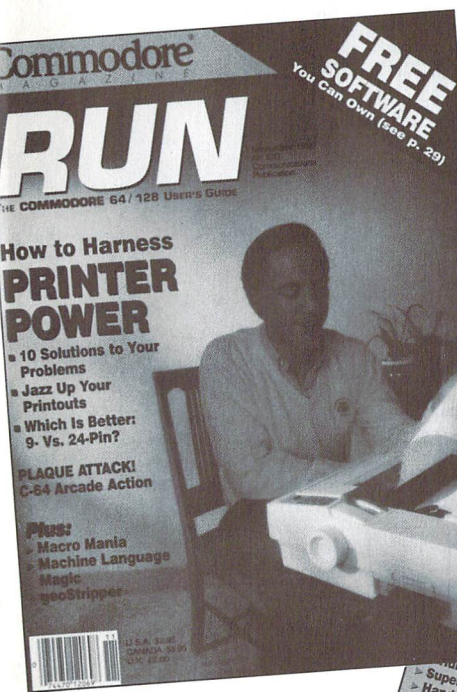


November/December 1990 Edition

# RE RUN

## RUN Programs on Disk

For the C-64 and C-128



**Plus: Extra Bonus Programs!**



[www.Commodore.ca](http://www.Commodore.ca)

May Not Reprint Without Permission

# Introduction

*November/December '90 ReRUN*

FATE NEVER FAILS. Just when I think we can't possibly come up with an assortment of programs better than the last, this edition of ReRUN comes along. The *RUN* editorial staff has amassed a wide range of games, a financial application, some programs for use with GEOS and utilities—to mention just a few!

Let's get started. For the November 1990 issue of *RUN*, I wrote my annual printer article, primarily in an effort to reduce the trials and tribulations of buying a printer. The editorial staff decided that the first program on this disk, William McLeod's Printer Controller 64/128, would be the perfect complement to my article. For that matter, it would complement any printer article!

With Printer Controller, you can set margins, select print styles and choose the size of your fonts. After running the program, selecting the Quit option returns you to Basic and leaves the new settings to your printer still intact. Printer Controller adds a new dimension of simplicity to printer programming.

If you like the traditional Pac Man game, you'll love our next program for the C-64—Plaque Man. Even if you don't like games, dust off the old joystick and introduce your children or grandchildren to Plaque Man. It'll keep them glued to the computer at home, instead of the video machines at the local arcade hall.

Many Commodore computerists know that the question mark is the Basic abbreviation for the command Print. However, most other keywords require at least two or more keypresses when abbreviated. Our next title on this disk is Macro Mania, a utility that makes programming on your 64 easier because it types an entire keyword at the press of the Commodore and shift keys. Best of all, instructions are included for customizing Macro Mania to your liking.

Machine Language Magic follows next. Use it to create windows, various fading effects and many other tricks, all without learning how to program in machine language on your 64.

The final program from the November issue is geoStripper, a handy little GEOS utility that comes out of that month's geoWatch column. Using GEOS and geoStripper on your 64 or 128, you can



remove the overhead code from geoBasic programs. This conserves precious disk space and lets you save more geoBasic programs to disk. GeoStripper also allows for faster loading of geoBasic programs.

The December issue of *RUN* is well represented, too. The first program, Shopping for a House?, written by real estate professional, Theodore Yonan, quickly determines your home-buying qualifications. Whether you want to own your first home, or are buying for the tenth time, Shopping for a House?, otherwise known as Prequalify 64/128, analyzes your income and expenditures to determine whether or not you qualify to borrow.

An edition of ReRUN just wouldn't be complete without a Tony Brantner game, and so we present GridRunner. Race around the grid and try to avoid the little varmints chasing you.

On a more serious note is BEE 64. No, it's not a spelling bee program; rather, BEE stands for Basic Editor Enhancer, a programming utility that adds a host of new commands to Basic 2.0.

We wrap up the December issue with Map Drawing Demo, which is from that month's 128 Mode column. Author Mark Jordan first takes you through the how-to's of Basic map drawing on your 128, and then illustrates his technique with a demo program that draws a map of the U.S. on your 40-Column mode screen.

As always, we conclude with a look at the bonus programs. First up is an imaginative, two-player game called Marked Square. What sets this game apart from others is that it works on the 64 and 128 in any mode. Plus, its design affords players the ability to make amazing comebacks near the end of a game, resulting in unpredictable outcomes. Play it at your next party and liven up the evening!

Finally, there's GEOS Disk Editor, a pure machine language disk editor for your GEOS disks. GEOS users itching to do a little disk programming can use it to perform anything from retrieving accidentally deleted files to installing personal copy protection. Be forewarned, however—if you change the wrong byte, irreversible damage can occur to your GEOS disks.

That's it for this edition. Enjoy your holidays. I'll be back in the new year with the first 1991 edition of ReRUN!



*Technical Manager  
RUN Magazine*



# Directory

PAGE	DOCUMENTATION	DISK FILENAME	FILE TYPE
		*MENU 128 _____	BASIC
		MENU 64 _____	BASIC
1	<b>PRINTER CONTROLLER 64/128</b>	PRINTER 64/128 _____	BASIC
4	<b>PLAQUE MAN</b>	PLAQUE.BOOT _____	BASIC
		DM _____	ML
		MAKE PLAQUE _____	BASIC
4	<b>MACRO MANIA</b>	MACRO MANIA _____	BASIC
		MACRO MANIA.ML _____	ML
		MAKE MACRO MANIA _____	BASIC
7	<b>MACHINE LANGUAGE MAGIC</b>	BOOT TEXT FX _____	BASIC
		TEXT FX _____	ML
		TEXT FX DEMO _____	BASIC
		TEXT FX WINDOW _____	BASIC
		TEXT FX-4 _____	BASIC
		MAKE TEXT FX _____	BASIC
12	<b># GEOSTRIPPER</b>	GEOSTRIPPER _____	BASIC
13	<b>SHOPPING FOR A HOUSE?</b>	PREQUALIFY _____	BASIC
15	<b>GRIDRUNNER</b>	GRIDRUN.ML _____	ML
		MAKE GRIDRUN _____	BASIC
16	<b>BEE 64</b>	BEE 64 _____	BASIC
19	<b>MAP DRAWING DEMO</b>	*US-MAP _____	BASIC
22	<b>£ MARKED SQUARE</b>	MARKED SQUARE _____	BASIC
23	<b>#£ GEOS DISK EDITOR</b>	GEOS DISK EDITOR _____	ML

\* — C-128 mode only

# — Requires GEOS

£ — Bonus program

Before you run a program, carefully read the documentation that pertains to it.



# How To Load

## LOADING FROM MENU

To get started, C-64 users should type LOAD "MENU 64",8 and press the return key. When you get the Ready prompt, the menu is loaded and you should type RUN to see a list of the programs on your disk. C-128 users need only press the shift and run-stop keys. When all the programs are displayed on the screen, you can run the one you select by pressing a single key.

## LOADING FROM KEYBOARD

If you do not wish to use the menu program, follow these instructions.

