.

3. Press D to call the Delete command; the computer will respond:

   Delete: <esc>

4. Delete the unwanted text with the space bar, TAB key, RETURN key, or CTRL keys. If text was deleted that was desired, move the cursor back over the deleted text to retrieve it. This retrieval will not work if you leave Delete and get back in.

5. When the unwanted text has been deleted from the screen, press <esc> to implement these changes.

Notes:

The Delete command is confined to working on the current page.

Example:

> Cromemco 16K Extended BASIC not only generates
> English-language error messages, but it also
> examines each statement as it is entered and prints
> error messages immediately.

To delete the word "Extended" from the first line, set the cursor at the first letter of the word and call the Delete command by pressing D. With the space bar, space over to the first character of the next word and press <esc>.

> Cromemco 16K BASIC not only generates
> English-language error messages, but it also
> examines each statement as it is entered and prints
> error messages immediately.

EXIT

instruction: Exit

format: E{U,Q}

where:

E is the only necessary character to call the Exit command.

U updates the file and exit SCREEN.

Q exits SCREEN without updating the file.

Exit will cause control to return to CDOS with the option of either updating the file or not updating the file.

Steps:

1. Press E to call the Exit command; the computer will respond:

```
Exit:  Quit Update <esc>
         Quit -exit without updating file
         Update -update file and exit
         <esc> -return to editor
```

2. Press Q to leave the editor without updating the file and return to CDOS. No BAK file is created if Q is entered.

   Press U to leave the editor, update the file, create a BAK file, and return to CDOS. If there is not enough room on your diskette for the updated file, you will receive the message:

```
disk overflow

insert blank diskette (not in unit N)

unit?
```

   where unit N is the drive with the input or temp file. It is recommended to use a blank diskette to insure there will be enough space for the remainder of the file. The two parts of the file may be concatenated using the CDOS XFER utility. Refer to the CDOS User's Manual for more information about XFER/A/V.

FIND

instruction: Find

format: [n]F/string/

where:

n — is an optional repeat factor.

F — is the only character necessary to call the Find command.

/ — is a delimiter. Any non-alphanumeric character not used in the string may be used as a delimiter.

string — is a sequence of characters.

Find will locate the beginning of a string.

Steps:

1. Position the cursor at one end of the section of text to be scanned.

2. Verify that the cursor direction is toward the other end of the text to be scanned. If it is not, change the cursor direction.

3. Enter a number to designate which occurrence of the string is to be found. If no number is entered, n will default to 1 and the first occurrence of the string will be found.

4. Press F to call the Find command; the computer will respond:

   ```
   n Find:  <string>
   ```

   where n is the nth occurrence of the string that was specified.

5. Preface the string with a delimiter. The delimiter may be any non-alphanumric character which does not occur in the string. End the string with the same delimiter as was used at the beginning.

FIND

Example:

```
100 PRINT"This is a string literal"
120 REM The spacing within the string literal
130 REM and the REMark statements will not be
140 REM changed by BASIC.
150  FOR I=1 TO 10
160  PRINT I;
170  NEXT I
180 END
```

To find the second occurrence of the word "PRINT" enter 2F. When prompted for the string, enter a delimiter, the word, then the delimiter again. In this example, /PRINT/, $PRINT$, or *PRINT* will work. Be sure that the delimiter is not an element of the string.

```
2 Find:  <string> /PRINT/
```

```
100 PRINT"This is a string literal"
120 REM The spacing within the string literal
130 REM and the REMark statements will not be
140 REM changed by BASIC.
150  FOR I=1 TO 10
160  PRINT I;
170  NEXT I
180 END
```

```
     instruction:  Insert

          format:  I<text>
```

where:

```
          I        is the only necessary character
                   to call the Insert command.
```

Insert will add text to a file. Text may be appended to the end of the file, entered before a given line, or before a given character.

