SCREEN EDITOR

Commodore 64

commodore COMPUTER

THE COMMODORE 64 SCREEN EDITOR

COPYRIGHT

DISCLAIMER

COMMODORE BUSINESS MACHINES, INC. ("COMMODORE") MAKES NO WARRANTIES, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THE PROGRAM DESCRIBED HEREIN, ITS QUALITY, PERFORMANCE, MERCHANTABILITY, OR FITNESS FOR ANY PARTICULAR PURPOSE. THIS PROGRAM IS SOLD "AS IS". THE ENTIRE RISK AS TO ITS QUALITY AND PERFORMANCE IS WITH THE BUYER. SHOULD THE PROGRAM PROVE DEFECTIVE FOLLOWING ITS PURCHASE, THE BUYER (AND NOT THE CREATOR OF THE PROGRAM, COMMODORE, THEIR DISTRIBUTORS OR THEIR RETAILERS) ASSUMES THE ENTIRE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION AND ANY INCIDENTAL OR CONSEQUENTIAL DAMAGES. IN NO EVENT WILL COMMODORE BE LIABLE FOR DIRECT, INDIRECT, INCIDENTAL OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY DEFECT IN THE PROGRAM EVEN IF IT HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. SOME LAWS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF IMPLIED WARRANTIES OR LIABILITIES FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THE ABOVE LIMITATION OR EXCLUSION MAY NOT APPLY.

PREFACE

The **Commodore 64 SCREEN EDITOR** software package allows you to create BASIC programs with the aid of a Screen Field Editor.

The SCREEN EDITOR program provides the user with a very powerful software tool that allows the definition and editing of data fields that are input through the CRT screen. Screen design and the input and editing of data, have always been one of the more difficult tasks facing the BASIC programmer. With the Commodore 64 SCREEN EDITOR, screen design and the input and editing of data is not only easier, but accomplished with the accuracy and speed of Assembly Language programming.

This package contains everything that the user will need to create, edit, and manipulate data fields from within a BASIC program. You will notice that like the software contained on the diskette, this user's manual is directed towards the experienced computer user that already has some familiarity with the BASIC language and the operations of the Commodore 64 computer.

This product is not intended to provide the knowledge of 'how to' in BASIC language, but provides the software tools for the experienced programmer to become even better.

USER CONVENTIONS

It is recommended that you familiarize yourself with the Commodore keyboard. Here is a brief description of certain keys and symbols, and their respective function in reference to the SCREEN EDITOR program.

Comma and Period , .	Both of these characters are normally used in the editing of numeric 'dollar' amount fields.
Left and Right Parentheses ( )	Both of these delimiters, i.e. beginning and ending symbols, are used to define a NUMERIC ONLY field.
Left and Right Bracket [ ]	Both of these characters are used to define an alpha-numeric field.
Greater Than and Less Than > <	Both of these characters are used to define an alpha-numeric field.
(CR)	To continue on with the program after a line of input, press the RETURN key.

TABLE OF CONTENTS

SECTION I INTRODUCTION to the COMMODORE 64 SCREEN EDITOR

1.1 Getting Started 1

SECTION II DESCRIPTIONS

2.1 Field Descriptions 2

2.2 Command Descriptions 3

2.3 Operation Descriptions 5

SECTION III OPERATIONS

3.1 Control Operation 6

3.2 Array Operation 6

SECTION IV SCREEN EDITOR FEATURES

4.1 Extended Status 8

4.2 The Credit Flag 8

4.3 Illegal Format 8

APPENDIX I PROGRAM EXAMPLE 9

SECTION I
INTRODUCTION TO THE COMMODORE 64 SCREEN EDITOR

1.1 Getting Started

The purpose of the Screen Editor is to provide the programmer with a standardization tool for the definition and editing of data fields that are input through the CRT screen. The Commodore 64 Screen Editor program, written in the 6502/6510 language, also gives the user several additional BASIC commands.

This package contains everything that you, as a programmer, will need to create, edit, and manipulate data fields from within a BASIC program. Just like the software contained on this diskette, this user's manual is directed towards the experienced computer user who is already familiar with the BASIC language and the Commodore 64 computer.