**C-64:** To load a C-64 program written in Basic, type: LOAD "DISK FILENAME",8 and then press the return key. The drive will whirl while the screen prints LOADING and then READY, with a flashing cursor beneath. Type RUN and press the return key. The program will then start running. To load a C-64 program written in machine language (ML), type: LOAD "DISK FILENAME",8,1

**C-128:** All C-64 programs can be run on the C-128 as long as your computer is in C-64 mode. All C-128 programs are clearly labeled on the directory page. Your C-128 *must* be in C-128 mode to run these programs. To load a C-128 mode program, press the F2 key, type the disk filename and then press the return key. When the program has loaded, type RUN.

## MAKING COPIES OF ReRUN FILES

Many programs on your ReRUN disk have routines that require a separate disk onto which the program writes or saves subfiles. To use these programs, you must first make a copy of the original program onto another disk that has enough free space on it to hold these newly written subfiles.

It's simple to make a copy of a Basic program. Just load it into your computer as outlined above, and then save the program back onto a separate disk that has plenty of free space for extra files.

Copying an ML program is not so simple. You cannot simply load and save an ML program; you'll need to use a disk-backup utility program, such as the one on your Commodore Test Demo disk.

---

RUN it right: C-64 or C-128; printer

# Printer Controller 64/128

By William McLeod

TODAY'S SOPHISTICATED DOT MATRIX printers require a hundred or more commands to fully utilize their capabilities, and many printing options remain unused because of the time required to look up and invoke the commands. Consequently, we settle for pedestrian printouts instead of the more attractive styles that are available. I developed Control Printer 64/128 to make it easier to use printer options by making it simpler to input printer control codes.

Control Printer 64/128 runs on the C-64 or C-128 (40 or 80 columns), automatically configuring itself to either computer. The program was written for the Star Micronics NP-10 and compatible printers, but many of the commands work with most other parallel printers. Control Printer can be easily modified for use with other printers. As written, it *won't* work with Commodore-dedicated printers.

The top line of the first screen presents the main menu: Style, Margins, Utility and Quit. As indicated by a command line at the bottom of the screen, you make a selection by using the cursor-right key to highlight your choice and then pressing the return key. Selecting Style displays a pull-down secondary menu of 11 styles of printing. Make your choice, press return, and your printer will be configured to that feature. Of course, the printer must be connected and turned on for the selection to work.

You can enter several commands, such as Elite, Italics, Expanded and Boldface, and the printer will combine them all and retain them until it is shut off or reset.

After you've made your selection of styles, press return, and you'll go back to the main menu. Now select Utility and press return; then choose and execute View Printout. The printer will print a list of the styles selected and a short sentence illustrating the style combination, with the paper rolling up an inch, so you can see what was printed. The Sample Printout option prints the same thing without rolling the paper up. If you haven't selected a style, the printer will respond with Pica, as this is the default on most printers.





The Margins menu is for setting margins, of course. However, it also lets you set the printer to skip over perforations and provide paging.

At any time, you can save your work to disk by typing RUN 20 and pressing return. Line 20 establishes the name of the program, 30 scratches the old version, and 40 saves the new version. If you make changes in the program, you should change line 20 to ZZ\$="1PRINTER" or some other name in order to preserve the original.

## MAKING CHANGES

You can change the titles in the main menu by changing the Data statements in line 120, but don't change END in line 130, as it serves as a terminator for the array. The menu is limited to four titles of 18 characters each for the C-128 in 80-Column mode and seven characters in 40-Column, or 64, mode. Anything longer will automatically be truncated. If you want more than four items in the main menu, you're on your own, as this requires extensive alterations to Control Printer 64/128.

Data for the Style menu is found in lines 220–230, Margins in 320–330 and Utility in 420–430. Each of their secondary menus can handle up to 11 titles, so there's room for a total of 11 more commands—five in the Margins menu and six in the Utility menu.

To add a command to the Margins menu, add the command's title to line 330 and its subroutine starting on line 1790, erasing the REM and using the same format as in the other command routines. Each new routine must start at the next available REM in that menu, and it must not occupy a line already containing a REM.

New commands that require entering a column number should be placed in the Margins menu, as it is equipped for that type of input. Of course, you can delete any menu item and substitute your own in its place.

## PRINTER COMMANDS

If Control Printer 64/128 runs all right, but the printer isn't responding properly, the interface probably isn't in Transparent or Semi-Transparent mode. This is handled throughout the program with an OPEN 5,4,S command, where S is set to 5 in line 90. Most interfaces use 5 as a secondary address for Transparent mode, but that number does vary. Consult your interface manual and change line 90 or flip the necessary DIP switches on your printer interface for either Transparent or Semi-Transparent mode.



Now we get down to the actual printer commands. First, run through all the menus to become familiar with the procedure. You'll find that it's very easy. Next, reset the printer and exit to the main menu. Now try each of the styles in turn, printing a sample and resetting the printer each time.

If any command fails to print correctly, look at lines 1000-1320, and note that each group of lines is labeled with its title and that the second line in each group is the command line. For emphasized print, the command is `PRINT#5, C$"E"`, where `C$` is previously defined as `CHR$(27)`. Most of the codes use `C$`. The code `C$"E"` can also be written `CHR$(27);CHR$(69)`, where `CHR$(69)` is the code for the letter E. Look up the code for emphasized print in your printer manual, and enter it in place of `C$"E"`. Any other printer commands that fail to work can be handled in a similar manner, assuming that your printer supports that style. All commands for the Margins menu start at line 1520, and those for the Utility menu start at 1820.

All of this is reasonably straightforward, except for the addition of line 1670 to disable the interface control for the right-hand margin. If your interface doesn't have a disable command, you may be able to set it at a high number such as 150 or 200.

## USING THE PROGRAM

For use with a word processor, set up the printer as desired before loading the word processor. The printer will retain the combination you set, but it's always possible that the word processor will have ideas of its own. In that case, look in the word processor's manual for a way to prevent that software from doing its own printer initialization.

By using Elite, Condensed and Subscripts commands and reducing line spacing to one-twelfth inch, you can produce a disk directory that can be pasted on the front of the disk sleeve. The type is quite readable, because it's a dense black. Just type `LOAD"$",8` and return, then `OPEN 4,4:CMD4:LIST`.

Double-height letters are ideal for name tags, and quad height works fine for posters. Form feed is great for printing out listings of long programs without printing over the perforations. You'll find many other uses for Control Printer 64/128, as you gain experience with it. And, it's fun printing out all the combinations of styles!



---

RUN it right: **C-64; joystick**

# Plaque Man

*By David Martin*

PLAQUE MAN IS A fun-filled, fast-paced C-64 derivation of the arcade game Pac Man. The object is to maneuver the white tooth through the on-screen maze, collecting enamel molecules—the small white dots—and earning points as you go. Sounds simple, doesn't it? Well, it isn't, thanks to the ever-present plaque. The globs of plaque are after the tooth and, if they make contact, it dies.

The plaque's intelligence level makes it almost impossible to avoid. Your best bet is to eat one of the red fluoride pills located near the four corners of the maze. Absorbing a fluoride pill enables your tooth to destroy the plaque, and you earn extra points.