Steps:

1. Position the cursor on the character that the inserted text should come before.

2. Press I to call the Insert command; the computer will respond:

   ```
   Insert:  <text...> <esc>
   ```

3. Enter the text to be added.

4. Press <esc> to exit the Insert command.

Notes:

If a line was appended to the added text, a <CR> was not entered at the end of the added text. If an error is made while in Insert, the user may backtrack and correct the error with the DELete key or CTRL-A.

Example:

To insert the word "Extended" before the word BASIC, first position the cursor on the space before the first "BASIC".

```
Real-time control applications often require
integer, 16-bit arithmetic.  Cromemco
16K▒BASIC provides this capability along
with direct memory, input/output, and machine-code
subroutine access through BASIC instructions.
```

Call the Insert command and enter " Extended". If the space before "Extended" were omitted, it would appear as "16KExtended". Press <esc> to leave the Insert command.

```
Real-time control applicatrions often require
integer, 16-bit arithmetic.  Cromemco
16K Extended▒BASIC provides this capability along
with direct memory, input/output, and machine-code
subroutine access through BASIC instructions.
```

JUMP

instruction: Jump

format: Jm

where:

J is the only character necessary to call the Jump command.

m is the marker that the cursor will Jump to.

Jump will cause the cursor to move to any user-set marker, the beginning of the file, or the end of the file.

Steps:

1. Press J to call the jump command; the computer will respond:

   ```
   Jump:  Begin End [1 2 3 4 5 6 7 8] <esc>
   ```

   where only the numbered markers that the user has set will appear.

2. Enter a marker designator. If B was pressed, the first page of the file will be displayed on the screen with the cursor at the beginning of the file. If E was selected, the last half page of the file will be displayed with the cursor positioned at the end of the file. If a user-set marker was selected, a page will be displayed with the cursor centered on the screen at the marker.

Example:

> A computer is a device which performs high-speed mathematical or logical calculations or which processes information derived from @oded data in accordance with a predetermined program.

Assuming that the marker labeled "1" is set at the "c" in "coded", jumping to that marker is accomplished by calling the Jump command and pressing 1.

LIST

instruction: List

format: Lm1m2

where:

L is the only character necessary to call the List command.

m1 is a marker.

m2 is a marker.

List will copy a section of a file to a line printer. The List command is displayed on the Other command selection menu at the top of the screen.

Steps:

1. Set markers at the beginning and the end of the section of the file to be listed to the line printer.

2. Press L to call the List command; the computer will respond:

   ```
   List:  Begin End [1 2 3 4 5 6 7 8] <esc>
   ```

   where only the numbered markers that the user has set will appear.

3. Enter the markers and the file section will be listed to the line printer.

WARNING:

The List command will pause indefinitely when used in a system without a line printer.

MOVE

instruction: Move

format: Mm1,m2

where:

M is the only character necessary to call the Move command.

m1 is the marker which designates the beginning of the text to be moved.

m2 is the marker which designates the end of the text to be moved.

Move will transfer text from one part of the file to another location in the file. Move deletes the text from the original position whereas Copy does not.

Steps:

1. Set a marker at the beginning of the portion of the file to be moved.

2. Set a marker at the character following the end of the section to be moved.

3. Position the cursor at the character following the location to which the selected text is to be transferred.

4. Enter M to call the Move command; the computer will respond:

   ```
   Move:  Begin End [1 2 3 4 5 6 7 8] <esc>
   ```

   where only the numbered markers that the user has set will appear.

5. Enter the beginning and ending markers. After the second marker has been entered, the Move function will transfer the file section.

MOVE

Example:

```
rapid, 14-digit arithmetic using the powerful,
binary-coded decimal (BCD) arithmetic instructions
One major feature of Cromemco 16K Extended BASIC is
which are unique to the Z-8Ø microprocessor.
```

The third line of the example will be moved to a location before the first line. Marker 1 is set at the beginning of the third line. Marker 2 is set at the beginning of the fourth line. The cursor is positioned at the beginning of the first line. The Move command is called and the two markers are entered.

```
Move: Begin End 1 2 <esc> 1 ,2
```

```
One major feature of Cromemco 16K Extended BASIC is
rapid, 14-digit arithmetic using the powerful,
binary-coded decimal (BCD) arithmetic instructions
which are unique to the Z-8Ø microprocessor.
```

OTHER

instruction: Other

format: O

where:

O is the only character necessary to display the Other command selection menu.