For 'Direct' load, input **'LOAD"EDITOR64",8,1'** (CR). Then input **'SYS49152'** (CR). The Screen Editor is now enabled.

For examples of how to automatically Load the Editor and use its specific features, load and RUN the first program on the diskette. A listing of the example program is contained in Appendix I.

SECTION II DESCRIPTIONS

2.1 Field Descriptions

The Commodore 64 Screen Editor uses three 'types' of fields for the purpose of data entry. These fields are:

- **Alphanumeric, non-field edit, left to right entry**
- **Alphanumeric, field edit, left to right entry**
- **Numeric, right to left entry**

Each of these three data types are defined on the screen by using the following delimiters:

< >	Names, addresses, cities, etc. Legal data for this field is any character from the keyboard.
[/ /]	Dates, part numbers, telephone numbers, etc. Legal data for this field is any character from the keyboard except the 'comma(,), period (.), slash(/), and dash(-)' which are simply passed over for easy editing by the Screen Editor program.
(, .)	Monetary amounts, quantities, etc. Any number is legal in this field. Commas and decimal points are passed over for easy editing by the Screen Editor program.

ALL of these data fields can be placed at any location of the screen except the first and last lines of the screen. The first and last lines are reserved as Program Status Lines.

***Please Note:** A negative (dash sign) before a field, or credit flag ('cr') after a field, will change the field sign. Also, an "Illegal Format" error message will result if the Screen Editor program detects any incorrect pattern mixing or incomplete sets of symbols, i.e. '>>', '[[', '<] >'.

2.2 Command Descriptions

The following commands have been added to the BASIC command list. A **description of each follows.**

***Please Note:** The Screen Editor program requires that the first character of each command be a '&' symbol.

&b	To display the bottom Program Status Line
&c	To change all screen colors
&e	To enter the data entry/edit mode, allowing the user to edit any field on the screen
&f	To edit a single field on the screen
&k	To disable the use of the RUN/STOP key
&l	To draw a horizontal line across the screen
&s	To enable the use of the RUN/STOP key
&t	To display the top program status line

Two of the above commands, **'&e'** and **'&f'**, have optional suffixes to make the command more versatile. For example:

The command '&e,(5)' places the cursor at the beginning of the fifth field on the screen. You may then edit as many fields as you desire. The range of the field selection is $0 < x < 99$.

The command '&f,(12)' places the cursor in the twelfth field. The twelfth field can then be edited. Control is then returned to the BASIC program.

If the field specified with '&e' and '&f' does not exist, the program prints the following message:

```
?undef'd field error
ready.
```

To change the color of the background, border, and characters, use the **'&c'** command. For example*: &c,"XYZ" specifies that the border color (first character) is to be 'X', the background color (second character) is to be 'Y', and the character color (third character) is to be 'Z'. The default colors are blue, blue, and white for the border, background, and character colors respectively. This command should have all characters specified, or the value not specified will be replaced by the default color.

***Please Note:** In the above example, 'XYZ' would of course be the actual control characters of the colors you desire.

The next three commands **'&l'**, **'&t'**, and **'&b'**, are for screen printing. These commands are for separating data, or to notify the operator of certain functions. A more detailed description of each follows:

The **'&l'** command directs the program to print a line of 40 characters across the screen at the next available print position. The default character to be printed is '='. To change this character, simply POKE the address with the ASCII value of the character you wish to have printed.

For example, to produce a line of dashes (-):

```
POKE 49360,ASC("-"):&l
```

The Screen Editor displays two bars referred to as the Program Status Lines. These lines are the top and bottom lines of the screen. The next two print commands are specifically for each of the Program Status Lines:

&t	The **'&t'** command displays the top program status line, erasing the previous contents.
&b	The **'&b'** command displays the bottom program status line, replacing the previous contents with "busy".
***Please Note:**	With either the '&t' or '&b' command, the Screen Editor program will automatically display "busy" on the bottom status line. This allows the programmer to print "busy" without having to code another BASIC subroutine.

The last two commands are concerned with the operation of the STOP key. With either command, the SCREEN EDITOR program will automatically print "busy" on the bottom Program Status Line.