After plugging a joystick into port 2, you begin the game with three teeth. After completing a level without losing all three, the action speeds up, and you earn an extra tooth.

That's all there is to it. Just remember the Golden Rule: Brush often and take your fluoride pills!

---

RUN it right: **C-64**

# Macro Mania

*By Stephen Dirschauer*

MACRO MANIA IS A short utility that's handy for any C-64 user. Why? Well, even in this age of point-and-click interfaces, typing is still a major part of interacting with a computer, and Macro Mania can save you countless keystrokes. How? By defining 45 macro keys that can be added to your C-64. Each key prints its own macro of up to ten characters.

Macro keys aren't new to the C-64. You can find them in any of the many utilities that let you program your C-64's normally functionless function keys. With the utility in place, hitting a function key automatically prints a Basic command or some other text. If you've used them before, you know they can save you a great deal of time.

However, they aren't perfect. My main gripe is that you're limited to four, or, at most, eight macro keys. Macro Mania provides 45. The other utilities are also useless with a number of programs that use the function keys themselves. Macro Mania doesn't meddle with unshifted function keys, so it's compatible with most programs that utilize them.

After loading the program from Menu 64, entering SYS 49699 turns Macro Mania on and off.

## USING MACRO MANIA

You access any of Macro Mania's macros by holding down the Commodore key and a shift key and hitting the appropriate macro key. This may sound like an acrobatic maneuver, but it's really quite simple and quick. Each key will print its own macro only if the Commodore and shift keys are held down. Otherwise, it will act as it normally would.

Although you can program your own macros (I'll show how below), Macro Mania comes with a built-in set, listed in Table 1 along with their corresponding keys. Italics in the table indicate nonprintable characters, listed in a format similar to *RUN's* program listings. For instance, 5 *CRSR DN*s is five presses of the cursor-down key.

## DEFINING MACROS

You can define any of the 45 macro keys like this:

SYS 50000,"key","macro"

For example, to set the P macro to poke, you'd enter:

SYS 50000,"P","POKE"

Macro Mania doesn't limit you to literal strings like Poke. Any string expression will do, as in these three examples:

SYS 50000,A\$,"APPLE"

SYS 50000,"K",JT\$(5)

SYS 50000,P\$,RIGHT\$(I\$,5)



To add the return key to a macro, just add CHR\$(13) to its definition. For example:

SYS 50000,"L","LIST"+CHR\$(13)

The L macro would then automatically list the program in memory. You can also put macro definitions in programs.

If you'd like to view a macro's contents, omit the second parameter:

SYS 50000,"A"

Clear a macro by setting it to a null string:

SYS 50000,"C",""

## COMPATIBILITY AND MORE

As a general rule, Macro Mania will work with any program that doesn't disturb memory from 49699 to 50919 (\$C223-\$C6E7) or change the keyboard vector at 655 (\$028F). That includes most Basic, and even many machine language, programs. I've used Macro Mania with games, utilities and two machine language word processors. However, the only way to find out if it'll work with a specific program is to experiment.

When you turn Macro Mania on and off with SYS 49699, the current macro set won't be harmed.

Finally, Macro Mania disables the Commodore/shift combination's usual function: toggling between the C-64's two character sets. With Macro Mania in place, you can hold down Commodore and shift and hit the delete key to get the same effect.

**Table 1. Default macro set.**

Key	Macro		
A	AND	I	INPUT
B	OR	J	INPUT#
C	CHR\$(	K	READ
D	DATA	L	LOAD
E	CLOSE	M	MID\$(
F	FOR	N	NEXT
G	GOTO	O	OPEN
H	GOSUB	P	PRINT

Key	Macro	Key	Macro
Q	SYS 49749	3	OPEN 4,4 <i>return</i>
R	RETURN	4	CMD 4 <i>return</i>
S	STOP	5	CLOSE 4 <i>return</i>
T	TO	6	LOAD "*",8:
U	THEN	7	LOAD "\$",8:
V	VERIFY	8	CRSR RT, DEL
W	WAIT	9	5 DELs
X	RUN	0	LIST: <i>return</i>
Y	RESTORE	F1	5 CRSR UPs
Z	GET#	F3	5 CRSR LFs
↑	SYS 50000,"	F5	5 CRSR RTs
@	<i>undefined</i>	F7	5 CRSR DNs
*	<i>undefined</i>	cursor-down	<i>undefined</i>
1	LOAD "*",	cursor-right	<i>undefined</i>
2	LOAD "\$",		

---

RUN it right: C-64

# Machine Language Magic

*By Brandon Corfman*

IT HAPPENS TO ALMOST every programmer: You get a great idea for a new game or utility on your C-64, but when it comes to writing the code, Basic is just too slow to handle all the routines. You need the speed of machine language, but it takes too much time to learn. If only there were a utility that would supply the machine language for you!

Look no further. Text FX provides all the machine language copying, moving, scrolling and window routines, with resulting special effects, that you need in your Basic programs, leaving you free to develop that ultimate program without worrying about speed.

Text FX is designed to be invisible to Basic. Windows are stored



under Basic and Kernal ROM, and the program itself takes no space from Basic RAM. Also, because Text FX uses SYS commands instead of keywords, the routines are as fast as possible.

To see Text FX in action, load and run Text FX Demo. This program gives you a short demonstration of the capabilities of Text FX, and, I hope, some ideas for using it in your own programs.

The starting memory address of Text FX is 49152 (\$C000), and a jump table is located at the beginning of the program for convenience in accessing the routines. Text FX uses addresses \$02, \$00F8–\$00FF, \$0334–\$033A, \$A000–\$BFFF and \$E000–\$FFFF for storage. The TextFill, MultiScroll, WindowSave, WindowRestore and InvertChar commands automatically calculate the current address of screen RAM and adjust their execution accordingly.

Note: At the beginning of a program with window commands, *always* reset the number of windows to zero with POKE 826,0.

## WINDOWINIT AND WINDOWFADE

WindowInit and WindowFade let you fade any character window to black, so it disappears into a black background. WindowInit, which must be activated first, sets up coordinates for the character window. WindowFade then does the actual step-by-step fading with a program loop. Run Text FX Window to see a sample routine that uses these commands.

Activate WindowInit with

```
SYS SA,X1,Y1,X2,Y2
```

where SA is the starting address of Text FX, X1 and Y1 are the X and Y coordinates of the top-left corner of the window, and X2 and Y2 are the bottom-right coordinates.

Activate WindowFade with

```
SYS SA+3
```

No parameters are needed here, but a loop is required to make the screen fade in steps. I designed WindowFade with a loop, so you can regulate the speed of the fade. Simply insert a delay between the SYS statement and the If statement in line number 40 of Text FX Window.

When the screen is completely black, WindowFade tells your

program by storing 1 in address 2. The loop will be broken, and your program will go on to the next line.

## **FADEINIT AND FADESCREEN**

This pair of commands fades the entire screen, including the background and border, to black. FadeInit must be activated before FadeScreen for your program to work correctly.