Other is a continuation of the command selection menu.

Steps:

1. Press O to call the Other command; the computer will respond:

```
>Edit:  List Other Read Verify Write
```

Notes:

When either command selection menu is displayed, all commands may be called. The original command selection menu will be restored when the next command is issued.

PAGE

instruction: Page

format: [n]P

where:

n is an optional repeat factor.

P is the only character necessary to call the Page command.

Page moves the cursor n pages through the file in the selected direction.

Steps:

1. Verify that the cursor direction is set to the desired direction. If it is not, change the cursor direction.

2. Enter a number "n" and press P to display the nth page of text in the selected direction.

```
n Page
```

READ

instruction: Read

format: Rfile-ref

where:

R is the only character necessary to call the Read command.

file-ref is the name of the file to be read.

Read will copy another file into the current file. The Read command is displayed on the Other command selection menu at the top of the screen.

Steps:

1. Position the cursor to where the new file is to be inserted.

2. Press R to call the Read command; the computer will respond:

   ```
   Read:  <filename> <esc>
   ```

   Enter the filename. Include the filename extension if one exists. Press the carriage RETURN and the file will be copied into the current file beginning at the cursor position.

SET MARKER

instruction: Set Marker

format: @n

where:

@ is the only character necessary to call the Set Marker command.

n is a number from 1 to 8 designating the marker label.

Set Marker will place reference points in a file with markers.

Steps:

1. Position the cursor.

2. Press @ to call the Set Marker command; the computer will respond:

```
Set Marker:  1 2 3 4 5 6 7 8 <esc>
```

Enter one of the eight numbers and the marker will be set. The cursor will return to the previous position in the text.

Notes:

If a file section containing markers is moved or copied, all markers in the file section will be reset to the last marker in the file section. A blinking marker number in the display above indicates the marker has been previously set by the user.

Example:

> One major feature of Cromemco 16K Extended BASIC is rapid, 14-digit arithmetic using the powerful, binary-coded decimal (BCD) arithmetic instructions which are unique to the Z-80 microprocessor.

To set a marker at the beginning of the word "BASIC" first position the cursor over the "B". For additional information refer to the section on cursor positioning.

Call the Set Marker command by pressing @.

Enter the desired marker number.

```
          Set Marker:  1 2 3 4 5 6 7 8 <esc> 1
```

A marker labeled "1" has now been set at the beginning of the word "BASIC". It is a reference point that may be referred to with the commands Jump, Copy, Move, List, Read, Write, and Zap.

SET TABS

instruction: Set Tabs

format: T

where:

T is the only character necessary to call the Set Tabs command.

Set Tabs will place reference points on a line with tabs. The Set Tabs command is displayed on the Other command selection menu at the top of the screen.

Steps:

1. To set tabs press T; the computer will respond:

   ```
   Set tabs:  <esc> <CR>
   ```

   ```
            1         2         3         4  ...
   123456789Ø123456789Ø123456789Ø123456789Ø  ...
   T▒      T       T       T       T         ...
   ```

2. Position the cursor with the space bar and enter T at the desired locations. Press <CR> to implement the updated tab settings and exit from Set Tabs. Press <esc> to exit from Set Tabs without updating the tab settings .

Example:

Call the Set Tabs command and press T.

```
               1         2         3         4  ...
      1234567890123456789Ø123456789Ø123456789Ø  ...
      T▒      T       T       T       T         ...
```

To set a tab every third character for the beginning of the line, with the space bar, space to the positions where a tab is desired and press T.

```
               1         2         3         4  ...
      1234567890123456789Ø123456789Ø123456789Ø  ...
      T  T  T  T  T  T▒       T       T         ...
```

Press <CR> to save the updated tab settings.