&s	To disable the STOP key*
&k	To enable the STOP key

***Please Note:** When the STOP key is disabled, the internal clock continues to function normally.

2.3 Operation Descriptions

It is relatively easy to operate the Commodore 64 Screen Editor. Simply type the data into the fields via the computer keyboard. Any typed character that does not comply with the rules of each particular field is ignored. The computer operator may also use the standard cursor control keys. The operation descriptions for the standard cursor keys are detailed below:

CRSR DOWN	The CRSR DOWN key (arrow pointing down) moves the cursor to the beginning of the next field. If there happens to be no other field, the Screen Editor program will automatically go into the Screen Accept Mode.
CRSR LEFT	The SHIFT key in conjunction with the CRSR LEFT key (arrow pointing to the left) moves the cursor one position to the left.
CRSR RIGHT	The CRSR RIGHT key (arrow pointing to the right) moves the cursor one position to the right.
CRSR UP	The SHIFT key in conjunction with the CRSR UP key (arrow pointing up) moves the cursor to the beginning of the previous field. If there doesn't happen to be a previous field, the command is ignored.
DEL	The DEL key deletes the previous character entered from the field and moves all data after the cursor, one position to the left.
HOME	The HOME key places the cursor at the beginning of the first field with no data loss.
INST	The SHIFT INST key inserts a space into a field and moves all data after the cursor one position to the right.
SHIFT CLR	The SHIFT CLR key clears all fields on the screen and places the cursor at its default location.
UP ARROW	The UP ARROW key deletes the character under the cursor and shifts the data after the cursor one position to the left.

The aforementioned function keys operate in any field except the numeric fields. The numeric fields cannot accept the INST key or the UP ARROW key because the data in the numeric fields is right justified. Therefore, the INST and UP ARROW keys become unnecessary.

SECTION III OPERATIONS

3.1 Control Operation

The Commodore 64 Screen Editor program has a built in 'control mode' routine. This routine can be modified by the programmer to produce status codes on any key on the keyboard. The 'defined' codes are limited to four.

To use the control mode, simply press the control (CTRL) key. The program then displays 'control mode' on the bottom Program Status Line. To exit back into the main program, simply press the 'Commodore' key .

The control mode has two preset options. These options are **'e'** and SHIFT **'q'**. The 'e' key exits from the basic program and loads the first program found on the disk drive. A RUN is also pushed into the keyboard buffer to assure the execution of the program.

The next command should be used with extreme caution. The SHIFT 'q' command resets the Commodore 64, i.e., this removes the basic Screen Editor program from memory.

Concerning the four function keys a programmer can define, the following table displays some possible ASCII values which can be POKEd into the program much like the &I (line) command. For example, to produce an ST 02 with the F1 key, input **'POKE 51350,ASC("F1")'**. The status, 'ST', value is the value which is returned to the BASIC program by the Screen Editor.

POKE	ST
51350	02
51361	03
51372	04
51383	05

The ST is used so that the BASIC programmer has easy access to the status of the Screen Editor. Other status values are produced and are listed later on in this manual.

3.2 Array Operation

The Screen Editor stores the contents of the screen fields in the string array 'SC$()'. This enables the BASIC programmer to process the data returned from the editor. The editor will fill the array with as much data as possible. The programmer **MUST** define the array and fill the array with spaces to match the field lengths.

The first field on the screen is stored in array element zero (0). The second field is in location one. This process continues for each field until the array is full.

However, if the programmer fails to fill the array with spaces, the Screen Editor aborts, and no data is returned to the BASIC program. Here are some examples of the array operations:

Example 1:
An array element has been defined as " " (three spaces). The corresponding field on the screen contains '<abcdef>'. The data returned to the BASIC program is 'def'. In this example, **the Screen Editor truncates data after the array element is filled.**

Example 2:
An array element has been defined as " " (ten spaces). The corresponding field on the screen contains '[815-555-2222]'. The data returned to the BASIC program is '8155552222'. In this example, the **Screen Editor ignores dashes. Commas and slashes are also ignored. However, if a number is required, decimal points are transferred.**