Activate FadeInit with

SYS SA+6

No parameters are needed, since FadeInit involves the entire screen, not just a window.

Activate FadeScreen with

SYS SA+9

No parameters are needed here, either, but a loop similar to the one in WindowFade is required. Here again, you can put a delay in the loop if needed. FadeScreen operates much like WindowFade, except for affecting the entire screen.

## **COLORFILL AND TEXTFILL**

As the name implies, ColorFill fills any portion of color RAM with the color you specify. It's much faster—and more versatile—than using Pokes from Basic. ColorFill should be activated prior to TextFill, especially on old C-64s where Pokes to screen RAM are invisible. A sample routine that uses these commands appears in Text FX-4.

Activate ColorFill with

SYS SA+12,X1,Y1,X2,Y2,COLOR

where X1, Y1, X2 and Y2 are the window coordinates, and COLOR is the number, 0–15, of the Commodore color to put in color RAM.

TextFill has two main purposes: to clear portions of the screen used for windows and to provide borders for windows. Activate it with

SYS SA+15,X1,Y1,X2,Y2,SC

where X1, Y1, X2 and Y2 are the window coordinates, and SC is a screen character code, 0–255. You can find the character codes in the *Commodore 64 User's Guide*.

## **WINDOWSAVE AND WINDOWRESTORE**

WindowSave saves any portion of the screen (characters and their

colors) to the area under Basic or Kernal ROM, where no memory can be stolen from Basic or most machine language programs. WindowSave's companion command, WindowRestore, retrieves the last window stored and places it on the screen.

Activate WindowSave with

SYS SA+18,X1,Y1,X2,Y2

where X1, Y1, X2 and Y2 are the window coordinates. Up to eight windows can be active at once.

Activate WindowRestore with

SYS SA+21

This command needs no parameters.

## **SCROLLINIT AND MULTISCROLL**

These commands are used together to scroll all or part of the screen in any direction. ScrollInit is the set-up routine, while MultiScroll does the actual scrolling.

Activate ScrollInit with

SYS SA+24,D,B1,B2

where D is the direction you want the screen to move, and B1 and B2 are the X or Y boundaries of the area that moves. The possible direction values are 1 for up, 2 for right, 3 for down and 4 for left. When scrolling up or down, B1 and B2 are X coordinates in the range 0-39; when scrolling right or left, B1 and B2 are Y coordinates in the range 0-24.

Activate MultiScroll with

SYS SA+27

This command needs no parameters. Once you've set a direction and boundaries with ScrollInit, you can use MultiScroll repeatedly until another Text FX command is executed.

## **INVERTCHAR**

The InvertChar command toggles the high bit of screen memory in the screen window you specify, inverting the video in the window.



This is especially handy for highlighting menu options and flashing part of the screen. Activate InvertChar with

`SYS SA+30,X1,Y1,X2,Y2`

where X1, Y1, X2 and Y2 are the window coordinates.

## **MEMFILL**

MemFill is a general-purpose command for clearing a hi-res screen or erasing a section of memory. Activate MemFill with

`SYS SA+33,ADDRESS 1,ADDRESS2,VALUE`

where ADDRESS 1 and ADDRESS 2 are the beginning and ending addresses for the fill operation, and VALUE is the number of the character to insert.

## **MEMCOPY**

Copying memory at machine language speed is easy with MemCopy. Try using this command to copy character sets or hi-res screens, or, with MultiScroll, to scroll screens for an arcade game. Activate MemCopy with

`SYS SA+36,ADDRESS 1,ADDRESS2,DESTINATION START`

where ADDRESS 1 and ADDRESS 2 are the beginning and ending addresses of the memory area you're copying from, and DESTINATION START is the beginning address of the memory area you're copying to.

## **DESIGNER'S NOTES**

I certainly liked the effect of the fade routines when I wrote them, but after a while, I began to wish for fade-ins, too. At first, I thought the mechanics of doing a fade-in would be beyond my ability and patience, but when I started thinking about how to use the window routines in programs, it came to me. While the screen is blanked, you can save each step of a fade-out as a window, then flip the screen back on and resurrect the text in a few easy steps. And, this takes only a few seconds! To turn off the screen, use `POKE 53265, PEEK (53265) AND 239`; to turn it back on, use `POKE 53265, PEEK (53265) OR 16`.

Text FX doesn't provide for restoring windows in a different order and at different locations, but it can be done—at your own risk and with your own error checking. Here's how:

Address 826 (\$033A) is used by Text FX to hold the number of the window currently open. This number increases or decreases accordingly each time a window is opened or closed. To change the window order, change the value in this address to the number of the window you want.

The X and Y values specified in each WindowSave command are stored as the first four bytes of the window's data with the starting memory addresses of the eight windows at \$A000, \$A800, \$B000, \$B800, \$E000, \$E800, \$F000 and \$F800. Changing these X and Y values changes the location on the screen where WindowRestore places the window. There's only one problem: WindowRestore doesn't check to see if the data it resurrects is off screen, so you could end up overwriting your Basic program.

Text FX will make a world of difference in your programming capability. Its ready-made machine language routines will free you to do the creative programming you want to do.

---

RUN it right: **C-64 or C-128; 1541 or 1571; GEOS (any version)**

# GeoStripper

*By William Coleman*

GEOBASIC LETS YOU CREATE "stand-alone" applications; that is, programs that can run without the need for the geoBasic file itself. Unfortunately, however, a stand-alone geoBasic file takes up much space, approximately 30K. I wrote geoStripper to remove extraneous data and return your program to the size of the original source code. GeoStripper, a short GEOS program, runs on any version of GEOS, in either 64 or 128 mode.

To use geoStripper, place a GEOS work disk in drive 8 of a 1541 or 1571—not a 1581—disk drive. When you run the program, it creates geoStripper on the work disk.

GeoStripper is easy to use: You simply double-click on the geoStripper icon. You'll then see a list of the stand-alone files on the disk. Select the file you want to strip and click on open. You can also click on the Disk icon if you want to insert a different disk. For

those users with more than one drive, I've included a drive icon that toggles among drives A, B and C.

Note that geoStripper directly modifies the file you've selected, so always make a duplicate of that file *before* stripping.

---

RUN it right: C-64 or C-128 (in 40- or 80-Column mode)

# Shopping for a House?

By Theodore Yonan

ONE OF THE MOST difficult aspects of buying a house is shopping for the mortgage. Will a financial institution trust you to pay back \$100,000 or more? Asking will be easier if you use Prequalify 64/128 to assess your qualifications *before* you go to the lender.

## GUIDELINES

Run Prequalify 64/128 from either Menu 64 or Menu 128. To use Prequalify, simply input the figures appropriate to your current financial situation at each prompt.

The qualification guidelines lenders use for conventional loans are simple, but exact figures differ from lender to lender. Prequalify uses typical figures and assumes that your monthly housing expenses shouldn't exceed 28 percent of your gross monthly income and, with monthly debt payments added in, shouldn't exceed 36 percent.