SUBSTITUTE

instruction: Substitute

format: [n]S[Q] /string1/,/string2/

where:

n — is an optional repeat factor.

S — is the only character necessary to call the Substitute command.

Q — is the optional Query mode.

/ — is a delimiter. Any non-alphanumeric character not used in the string may be used as delimiters.

string1 — is the original string.

string2 — is the replacement string.

Substitute will replace one or several occurrences of a string with another string.

<u>Steps</u>:

1. Position the cursor before or after the text that contains the string(s) to be replaced.

2. Verify that the cursor direction is toward the strings that are to be replaced. If it is not, change the cursor direction.

3. Enter the number of occurrences of the original string to be replaced.

4. Press S to call the Substitute command; the computer will respond:

   ```
   n Substitute:  [Query]<string1>,<string2>
   ```

   where n is number of replacements desired.

5. Press Q to enable the Query option; the computer will respond:

   ```
   n Substitute Query:  <string1>,<string2>
   ```

   where n is the number of replacements desired. The word "Query" will now be blinking. Query will allow the user to selectively Substitute strings.

6. Preface string1 with a delimiter. The delimiter should be a non-alphanumeric character that does not occur in string1. End string1 with the same delimiter. Preface string2 with a delimiter that is not a character in string2. When the delimiter is entered to end string2, the replacement(s) will occur.

7. If Query was not selected, after the two strings have been entered with appropriate delimiters the replacements will be made.

8. If Query was selected, after string1 and string2 have been entered the cursor will jump to the first occurrence of string1. The command display at the top of the screen will be:

   ```
   Substitute Query:  Yes No <esc>
   ```

   Press Y to Substitute the string. Press N to not Substitute the string. The cursor will now jump to the next occurrence of string1 and the user will again be queried.

9. Press the ESCape key to exit at any time during the command.

SUBSTITUTE

Example:

For many of today's demanding applications, more features are required than were provided in the original Dartmouth language. A language that provides advanced features and capabilities is frequently called an Extended language.

To replace the word "language" with "BASIC", first position the cursor at one end of the section containing the strings to be replaced. Be sure that the cursor direction is toward the strings that are to be replaced. Enter the number of replacements to be made. Call the Substitute command. Enter a non-alphanumeric delimiter that is not contained in the word to be replaced. Enter the original word. Terminate the original word by the same delimiter. Enter another delimiter followed by the new word terminated by the same delimiter.

```
3 Substitute: ... <esc> /language/,/BASIC/
```

For many of today's demanding applications, more features are required than were provided in the original Dartmouth BASIC. A BASIC that provides advanced features and capabilities is frequently called an Extended BASIC.

VERIFY

instruction: Verify

format: V

where:

V — is the only character necessary to call the Verify command.

Verify will redisplay the current screen contents. The Verify command is displayed on the Other command selection menu at the top of the screen.

Steps:

1. Enter V to call the Verify command; there is no response other than the redisplay of the current page.

WRITE

instruction: Write

format: Wm1m2file-ref

where:

W — is the only character necessary to call the Write command.

m1 — is the marker designating the beginning of the section of the file.

m2 — is the marker designating the end of the section of the file.

file-ref — is the name of the file to be read.

Write will transfer a section of a file to an external file. The Write command is displayed on the Other command selection menu at the top of the screen.

Steps:

1. Set the markers that designate the beginning and end of the section of the file.

2. Press W to call the Write command; the computer will respond:

   ```
   Write:  Begin End [1 2 3 4 5 6 7 8] <esc>
   ```

   where only the numbered markers that the user has set will appear.

3. Enter the markers. Begin and End may be used. A comma does not need to be entered. The computer will respond:

   ```
   Write: Begin End ... <esc > m1, m2 to  <filename >
   ```

4. Enter the filename desired. Press carriage RETURN and the file will be created.

Notes:

Any existing file with the specified name will be deleted.