Example 3:
An array element has been defined as " " (eight spaces). The corresponding field on the screen contains '(12,444.55)'. The data returned to the BASIC program is '12444.55'. **In this example, the Screen Editor ignores the comma and transfers the decimal point.**

Example 4:
An array element has been defined as " " (eight spaces). The corresponding field on the screen is '-(12,444.55)'. The data returned to the BASIC program is '12444.55'. **Because the SC$() is not large enough, the Screen Editor program does not transfer the negative sign.**

Example 5:
An array element has been defined as " " (nine spaces). The corresponding field on the screen is '-(12,444.55)'. The data returned to the BASIC program is '-12444.55'. **Because the SC$() was large enough, the Screen Editor transferred the negative sign.**

Example 6:
An array element has been defined as " " (nine spaces). The corresponding field on the screen is '(12,444.55)'. The data returned to the BASIC program is '12444.55'. **The Screen Editor ignores commas.**

Example 7:
An array element has been defined as " " (eight spaces). The corresponding field on the screen is '(12,444.55)cr'. The data returned to the BASIC program is '12444.55'. **Because the SC$() is not large enough, the negative value, notated by the credit 'cr', is truncated.**

Please Note: The Screen Editor ignores all other arrays except the 'SC$()' array.

Example 8:
An array element has been defined as " " (nine spaces). The corresponding field on the screen is '(12,444.55)cr'. The data returned to the BASIC program is '-12444.55'. Because the SC$() was large enough, the Screen Editor transforms the credit 'cr' into a negative sign.

SECTION IV SCREEN EDITOR FEATURES

4.1 Extended Status

The status variable is set to zero (0) if the Screen Editor has no errors. This means that the data is valid and processing can continue. The status variable is set to one (1) if the RUN/STOP key is pressed. In normal program operation, the RUN/STOP key is primarily used when the operator needs help. Any disk, printer, or tape operations will clear the status for helping the operator.

4.2 The CREDIT Flag

As previously seen in this manual, the programmer may select either a negative sign (dash) '-', or a Credit flag 'cr', on example types such as 'Example 3'. The 'cr' flag will set ALL the fields, not only one field. Before going to the edit mode, the programmer should set the format which he/she wishes to use. The 'cr' flag is located at 52921 and the values are as follows:

POKE 52921,1	To set the Screen Editor to print 'cr'
POKE 52921,0	To set the Screen Editor to print a dash(-)

The default value for the 'cr' flag is zero.

