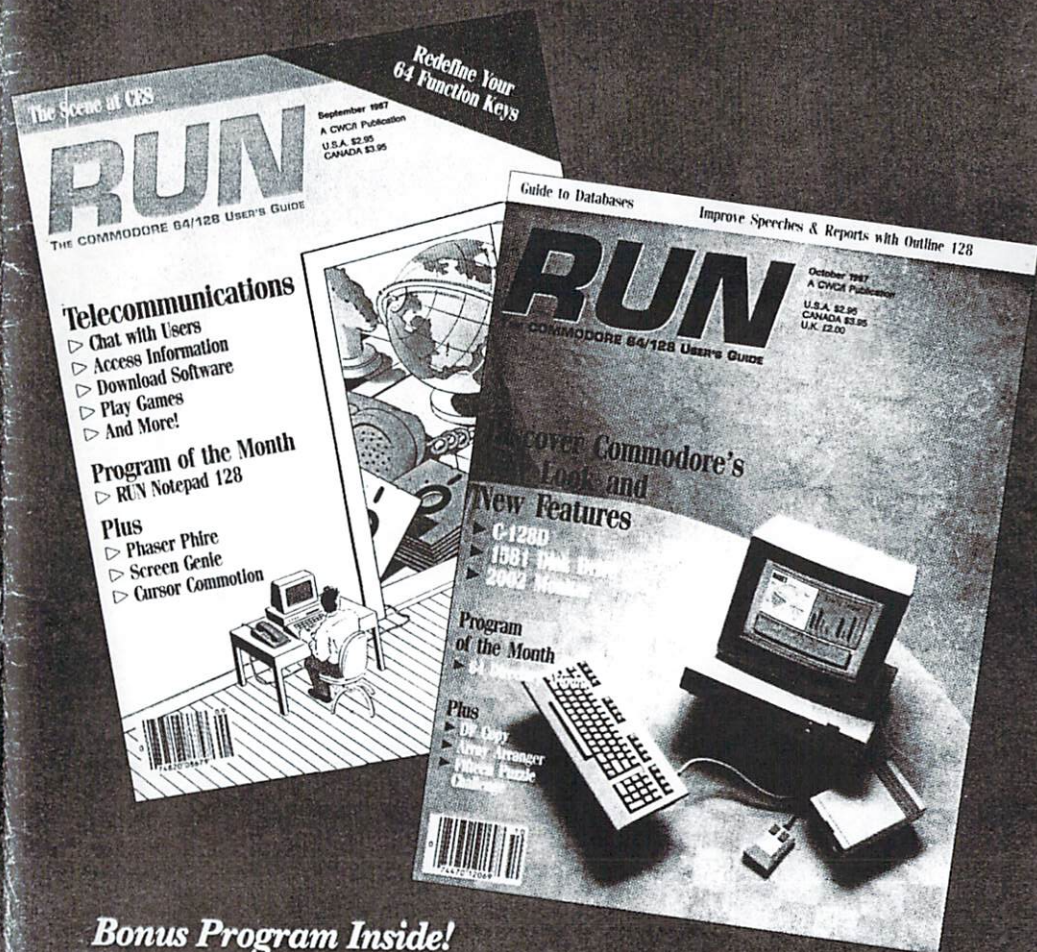


September/October 1987 Edition

RE_≡RUN

RUN Programs on Disk

For the C-64 and C-128



Bonus Program Inside!

Introduction

September-October '87 ReRUN

Before I begin describing the programs found on this ReRUN disk, let me tell you about some features unique to this edition. On the flip side of the disk, you'll find both C-64 and C-128 QuantumLink programs that enable you to get connected with that Commodore-specific network.

If you already own, or are planning to buy, a modem and want to get on-line with QuantumLink, just follow the instructions on the enclosed information booklet. If you have questions about QuantumLink, or simply want to learn more about it, just call the toll-free (in the U.S.) QuantumLink customer service hotline at 1-800-392-8200.

Two other noteworthy features of this disk are the September and October Mega-Magic programs, called Cursor Commotion and Talking Data Statements, respectively. Words such as "amazing" and "dazzling" could describe the screen displays created with Cursor Commotion, but those words have lost some of their meaning through overuse.

Rather than try to describe the program's attributes, I've created

a demo program aptly called Cursor Commotion Demo. Load and run it in 64 mode, and you'll witness a display that is both amazing and dazzling. Let your imagination run free with Cursor Commotion, then send me a disk containing a copy of Cursor Commotion demos of your own design.