XCHNG

instruction: Xchng

Format: X <text> <esc>

where:

X is the only character necessary to call the Xchng command.

Xchng overwrites text. Each character entered will replace the character at the cursor position.

Steps:

1. Position the cursor at the beginning of the text that is to be overwritten.

2. Press X to call the Xchng command; the computer will respond:

   Xchng: <text> <esc>

3. Enter new text. CTRL-A or CTRL-H may be used for backtracking to correct errors in overwriting. The RETURN key in Xchng acts as a control character and will not overwrite text with blanks.

4. Press <esc> to exit from Xchng and return to the command level.

Example:

```
Cromemco 16K Extended BASIC is designed to maximize
computatasional precision, programming power, and
speed of execution by fully utilizing the extensive
158 instruction set of the Z-80 microprocessor.
```

To correct the misspelling of "computasional", first position the cursor over the incorrect letter. Call the Xchng command and enter the correct letter. Press <esc> to leave the Xchng command.

```
Cromemco 16K Extended BASIC is designed to maximize
computational precision, programming power, and
speed of execution by fully utilizing the extensive
158 instruction set of the Z-80 microprocessor.
```

ZAP

instruction: Zap

format: Zm1,m2

where:

Z is the only character necessary to call the Zap command.

m1 is a user-set marker or one of the preset Begin or End markers.

m2 is a user-set marker or one of the preset Begin or End markers.

Zap will delete sections of a file.

Steps:

1. Set a marker at the first character of the section of the file that is to be deleted.

2. Set a marker at the character just beyond the end of the section of the file that is to be deleted.

3. Press Z to call the Zap command; the computer will respond:

   ```
   Zap:  Begin End [1 2 3 4 5 6 7 8] <esc>
   ```

   where only the numbered markers that the user has set will appear.

4. Enter the beginning and ending markers. After the second marker has been entered, the file section will be deleted.

Example:

```
All computers are based on one fundamental concept
-- the binary digit.  The binary number system uses
only the digits Ø and 1.▒
```

The second sentence of this example will be deleted. Set marker 1 at the beginning of the second sentence. Set marker 2 immediately after the end of the second sentence. Call the Zap command and enter the two markers. This would delete the entire second sentence.

```
Zap:  Begin End 1 2 <esc> 1 ,2
```

```
All computers are based on one fundamental concept
-- the binary digit.▒
```

Cursor Positioning Summary

Control	Used In Command	Direction Set	Description
[n]CTRL-A [n]CTRL-H [n]F1, ←	D,X	< or >	Moves the cursor to the left.
[n]CTRL-D [n]CTRL-L [n]F2, →	D,X	< or >	Moves the cursor to the right.
[n]CTRL-W [n]CTRL-K [n]F7, ↑	D,X	< or >	Moves the cursor up n lines.
[n]CTRL-Z [n]CTRL-J [n]F8, ↓	D,X	< or >	Moves the cursor down n lines.
[n]space	D	<	Moves the cursor to the left n spaces.
[n]space	D,T,X	>	Moves the cursor to the right n spaces.
[n]tab	D	<	Moves the cursor to the left n tabs.
[n]tab	D,X	>	Moves the cursor to the right n tabs.
[n]<CR>	D	<	Moves the cursor up n lines along the left margin.
[n]<CR>	D,X	>	Moves the cursor down n lines along the left margin.

Notes:

The repeat factor n may only be used at the command selection level and in the Delete command.The default of the repeat factor n for cursor positioning is 1.

Command abbreviations are:

D	= Delete
T	= Set Tab
X	= Xchng

All cursor positioning methods will work at the command selection level.

F1, F2, F7, and F8 are preprogrammed function keys of the Cromemco 31Ø1 terminal.

Command Summary

The following is an alphabetical list of the Screen Editor commands.