***Please Note:** The CR flag **MUST** be set to either zero or one for the Screen Editor program to operate properly.

4.3 Illegal Format

The prompt "ILLEGAL FORMAT" will be displayed if any delimiters are mixed, or an incomplete set is input. For example, if the screen editor encounters a field defined by ' [( ) > '.

```
1000 GOTO1140
1010 "********************************
1020 "*          ⌐DITOR ⌐EMO V64       *
1030 "*              (⌐)1982           *
1040 "* ⌐OMMODORE |USINESS \ACHINES    *
1050 "*                                *
1060 "* ⌐ATE ωRITTEN: 11/08/82         *
1070 "* ♠UTHOR        : \ICHAEL ●CHAFF *
1080 "*                                *
1090 "* ⌐ONFIGURATION:                 *
1100 "*          ⌐OMPUTER: ⌐OMMODORE 64 *
1110 "*                                *
1120 "********************************
1130 :
1140 REM ** LINES 1160 THRU 1180 ARE THE EDITOR AUTO BOOT! **
1150 :
1160 IFPEEK(49153)=19ANDPEEK(49162)=198ANDPEEK(49163)=104THEN1190
1170 PRINT"[][][][]LOAD"CHR$(34)"EDITOR64"CHR$(34)",8,1[][][][][][][][][]SYS49152[]"
1180 POKE198,3:POKE631,13:POKE632,13:POKE633,131:END
1190 PRINT"[][][][]"
1200 :&C,"[][][]"
1210 REM" ↑↑↑ /
1220 REM" ||└BACKGROUND COLOR
1230 REM" |└BORDER COLOR
1240 REM" └CHARACTER COLOR
1250 :&S REM                                        *    KILL THE STOP KEY
1260 POKE49360,ASC("*"):REM                         *    SET TYPE OF LINE
1270 POKE51350,ASC("▆"):REM                         *    SET STATUS 02
1280 POKE51361,ASC("▀"):REM                         *    SET STATUS 03
1290 POKE51372,ASC("▌"):REM                         *    SET STATUS 04
1300 POKE51383,ASC("▐"):REM                         *    SET STATUS 05
1310 POKE52921,1:REM                                *    SET 'CR' FLAG
1320 GOTO1580:REM                                   *    GOTO MAIN LINE!
1330 :
1340 REM" _OUTINE FOR DISPLAYING TOP STATUS LINE
1350 :&T:PRINT"[][]"TAB(((40-LEN(TM$))/2))TM$"[][]":RETURN
1360 :
1370 REM" _OUTINE FOR DISPLAYING BOTTOM STATUS LINE
1380 :&B:PRINT"[][][][][][][][][][][][][][][][][][][][][][][][][]"TAB(((40-LEN(BM$))/2))BM$"[][]":RETURN
1390 :
1400 REM" _OUTINE FOR ●-_¯¯/ #1
1410 PRINT"[]":&B:&T:TM$="⌐OMMODORE \AIL \ATE V64":GOSUB1340
1420 PRINT"[][][]       ⌐ROGRAM \ODE              <"SC$(0)">[]":&L
1430 PRINT" /AME      <"SC$(1)">":PRINT" ♠DDRESS <"SC$(2)">"
1440 PRINT"[][] ⌐HONE /UMBER        ["LEFT$(SC$(3),3)"-"MID$(SC$(3),4,3)"-";
1450 PRINTRIGHT$(SC$(3),4)"][][]":PRINT" ⌐ATE ["LEFT$(SC$(4),2)"/";
1460 PRINTMID$(SC$(4),3,2)"/"RIGHT$(SC$(4),2)"]        ⌐UES $("MID$(SC$(5),2,1);
1470 PRINT","RIGHT$(SC$(5),6)")":IFLEFT$(SC$(5),1)="-"THENPRINT"[]"SPC(37)"CR"
1475 PRINT"[][] \EMBER TIL ▌[▌"LEFT$(SC$(6),2)"/";
1480 PRINTMID$(SC$(6),3,2)"/"RIGHT$(SC$(6),2)"▌]▌":PRINTSPC(12)"↑           ↑"
1490 PRINTSPC(12)" |IDDEN FIELD OPTION!!!":&L:RETURN
1500 :
1510 REM" _OUTINE FOR FILLING BL$() W/NULL VALUES
1520 BL$(0)="    ":BL$(1)="                              ":BL$(2)=BL$(1)
1530 BL$(3)="            ":BL$(4)="        ":BL$(5)="      .00":BL$(6)=BL$(4)
1540 FORA=0TO6:SC$(A)=BL$(A)+"":NEXT:RETURN
1550 :
1560 :
1570 :
```

```
1580 REM **** MAIN LINE PROGRAM ****
1590 GOSUB1510
1600 GOSUB1400:BM$="┐RESS ♥|┌┐ FOR  |ELP":GOSUB1370
1610 :&F(0):IFST=0THEN1640
1620 PRINT"❑":&T:&B:PRINT"■■■■■■■■"SPC(10)"♥TATUS IS:"ST:FORA=1TO1000:NEXT
1630 GOTO1600
1640 GOSUB1370:&E(1):IFST>0THEN1620
1650 PRINT"❑":&T:&B:TM$=" |UR SCREEN DATA":GOSUB1340:PRINT"■■■■■"
1660 FORA=0TO6:PRINT"SC$("RIGHT$(STR$(A),1)")="CHR$(34)SC$(A)CHR$(34):NEXT
1670 BM$="┐RESS ANY KEY TO EXIT":GOSUB1370
1680 GETA$:IFA$=""THEN1680
1690 GOTO1580
```

commodore COMPUTER

Commodore Business Machines, Inc.
1200 Wilson Drive • West Chester, PA 19380

Commodore Business Machines, Limited
3370 Pharmacy Avenue • Agincourt, Ontario, M1W 2K4

C64109

Printed in U.S.A.