October's Mega-Magic, Talking Data Statements, is equally impressive. It lets you load in a program, such as *RUN's* 64 Perfect Typist; then it verbally reads the numbers from the data statements with a distinct Southern accent, as though a computerized Jerry Falwell were hollering out numbers.

Even if you don't type in programs containing lots of data statements, the program readily lends itself to a variety of entertainment purposes. Before you can use Talking Data Statements, you have to obtain a copy of Covox's Talking Demo Disk (see article).

But there's more. Many of you are undoubtedly familiar with the Notepad 64 series written by Bob Kodadek and published in

ReRUN in 1986 and 1987. Well, Bob came up with Notepad 128, a sensational program that offers C-128 owners all the great features found in Notepad 64, plus 100% compatibility with RUN Script 128, version 2.40, so you can have Notepad 128 memory resident with RUN Script 128.

Alas, a C-64 game! Phaser Phire pits you against a Klingon battle group, be it your mother-in-law (a ruthless opponent) or the computer (a Klingon of the nastiest order).

C-64 Function Keys redefines the C-64's function keys to behave in much the same manner as those found on the C-128. For instance, after running C-64 Function Keys, pressing F6 runs the program, F7 lists it, and so on.

Rounding out our September lineup is Screen Genie, which automatically determines screen and color RAM addresses and row and column locations from anywhere on the screen. Programmers are sure to appreciate the usefulness of this program.

The October lineup is equally impressive. Topping the list of C-128 breakthroughs is Outline 128, a "thought processing" outline program. It will actually help improve your speeches or reports by letting you organize your thoughts in an outline format.

Pop-up Directory Window offers the unique ability to display

your disk directory in a window that does not disturb the current screen.

Array Arranger, another impressive program for the 64, does just that—arranges your arrays. Try the demo and see what happens.

DFCopy provides yet another way to copy files created with RUN's Datafile series of programs. You define the criteria of the data within the files you want to have copied, and DFCopy does the rest.

To teach youngsters all about money, we've included an educational program—The Money Program—that appeared in the October Resource Center. And just for the fun of it, we also offer a program called Fifteen-Puzzle, which will generate a great deal of competition among friends to see who can solve it in the fewest number of moves.

Finally, we wrap things up as usual by announcing this issue's bonus program—Disk and File Librarian. It catalogs and organizes all of your disk files by the ID number on the disk.

As always, I hope you enjoy this ReRUN as much as I've enjoyed putting it together. Keep those letters coming.

Tim Walsh
Technical Editor
RUN magazine

Directory

Page	Article	Disk Filename	File type
		MENU 128	BASIC
		MENU 64	BASIC
1	128 Notepad	128 NOTEPAD BOOT	BASIC
		128 NOTEPAD DATA	BASIC
		128 NOTEPAD.OBJ	ML
6	Phaser Phire	PHASER PHIRE	BASIC
8	C-64 Function Keys Made Useful	64 FUNCTION KEYS	BASIC
10	Screen Genie	SCREEN GENIE	BASIC
12	Outline 128	OUTLINE 128	BASIC
15	Pop-up Directory Window	DIRECTORY WINDOW	BASIC
17	Array Arranger	ARRAY ARRANGER	BASIC
		ARRAY ARRAN.DEMO	BASIC
20	DFCopy	DF COPY	BASIC
22	The Money Program	MONEY PROGRAM	BASIC
23	Fifteen Puzzle	15 PUZZLE	BASIC
26	Cursor Commotion	CURSOR COMMOTION	BASIC
		CURS. COMM. DEMO	BASIC
27	Talking Data Statements	TALKING DATA ST.	BASIC
28	Disk and File Librarian	DISK LIBRARIAN	BASIC
		CATEGORIES	SEQ
		RERUN	SEQ

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-mode 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 Disks

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

If the program is written in Basic, it is simple to make a copy of the program. Just load the program into your computer following the procedures outlined above, and then save the program back onto a separate disk that has plenty of free space for extra files.

If the program is written in ML, copying is not so simple. You cannot simply load and save an ML program. In this case, you'll need to use a disk-backup utility program, such as the one on your Commodore Test Demo disk.