Refer to Table 1 to see how the guidelines are calculated. Monthly income (A) includes all your predictable income (salaries, bonuses, commissions, interest income, and so on). Monthly housing expenses (B) include payments on loan principal and interest, insurance, taxes, homeowners' association fees and assessments for special community services. Liabilities (C) include monthly payments on alimony and child support, loans, charge accounts and credit cards (including a 5 percent monthly charge against total card indebtedness), tuition, other real estate, and any other regular obligations.

Note that if your down payment is less than 10 percent of the purchase price, guideline 1 drops to 25 percent and guideline 2 to 33 percent.

Also, if your down payment is less than 20 percent of the purchase price, the lender will insist that you take out private mortgage insurance (PMI) at a cost of 1–1.25 percent of the loan amount, paid up front at closing, plus 0.3–0.4 of 1 percent, paid monthly (\$25–\$33 per month on a \$100,000 loan).

Income minus housing and liability payments equals your monthly expendable income, which covers food, clothing, entertainment, medical, dental, education, transportation and other “discretionary” expenses. It is expendable income that determines to a substantial degree your quality of life, and the VA and FHA weigh it more heavily than banks in determining qualification for a loan. A conventional loan is granted on the basis of credit worthiness and the value of the property held for security; the FHA and VA make sure you’ll end up with enough cash to live decently.

After the screening to see if you qualify financially, most lenders want to check your financial, personal and employment records. Prequalify prints a checklist of such items you can review before going to your lender.

## **OTHER COSTS**

Assuming your loan is approved, you must have sufficient funds to cover closing costs (the down payment, points, and so on).

On a conventional loan, a down payment is 10–20 percent of the purchase price. On an FHA loan, it’s 3 percent of the first \$25,000, plus 5 percent of the amount beyond \$25,000. On a VA loan, no down payment is needed if the price is equal to four times the

**Table 1. Loan qualification guidelines.**

### **Components:**

- A. Monthly income
- B. Monthly housing expense
- C. Monthly liability (debt) payments

### **Guidelines:**

- 1. B divided by A not to exceed 0.28 (28 percent)
- 2. B plus C divided by A not to exceed 0.36 (36 percent)

entitlement (benefits provided by the government to a specific group), but for any amount over four times the entitlement, a down payment is 25 percent.

Then there are points, each representing approximately 1 percent of the loan amount. Don't be bashful about shopping around to get the lowest percentage.

You may also have to impound escrow money to cover property taxes, mortgage insurance premiums and community service fees.

A loan origination fee may be necessary. It can range from 0.5 to 2 percent of the loan amount, so, once again, shop around.

Finally, count on other fees and charges approaching 1-2 percent of the loan amount.

Shopping for a home will challenge your business acumen and negotiating skills. Use Prequalify 64/128 to help you play the game prudently.

---

RUN it right: **C-64; joystick**

# GridRunner

*By Tony Brantner*

IT'S EASY TO PILOT your ship along the lines of the grid, collecting diamonds as you go. Easy, that is, until the guards roaming the outskirts of the grid start shooting at you! When that happens, you'll see why GridRun 64 is such a challenging and addictive game.

To play, make sure you have a joystick plugged into port 2, then load and run GRIDRUN.ML from Menu 64.

When the game screen appears, your ship is centered at the bottom of the grid. Move the ship in any direction to start play, then steer the ship around the grid. Passing over diamonds collects them and earns you points. Each diamond is worth ten times the current level of play.

Soon after your ship starts to move, the guards prowling the edges of the grid will start launching blasts of atomic energy that can send your ship into a tailspin. Getting hit not only costs you a ship, but resets the grid with more diamonds. However, if you clear the grid



of diamonds without getting hit, you advance to the next, slightly faster, level of play.

You start a game with three ships and get another for every 5000 points earned. The game ends when you have no more ships in reserve. If you'd like to play again, simply press the firebutton.

As you play, the screen continually displays your score in the current game, your highest score so far in the session, the current level of play and the number of ships you have left.

Now, place your ship on the line. . .

---

RUN it right: C-64

# BEE 64

*By Michael Rebella*

THE LACK OF COMMANDS in the C-64's ROM editor can make it difficult to enter programs that are written in Basic. Super BEE (BEE stands for Basic Editor Enhancements) adds eight new commands to the editor to remedy many of its problems. The commands include automatic line numbering, sending disk commands, deleting a group of lines, giving a directory, finding a string, merging files, renumbering lines and unnewing.

Since Super BEE resides in memory locations 49152-50382, it doesn't cause any problems with your Basic programs. Neither does it affect any of the regular Basic commands.

Each Super BEE command starts with a single quote (shift/7), which lets the Basic interpreter recognize the command without tokenizing it. Not only does the single quote allow my program's commands to be executed, but it also lets you distinguish, at a glance, Super BEE commands from Basic 2.0 commands.

With the exception of the single quote, enter Super BEE com-

mands exactly like Basic commands. You can enter them one per line or several on a line separated by colons.

Following are descriptions of Super BEE's eight commands.

## THE COMMANDS

**'AUTO step** enables automatic line numbering for typing in a program. "Step" specifies the amount (greater than zero) to be added to the previous line number. You can stop entering lines by pressing the return key immediately after a line number, without entering any other text. Note that to avoid erasing this line, you should hold down the shift key and then press return, or use the cursor keys to move to another line. This does not turn off automatic line numbering, so the next time you enter a line number with following text, another line number will be printed. To turn off line numbering, enter 'AUTO without specifying "step."

Examples:

'AUTO 10 enables automatic line numbering with an increment of 10.

'AUTO turns off automatic line numbering.

**'COMMAND string,device** sends the commands in "string" to the device number specified in "device." If a device number isn't specified, 'COMMAND defaults to drive 8. In "string," use the same commands as those specified in the disk drive owners' manual.

Examples:

'COMMAND "N0:DISKNAME,ID"

'COMMAND "S0:FILE",9

'COMMAND A\$,D

**'DELETE first TO last** deletes all lines from "first" through "last," where "first" is less than or equal to "last." If "first" is omitted, all lines from the beginning of the program through "last" are deleted. If "last" is omitted, all lines from "first" through the end of the program are deleted.

Examples:

'DELETE 30 TO 50

'DELETE TO 50 deletes all lines except those after 50.

'DELETE 100 TO leaves only the lines numbered less than 100.



**'\$, device** gives the directory of a disk without affecting the program in memory. Again, "device" specifies the drive number and defaults to drive 8.

**'FIND string,first TO last** searches for all occurrences of "string" in the program from lines "first" through "last" and lists the lines containing "string" to the screen. "String" can be either a string or a string variable. To find a string containing quotation marks, substitute +CHR\$(34) for the marks. "First" and "last" follow the same rules as in 'DELETE, except they can both be left out to search through the whole program. You can stop the screen listing by pressing the stop key.

Examples:

'FIND "GOTO",50 TO 100 finds all occurrences of "GOTO" in lines 50 through 100.

'FIND "IF A=1" searches for "IF A=1" in all lines of the program.

'FIND "PRINT"+CHR\$(34) searches for "PRINT" in all lines of the program.