Command /Subcommand	Markers Needed	Description
@		Set Marker - Sets positional reference points in a file.
C	2	Copy - Duplicates text to another location in the file.
D		Delete - Eliminates text with the space bar, repeat factor n may only be used at the selection level.
E		Exit - Leaves SCREEN and returns to CDOS.
/U		Update - Implements the edit changes to the file.
/Q		Quit - Does not update the file.
nF		Find - Locates the nth occurrence of a string where n is a positive integer in the specified direction.
I		Insert - Adds text to a file.
J	1	Jump - Relocates the cursor to a specified marker.
L	2	List - Sends text between two markers to a line printer.
M	2	Move - Transfers text to another location in the file.

Command /Subcommand	Markers Needed	Description
O		Other - Displays additional commands on selection menu.
nP		Page - Displays the nth screen of text in the specified direction.
R		Read - Inserts an external file into the current text at the cursor position.
nS		Substitute - Replaces one string with another n times in the specified direction.
T		Tab - Resets reference points on a line.
W	2	Write - Creates an external file containing the text between two markers.
X		Xchng - Overwrites text.
Z	2	Zap - Eliminates text between markers.

Error Messages

disk overflow

A disk overflow may caused by EXITing and UPDATEing a file which would exceed the capacity of the diskette or by moving through a large file. The computer will print the message:

```
disk overflow

insert blank diskette (not in unit N)

unit?
```

where unit N is the drive with the input or temp file. It is recommended to use a blank diskette to insure there will be enough space for the remainder of the file. The user will return to CDOS after writing the last section of the file to the blank diskette. The two parts of the file may be concatenated using the CDOS XFER utility. Refer to the CDOS User's Manual for more information about XFER/A/V.

disk read error

Receiving a disk read error generally indicates that the file being read in has problems. If this error occurs during the execution ofthe READ command, the user will stay in SCREEN and return to the command selection level.

disk write error

A disk write error will occur when trying to WRITE to a file which would exceed the capacity of the disk. If this error occurs during the WRITE command, the user will stay in SCREEN and return to the command selection level.

file not found

This error will be the result of trying to READ a non-existent file, a file with a misspelled filename, or a filename with a missing extension. The user will exit the READ command and return to the command selection level.

Glossary

definition {} - Indicates a choice of the elements between the braces.

definition [] - Indicates an option of the elements between the brackets.

call - Transfers control to a command or subsystem.

character - A character may be a letter, number, symbol, <CR>, tab, space, etc..

command - A command is an instruction to the computer which specifies an operation to be performed.

<CR> - Represents the RETURN key.

CTRL- - A control character is a non-printing ASCII character which is (usually) used to transmit control signals between a peripheral device and the computer. For example, a CTRL-W moves the cursor up in the Screen Editor.

cursor - The screen pointer is positional reference indicator on the screen.

default - A reversion to a value already programed into the Screen Editor.

delimiter - A character which limits a string of characters.

enter - Synonymous with "type in".

<esc> - Represents the escape key.

file - A file defines a group of related information. This information is addressed by means of a file name and usually resides on a floppy diskette.

file name - This is a one to eight character label which is used to refer to a file. Several files may have the same file name. These files may be uniquely identified by the use of a disk specifier and/or a file name extension.

file ref - See the definition of file name above.

flag - An indicator used for identification. In this manual, the flag indicates the cursor direction.

instruction - A statement that specifies an operation.

line - A horizontal row of characters.

marker - Markers are positional reference points in a file. They may accessed with the Jump command.

repeat factor - Incremental indicator for repeated commands. The Find command will locate the nth occurrence of a string. The Replace command will substitute n occurrences of a string with another string.

rubout - A rubout is a backspace and erasure of a character.

string - A string is a sequence of characters.

tabs - Tabs are positional reference points on a line. They may be accessed with the tab key or CTRL-I.

INDEX

Cromemco
incorporated
Tomorrow's Computers Today
280 BERNARDO AVE. MOUNTAIN VIEW, CA 94043