ReRUN Staff

Technical Editor: *Tim Walsh*

Managing Editor/Production: *Swain Pratt*

Copy Editor: *Peg LePage*

Proofreader: *Harold Bjornsen*

Design and Layout: *Anne Dillon*

Typesetting: *Doreen Means, Beth Krommes, Ken Sutcliffe*

Special Products Director: *Paul Finch*

Special Products Assistants: *Debbie Bourgault, Robyn Johnson*

Direct Marketing Coordinator: *Debbie Walsh*

128 Notepad

By Bob Kodadek

RUN It Right

C-128

The 128 Notepad program is a commercial-quality desktop accessory for your C-128 that provides many functions. Above all, it offers a text window for recording and recalling programming and other notes, but it also includes a digital time display; a menu for viewing the disk directory and for saving, loading and printing your Notepad notes; and a DOS command center for file maintenance and other disk operations.

Once 128 Notepad is activated, you can interrupt the program you're running, open the window and enter the text editor by pressing control/W. In turn, control/V exits 128 Notepad and returns you to the original program, where the screen display will be exactly the same as you left it, right down to the cursor location.

Files produced with the 64 Notepad Update from last Jan-

uary's issue of *RUN* are compatible with 128 Notepad in the C-128's 80-Column mode.

The 128 Notepad program is written entirely in machine language, resides completely in RAM and remains transparent to most other programs. You can use it in Direct mode as a stand-alone application, with the C-128's machine language monitor or with any Basic or machine language program, including RUN Script 128 Version 2.40 (*RUN*, December 1986, and ReRUN Productivity Pak II) and the Time-DOS 64/128 date-and-time-stamp utility from Family Software. When 128 Notepad is operating with Time-DOS 128 and a Basic program, you'll have three programs interfacing at once on your C-128!

The disk drive's error channel is continually monitored while 128 Notepad is running, and its status is displayed after every operation.

LOADING

The 128 Notepad program ac-

tually consists of two listings: Boot 128 Notepad and ML File Generator. A fast-loading binary file appears in the disk directory as 128 NOTEPAD.OBJ. To activate it, you must run 128 Notepad Boot. The boot program loads the object file and then asks you to enter the current time in hours and minutes. If you press the return key without a numeric entry at these two prompts, the TOD (time-of-day) clock will be set to zero, so it functions as a timer.

If you have Time-DOS 128 in memory, the time is set automatically, and all you need to do is activate the object file. If you own a 1571 disk drive, type BOOT "128 NOTEPAD.OBJ" and press return; if your drive is a 1541 or compatible, type BLOAD"128 NOTE PAD.OBJ", press return, type SYS 7181 and press return again.

Owners of a CCSZ clock/calendar card from the Jason-Ranheim Company must permanently set the TOD master to CIA #2 and disable the time tick, in addition to inputting the Boot or BLoad command above. In most instances, I've found it best to turn off the CCSZ when powering up by typing in POKE 165,30:SYS 999. The TOD clock will continue to keep the correct time, even after you disable the card.

Since 128 Notepad relocates

the start of Basic to memory address \$4000, you must always input the New command before loading a Basic program when Notepad is in use. New resets all the Basic pointers.

USING THE NOTEPAD

Once you've activated 128 Notepad, press control/W to open the window and access the text editor. As the window opens, it will display a Help screen. To clear the window and go on to typing in your notes, press the shifted clear-home key.

All of the C-128's screen-editing features, including the Escape command sequences, function within the window. Insert/delete, the cursor controls and the extended keyboard also operate normally, effectively making 128 Notepad a mini word processor. The program provides 562 bytes of text storage, consisting of 14 lines of 38 spaces each and a 15th line of 30 spaces. The lower-right corner of the window always displays the current time.

When you're done using the window, close it by pressing control/V from the text editor. If you want to leave 128 Notepad altogether, press run-stop/restore. You can reactivate it by inputting SYS 7181.

If you ever want to open the Notepad window while the screen is scrolling, you *must* first stop the

scrolling with the no-scroll key or control/S, or the registers in the 128's 8563 80-column video chip will be confused. If it should become necessary to reset the chip, close the Notepad window and press run-stop/restore.

THE OPTION MENU

To access the 128 Notepad option menu, press control/D while the window is open. The menu choices are listed in Table 1.

Pressing D displays the disk directory. While the directory is scrolling, you can pause it with the no-scroll key or halt it altogether with the stop key.

When you press S to save or L to load a Notepad file, the prompt SAVE:N. or LOAD:N. will appear. The N. is a prefix that 128 Notepad automatically adds to the names of its files. The program's custom input routine will accept any filename that's no longer than 14 characters. To abort a save or load, press the return key without entering a filename, or press the stop key.