**'MERGE filename,device** adds the Basic program in the file specified by "filename" to the end of the Basic program already in memory. The combined program is stored in memory and renumbered with the same starting number and step size that were last used in the 'NUMBER command described below. Again, "device" specifies the device number and defaults to drive 8.

Examples:

'MERGE "END PROG",1

'MERGE "PROGRAM"

**'NUMBER first,step** renumbers a program, starting with "first" and with each successive line number "step" larger than the previous. If you wish to use the same values for "first" and "step," simply enter 'NUMBER without specifying "first" and "step." 'NUMBER does not change line numbers in GoTo and GoSub statements; you must change these yourself.

**'UNNEW** revives a Basic program that you erased with the New

command. 'UNNEW will not work if you assigned any variables or entered any new lines since the New command.

### **ADDITIONAL NOTES**

Always make sure a formatted disk is in the drive when using the '\$ command. Otherwise, you will find garbage, followed by an infinite number of carriage returns, printed to the screen. If this happens, press run-stop/restore and then enter CLOSE 1. Super BEE is not disabled by the restore key or in any way other than turning off the computer.

---

RUN it right: **C-128 (in 40-Column mode)**

# Map Drawing Demo

*By Mark Jordan*

IT'S EASY TO DRAW MAPS on a C-128. All you need is a rudimentary knowledge of Basic (which follows) and a map to refer to.

Map-making on the 128 is so simple because its Graphics mode parallels the way maps are drawn: Both use a set of coordinates to draw lines. On the 128, these coordinates are known as X and Y, and they correspond to the horizontal and vertical axes. On a map, the corresponding coordinates are latitude and longitude. Since latitude and longitude can be determined for any point on the Earth's surface, duplicating that point on your 128 is as easy as putting map coordinates into the Draw command in a continuous stream.

### **DRAWING A CONTINENT**

Let's mosey through a very simple map-making venture in Basic 7.0. Since Australia is a neat, well-defined continent, we'll draw it. To show how simple the process is, we'll do the whole thing from Direct mode, which means no program lines are needed.

First get into Graphics mode with:

GRAPHIC 2,1,10

This puts us into split-screen graphics (with the split at line 10), so that we can type in our commands at the bottom half of the screen and see the map at the top.

Next, we need a map of Australia that has longitude and latitude lines. (It's a good idea to have the map in front of you now, so you can follow along.) Australia's northeastern peninsula comes to a sharp point, so let's use it as our starting point. My map says the longitude is about 143, and the latitude is approximately 11. Begin with:

```
DRAW 1,143,11
```

If you press the return key after typing the above, you'll just get a dot on your screen. We want to use the continuous drawing feature for our map, so don't press return just yet.

From this point, we need to move in a straight line as far as we can. Since the eastern slope of Australia is roughly straight all the way to Brisbane, let's make it our next set of coordinates (153 longitude, 28 latitude). The line should now look like this:

```
DRAW 1,143,11 TO 153,28
```

See how easy it is? We can go to the next prominent point by adding another TO onto the same line. Let's go down to just east of Melbourne (149,38), then to just west of Melbourne (140,38), then arc northwest to the southern border between West Australia and South Australia (130,30), and finally over to the southwest corner (116,34). Here's what the line looks like so far:

```
DRAW 1,143,11 TO 153,28 TO 149,38 TO 140,38 TO 130,30 TO 116,34
```

There's a place called Onslow that's almost a straight shot up Australia's western coast. It's at 115,22. From Onslow, we need to move northeast to near Darwin (132,12) before heading back south to Normanton (141,18). We complete our map by going back to the same coordinates we started with—143,11. Here's the entire continent in one Direct mode line:

```
DRAW 1,143,11 TO 153,28 TO 149,38 TO 140,38 TO 130,30 TO 116,34  
TO 115,22 TO 132,12 TO 141,18 TO 143,11
```

Go ahead, press return and watch what happens. Viewing the drawing in progress should convince you of two things: anybody can do this, and ours would be a more refined map if we had taken the time to plot more points.

The more coordinates, the more accurate the map. However, you

would need to turn your draw lines into program lines. When you do that, you must start each succeeding Draw command with relative coordinates. Here's how:

```
DRAW 1,+0,+0 TO. . .
```

The two +0s keep the pixel cursor where you last left it, so you can add more coordinates and continue your drawing.

Our map of Australia has only one problem—it's small. To double its size, we simply need to multiply the coordinates by 2; to triple its size, multiply by 3; and so forth. It's a good idea to put your coordinates into Data statements and read them out from inside a loop. That way, you can factor in the constant as you read them. If you do it this way, you'll need to type in an extra Draw command at the beginning to establish the starting point, like this:

```
10 DRAW 1,143*7,11*7
20 FOR T=0 TO 9:READ X,Y:DRAW 1,+0,+0 TO X*7,Y*7:NEXT
30 DATA 143,11,153,28,149,38,140,38,130,30,116,
    34,115,22,132,12,141,18,143,11
```

Enlarging our map to fit the screen presents another problem—centering it. This can be done by adjusting the coordinates of the northwest corner of the map. First, find the west-most point (for Australia, it's 115) and deduct that value from all longitude points. Then, find the northern-most point (11) and deduct that value from all the latitudes. The loop would be readjusted like so:

```
10 FOR T=0 TO 9:READ X,Y:DRAW 1,+0,+0 TO (X-115)*8,
    (Y-11)*8:NEXT
```

## SOME CAUTIONS

This exercise helps to understand the nature of mapping and its inherent difficulties. One obstacle is that Earth is round, not flat, and any attempt to display it on a two-dimensional surface will do injustice to the place you're drawing. Another factor is that nature's borders, unlike man-made borders, are never straight lines. Compared to Australia, drawing Colorado is a snap.

The biggest difficulty you'll run into is the fact that longitude is measured west or east from the Greenwich meridian up to 180 degrees, and latitude is measured south or north from the equator up to 90 degrees. This means that, for areas in the western and northern hemispheres, longitude coordinates increase from right to left, and latitude coordinates increase from bottom to top, which is

exactly opposite from the normal way of reading coordinates. I chose Australia for our tutorial map because it's in the southern and eastern hemispheres, and thus the map draws the way we would expect it to, with longitude increasing from left to right and latitude from top to bottom. The United States is just the opposite; draw its coordinates directly, and you'll get a map with Florida pointing towards the upper-left corner of your screen!

The way around the problem is simple: Subtract all longitude and latitude values from 180 to reverse them. The accompanying demo program (which draws a very crude U.S. map) demonstrates this. To see the demo, run US-MAP in 40-Column mode.

The demo also shows another Basic command at work: Scale. With Scale, you can stretch and shrink maps in many interesting ways. When you run the demo, feel free to experiment.

---

RUN it right: C-64 and C-128 (40- or 80-Column mode)

# Marked Square

*By Brian Ellis*

THE OBJECT OF MARKED SQUARE is to obtain a score higher than your opponent's. The concept behind this two-player game is quite simple, but winning requires skill and planning.

All input is via the keyboard, so after loading the program (using Menu 64), player 1 begins by entering his or her choice of X and Y grid coordinates (any combination of two numbers between 1,1 and 9,9). The square that the coordinates designate is marked blue when it's player 1's turn and yellow when player 2 makes a move. The player making the last move to fill in a vertical, horizontal or diagonal line within the grid is awarded one point for each square captured, and the line fills in with that player's color.