When you press P to print out a window, only the Notepad border and its text are output, so it's easy to fit several hard copies on a single sheet of paper.

Pressing X accesses disk operations, including copying, renaming and scratching files; initializing and resetting the drive; and validating and formatting

disks. When the prompt for the command string appears, respond with the standard format for the command you want. For example, to rename a file, you'd input R0:newname=oldname, and to scratch a file, you'd input S0:filename. Filenames must include the N. prefix.

If you own a 1571 drive, you can try sending User 0 commands from the Notepad window. These commands aren't mentioned in the user's manual for the drive, but they're quite useful. A list of some of the more interesting ones appears in Table 2. The format for User 0 commands is U0>xx, where xx is the operation you want.

The last command listed in Table 2 will keep your drive head from banging. The x is the number of read attempts in the drive's zero page location, \$6A. Try U0>R1.

The last choice in the option menu is Q, which exits from the menu and takes you back to the text editor. If you want to go further and close the window, press control/V.

THE NOTEPAD AND RUN SCRIPT

To use 128 Notepad with RUN Script 128, first load and run the Notepad boot program, then load and run the RUN Script 128 boot program. *Never press the*

restore key with the window open when you're using the Notepad with RUN Script 128. If you do, the computer will lock up, because RUN Script alters the NMI (nonmaskable interrupt) vector, and you'll have to reset the computer.

After you've activated 128 Notepad, the interrupt waits for a control/W. When it detects one, a flag diverts all future interrupts, and the current screen display (including attribute memory) is transferred to an alternate location at address \$1000 in the 8563 chip's RAM. This chip comes with its own block-move routine, so the process is easy.

Once 128 Notepad has saved the contents of the zero page and the 8563's VDC (video display controller) registers, it switches in an alternate screen that exactly duplicates the original. The Notepad window will appear in this duplicate. When the alternate screen is in place, the program clears the interrupt-disable bit. Specific Notepad sub-routines perform all operations on the new screen.

Once you close the window, the program switches back to the original screen by reinstating the contents of the zero page and the VDC registers that were stored. It then clears the flag and jumps to location \$FF33 to return control to Basic or to the pro-

gram that's in memory.

The official *Commodore 128 Programmer's Reference Guide* contains almost all the information you'll need to program the 8563 chip. However, it fails to mention that free RAM at \$1000-\$1FFF can hold an alternate screen and machine language routines for reading and writing to the 8563's VDC registers from Basic.

These ML routines exist in the screen editor ROM and appear in bank 15 at \$CDCC and \$CDDA, respectively.

To write to a 8563 VDC register from Basic 7.0, use the format `SYS DEC("CDCC"), A,X`. The X register holds the number of the VDC register to be accessed, while the A register contains the value to be stored in it. For example, to change to the alternate starting address of display RAM, write the value 16 to VDC register 12 by using the code `A = 16:X = 12:SYS DEC("CDCC"),A,X`.

To read a register, load the X register with the number of the VDC register to be read from and call the routine at \$CDDA. The routine will return to Basic with the value in the A register. To change the value into a Basic variable, use the RREG command. For example, to read VDC register 12 in Basic, the code would be `X = 12:SYS DEC`

("CDDA"),A,X:RREG A. The variable A will equal the value found—in this case, 16.

RUN readers have been en-

thusiastic about the 64 Notepad program (September 1986 issue). I think you'll be just as pleased with 128 Notepad. ■

Phaser Phire

By Robert Cook

RUN It Right

C-64

Those despicable Klingons are attacking again, and you, Commander of the Starfleet, must locate and destroy their space-ships before they destroy yours. The stakes are high: If you fail, the Klingons will invade the Federation of Planets and bring it to its knees. The forces are arrayed on the display screen, and Phaser Phire begins! It's played much like the popular board game, Battleship.

When you run the program, there'll be a wait while some custom characters and a sprite are read in. If you're using a C-128 in 64 mode, you can speed up initialization by deleting the REMs in lines 106 and 112. This will activate Pokes that will, in turn, disable the screen update and reduce the set-up time considerably.

The game display includes two side-by-side grids, one for the Starfleet and one for the Klingons. Below each grid is a list

of the four ships in that fleet. The length of each ship, in grid squares, is represented by a line of dashes after the ship's name—one dash per square.