Once a number of squares are filled in, it is possible to capture a line simply by entering one set of coordinates. Your score increases every time you complete a line. The winner of the game is the player with the greatest number of squares captured, not the player with the most lines filled in.

# GEOS Disk Editor

By Greg Badros

GEOS PROGRAMMERS, like all programmers, need a good sector editor for inspecting and changing bytes on disk. Unfortunately, standard disk editors aren't very useful for editing GEOS disks, because they support only Commodore PETSCII, not the true ASCII that GEOS uses. GEOS Disk Editor solves this problem, making editing GEOS disks a breeze. It lets you view and alter everything on the disk including file types, date and time stamps, icon data, file comments, disk names, filenames, text within GEOS or an application, and even actual programming: The possibilities are endless. It can also be used as a traditional sector editor.

Note that GEOS Disk Editor is a very powerful program, but with power come hazards. Writing a sector with just one wrong byte could erase an entire file (at least until the wrong byte is fixed). *You must be very careful when editing a disk.* In fact, you probably should make a backup copy of the disk before changing anything.

Load and run GEOS Disk Editor just like a Basic program; you don't have to run it from within GEOS. It's compatible with any printer that accepts PETSCII characters and all Commodore disk drives, including the 1581. When you run the program, pressing the U key toggles between device 8 and device 9 to let you choose a drive. Press the return key to start work, then at the prompt, enter the track and sector that you want.

## THE SCREEN DISPLAY

The editing screen is divided into two main sections. The top half is the block (sector) contents window; the bottom half shows other information you will need to intelligently edit the current sector. All values are given in both decimal and hexadecimal. When you press the £ key, the bottom of the screen switches to a Help screen that lists most of the editing commands.

## MOVING THE CURSOR

Once a sector has been read in and displayed, a flashing cursor



appears in the upper-left corner of the block contents window. To edit the correct bytes within the sector, you must use the cursor keys to move the cursor to the proper position. Note that the keys wrap both horizontally and vertically.

There are also several special keys for moving the cursor around the screen. The space bar acts like cursor-right, except it always moves to the next position in the sector. For instance, when the cursor is at the right edge of the block contents window, cursor-right wraps it to the far left of the same line, while space moves it down one line as well. The delete key is the opposite of space; it moves the cursor to the left. Both space and delete wrap between positions 0 and 255.

The home key works as usual, returning the cursor to the upper-left corner of the block contents window.

The last special cursor control is shift/return. It returns the cursor to the far left of the current line.

Once the cursor has been moved, all information is updated to reflect the new position.

## THE COMMANDS

GEOS Disk Editor has many commands that make editing disks easy. An explanation of each command, along with the keypress to activate it, follows.

Note that all commands that require numerical input can accept both decimal and hexadecimal values. Just press return after typing in the number, and be sure to precede hexadecimal numbers with a \$. If you decide to abort any of these commands, press return without entering a number.

**F1/F2:** increments/decrements the track number. Note that these commands do not read in the sector (see F5/F6).

**F3/F4:** increments/decrements the sector number. These commands don't read in the sector, either (see F7/F8).

**F5/F6:** increments/decrements the track number and reads the sector in sequential order after changing the track.

**F7/F8:** increments/decrements the sector number and reads the sector after changing it.

**A:** changes a two-byte address value. The A command works like the B command, except it changes two bytes and moves the cursor two positions to the right.





**B:** changes a single byte value. You will be prompted to enter a value to store at the current position. After you change the byte, the cursor advances to the next position.

**C:** shifts to a different disk. GEOS Disk Editor will prompt you to insert the new disk, then you'll enter a track and sector to be read and edited (see the S command).

**D:** sends a command to the disk drive. GEOS Disk Editor sends the command, then reads and displays the error channel. Note that you must send a u0>m1 command to switch a 1571 to Double-Sided mode, then u0>m0 to switch back to Single-Sided mode.

**J:** reads the sector pointed to by the track and sector link under the cursor and in the next position. For instance, in a GEOS directory entry the two bytes after the filename represent the track and sector of the file header. When you position the cursor at the first of these bytes and press J, GEOS Disk Editor automatically reads in the file header.

**L:** reads the sector that was read just before the current sector was read in. (See the information below about the list of accessed sectors.)

**N:** reads the next sector in the list of sectors accessed. If you haven't used the P command or are at the end of the list, N works like the + command. (See the information below about the list of accessed sectors.)

**O:** outputs the currently displayed sector to the printer. You will be prompted to turn on the printer. Pressing return starts the printing, while any other key aborts it. The printout will consist of a position within the sector (displayed in hexadecimal), followed by 16 consecutive hexadecimal values from the sector and the true ASCII characters they represent.

**P:** inputs new text at the current cursor position in the sector. All the keys you press are stored in memory as Commodore PETSCII values. Pressing return exits this mode. You must follow the P command with the W command to write the sector to disk.

**Q:** quits editing and returns to Basic.

**R:** reads into memory and displays the information in the sector currently listed.

**S:** changes the sector. You will be prompted to input values for the track and sector. Note that this command automatically reads and displays the sector. Use the function keys instead if you don't want the sector to replace the current one in memory.

**T:** like P, inputs new text at the current cursor position in the

sector. However, here all keys are stored in memory as true ASCII for use with GEOS programs. Return exits this mode also, and, once again, the W command is required to write to disk.

**U:** toggles the drive device number between 8 and 9. See the ↑ command for setting the drive type.

**W:** writes the sector (in the block contents window) to the track and sector currently listed. You *must* use this command to save any changes you've made before you edit a new block.

**+**: reads the sector pointed to by the next sector link information at the beginning of the current sector.

**\***: hunts for a string of values. This is GEOS Disk Editor's most advanced command, and probably the most useful. (See the information on Hunt mode below.)

**↑:** cycles through the disk drive types. The box to the right above the block contents window changes according to your key presses. Among other things, the drive type determines which sector contains the directory and how many tracks and sectors (maximum) there are.

**←:** cycles through the sector type messages. (See the information about messages below for details.) This is the only command that's not listed on the Help screen.

## THE ACCESSED SECTORS LIST

Two of the above commands control an ordered listing of accessed sectors. Every time a sector is read in, GEOS Disk Editor remembers its location and type. Then, after you've read in another sector, you can reread the first just by pressing L. You can continue using L to trace backward through the sectors you've already accessed, then, at any time, use N to reverse direction and move forward through those sectors. It's possible to step back and forth through up to 30 sectors using the L and N commands. However, if you step back too far, GEOS Disk Editor will inform you of a disk error, and you'll have to enter a new sector number. Changing disks or toggling the device number clears the list.

## THE MESSAGES

One thing that sets GEOS Disk Editor apart from other editors is its fully documented sectors. Whenever you read in a sector, the program checks to see what type it is and displays that information in the sector-type display area. Then, depending on the type and the cursor position, the byte information area displays specific messages about what the current byte represents.

For example, if you've just read in track 18, sector 0 with a 1541 drive, the sector type will be Dir. Header. Then, if you move the cursor to position 171, GEOS Disk Editor will say in the byte information area that the current byte is a link to the off-page directory sector that GEOS uses. Pressing J (jump to cursor link) then reads that sector. Now the sector type will be Off-Page Dir., and the byte messages will be appropriate for an off-page directory sector.

In general, GEOS Disk Editor is good at determining what kind of sector it has in memory. However, if the sector type is the default, Data Block, and you know you're editing a special type of sector, use the ← command to cycle through the sector types until the one displayed is correct. This will enable GEOS Disk Editor to give you the appropriate byte information.

## HUNT MODE

GEOS Disk Editor has many powerful features, but the Hunt command is a special joy to use. When you press the H key, the bottom message box will ask what you want to do. If you want to return to editing, just press return and the messages will disappear. Otherwise, you have two options: hunt for a string or set the Hunt mode.

If you want to hunt for a string, you have several choices of exactly what to look for. You can enter either a string of ASCII characters (press A), a string of PETSCII characters (press P), a sequence of bytes (press #) or whatever you hunted for last (press N, for next occurrence). Once you make that selection, GEOS Disk Editor starts looking from the current cursor position for the string or bytes you entered. If it can't find them in the current sector, it automatically continues looking in the next. The Hunt mode determines where the editor will find the next sector. If the hunt is successful, the cursor will appear at the start of the area you searched for, and you'll see a Found! message. If it's unsuccessful, a Not Found! message appears.

Setting the Hunt mode is also easy. After pressing H, just press F if you want GEOS Disk Editor to follow the track and sector links in the first two bytes of each sector, or press D if you'd rather go sequentially through each sector on the disk. The File mode ends when a track link of 0 is found, while the Disk mode ends after searching to the end of the disk.

## AN EXAMPLE

Let's edit one of the text messages in geoPaint to demonstrate



several functions of GEOS Disk Editor. After running the program, insert a 1541-formatted work disk that contains a copy of geoPaint into the drive, and enter 18 and 0 when prompted for the track and sector. You should then see the sector type Dir. Header displayed, along with the screen codes for each of the 256 bytes in track 18, sector 0 of the disk in device 8.

Since we want to change information stored in the geoPaint application, we must first find its directory entry, which will tell where on the disk the data for the application is stored. Before doing the search, however, we must set the editor to Hunt mode, so press H and then F. Now press H to hunt, then A to signify an ASCII string search. When prompted for the string, enter GEO-PAINT (notice it's all capital letters).

Since the Search mode defaults to File mode, the track and sector links are followed through the entire disk directory. In fact, you'll see each successive sector that GEOS Disk Editor loads as it's looking for the proper file entry.

Before you know it, your cursor will be sitting on the G of GEOPAINT, the ASCII string display area will show the complete filename, and the message box will tell you the string has been Found!.

Next, we want to go to the place on the disk where the program and text for geoPaint are stored. To do this, cursor back (left) twice. Now the byte information box will display Link: 1st Data Block. Note that two consecutive bytes produce this message. Be sure you're on the first of these (the one that points to the track) before pressing J to jump to that sector. Now a sector type of VLIR Index will appear.

This new sector is actually the index to the variable length records (remember, GEOS uses a VLIR file system for many applications). Each two bytes starting at position two are a track and sector pointer to a record in that file. Usually, the first of these records will contain any text you would want to change, so cursor right over to position two and press J again to go to the first sector in that record. Now the sector type Data Block will appear.

Next, we're ready to change the text. It's much easier to shorten than lengthen a message, so, for the sake of this example, let's change "update" in the File menu to "save." First, we must find where "update" appears in the record. Note that GEOS Disk Editor is still set to File mode, so issue an H command followed by an A to again hunt for an ASCII string. This time enter update (all



lowercase letters) and wait a minute or so as the editor searches for the string. When it's been found, your cursor will appear on the u in update. Now press T to enter ASCII text at the current cursor position, type in save, and press return. Since save is two bytes shorter than update, you'll need to show GEOS that the menu text ends after the e by using the B (Byte) command with a value of 0. This null signifies the end of the string. After doing all that, press W to write the sector as modified back to disk.

You can now load GEOS and double-click on geoPaint. Hereafter, whenever you use that copy of geoPaint, "save" will appear in the File menu instead of "update." Try it and see!

Remember, you don't need to limit yourself to menu choices. Try changing messages that appear in dialog boxes. You can even edit the deskTop!

## CLOSING NOTES

Although the byte information is very helpful, especially in finding the proper byte position to edit, sometimes interpreting the value stored there can require a bit more assistance. A copy of *The Official GEOS Programmer's Guide* will come in handy while editing, or even just viewing, disks.

Don't forget that GEOS Disk Editor isn't just for GEOS disks. It works well with regular disks, too. Just ignore the byte information and the sector type. The ASCII String and ASCII Text mode will also prove useless, but everything else applies to standard disks as well as GEOS disks. GEOS Disk Editor should save hours of work and frustration for the intermediate user and for the advanced programmer. ■

# **RE == RUN**

EDITOR-IN-CHIEF

**DENNIS BRISSON**

TECHNICAL MANAGER

**TIM WALSH**

MANAGING EDITOR

**BETH S. JALA**

SENIOR EDITOR

**HAROLD R. BJORNSEN**

COPY EDITOR

**PEG LePAGE**

ART DIRECTOR

**HOWARD G. HAPP**

DESIGN AND LAYOUT

**ANN DILLON**

TYPESETTING

**DEBRA DAVIES**

**KEN SUTCLIFFE**

FULFILLMENT CONSULTANT

**DEBBIE BOURGAULT**



[www.Commodore.ca](http://www.Commodore.ca)

May Not Reprint Without Permission

# 11 Programs Included on this Disk

## *From the November RUN:*

- ▶ Printer Controller 64/128
- ▶ Plaque Man
- ▶ Macro Mania
- ▶ Machine Language Magic
- ▶ GeoStripper

## *From the December RUN:*

- ▶ Shopping for a House?
- ▶ GridRunner
- ▶ BEE 64
- ▶ Map Drawing Demo

## *Plus: Extra Bonus Programs!*

- ▶ Marked Square
- ▶ GEOS Disk Editor

If any manufacturing defect becomes apparent, the defective disk will be replaced free of charge if returned by prepaid mail within 30 days of purchase. Send it, with a letter specifying the defect, to:

**RUN Special Products • 80 Elm Street • Peterborough, NH 03458**

Replacements will not be made if the disk has been altered, repaired or misused through negligence, or if it shows signs of excessive wear or is damaged by equipment.

The programs in ReRUN are taken directly from listings prepared to accompany articles in *RUN* magazine. They will not run under all system configurations. Use the RUN It Right information included with each article as your guide.

The entire contents are copyrighted 1990 by IDG Communications/Peterborough. Unauthorized duplication is a violation of applicable laws.

©Copyright 1990 IDG Communications/Peterborough



[www.Commodore.ca](http://www.Commodore.ca)

May Not Reprint Without Permission