To begin play, you and the Klingon commander must position your fleets on your respective grids. The locations of ships must be secret, so, if your opponent is a person, no peeking.

To place a ship, use your joystick or the £, ↑, = and * keys to move the cursor to the square where you want one end of the ship to be and press the fire-button or return key. Then push the joystick to the right or press the ↑ key to place the ship horizontally to the right of that square; or push the joystick down or press the = key to place it vertically down from the square. Phaser Phire won't let you place one ship on top of another or across a border.

Once the battle lines are drawn, you and the Klingons begin firing phasers at each others' fleets. To fire, move your cursor, which is now on the Klingon grid,

to where you think one of their ships might be, then hit the joystick fire-button or the return key. If you're lucky, you'll hit a Klingon ship, but the first hit won't destroy it. To do that, you must use subsequent turns to hit every square the ship occupies. To win, you must destroy all the Klingon ships before they destroy yours.

Each time you fire, the square you hit gets filled in with a square dot. Each time you hit a ship, a dash after its name in the list is replaced by an asterisk.

The satellite in each fleet has special significance. If you destroy the Klingons' satellite, all the dots in your grid will disappear, so the Klingons won't know where they've already shot or the locations of your ships they've already discovered. Of course, it works the other way around, too. Strategy hint: If you locate the Klingons' satellite early in the game, note the location and destroy it later, after they've taken more shots.

Ready, aim, phire! ■

C-64 Function Keys Made Useful

By Kenny Lawson

RUN It Right

C-64

When I upgraded to a C-128 after two years of using a C-64, it was amazing how quickly I came to depend on the 128's preprogrammed function keys. Having single keypresses to input mundane commands like Load and List has really enhanced my creativity in programming, because they interfere very little with my train of thought. Why, I wondered, can't the C-64 have this same feature?

Well, the truth is that it can! My C-64 Function Keys program redefines the 64's function keys to behave much as the C-128's function keys in their default definitions.

When C-64 Function Keys is first run, the Data statements are poked into memory, and then a list of the key functions is displayed on the screen.

THE FUNCTIONS

Here's a brief description of what each key does:

F1: toggles between the normal text screen and a previously defined hi-res bit-mapped screen.

F2: loads a file. First, it displays the word LOAD, followed by a double quotation mark. When you add the name of the file you want, a close quote and a ,8 (or ,8,1 for machine language programs), and then press return, the file loads as usual.

F3: calls up the directory of the disk in the disk drive, without disturbing any program in memory.

F4 and F8: cycle through the 16 available colors for the border and background of the screen display. I chose to implement these screen-color options, because, for one reason, the C-64 lacks a machine language monitor like the one F8 accesses on the C-128. Also, the 64's hi-res

screen is more movable than the 128's, so a function similar to the 128's F4 screen-clear might erase the wrong section of memory on the 64. Of course, it's also much easier to press individual keys than to keep typing POKE 53280,X and POKE 53281,X in the attempt to find a pleasing color combination.

F5: saves a file with a procedure similar to that for the F2 load above. First, the word SAVE and a double quotation mark appear. Then you add the name you want the file to have, a closing quote and a ,8, and press return to save the file to disk.

F6: runs the program currently in the 64's memory.

F7: lists program in memory.

PROGRAM PARTICULARS

C-64 Function Keys resides in memory locations 52772-53242, leaving 3.5K free starting at 49152 and making the program compatible with many other

wedge-type programs. Unfortunately, it isn't compatible with the DOS 5.1 DOS Wedge, because they occupy the same area, and C-64 Function Keys is difficult to relocate.

Pressing run-stop/restore disables the program by restoring the original IRQ vectors. However, a SYS 52800 will enable it again. On the C-128, SYS 52800 even works when you've gone to 128 mode with a reset and returned to 64 mode with a reset or a command.

The C-64 Function Keys program is meant to be used in Immediate mode. If you use it with a program that includes Get loops (especially loops that look for a function key press), you'd be wise to press run-stop/restore before running the other program.

I've identified the beginning of each function-key routine in the listing with a REM statement, so you can study how it works and alter it if you like. ■

Screen Genie

By John Ryan

RUN It Right

C-64

You've just finished writing the last line of your all-time greatest graphics game, and it's time to add some character graphics. Let's see... The playfield begins at locations 1104 and 1134. How many rows apart is that? One, two, three... Wait. Hmmm. That's not right... Maybe it's one, two, three, four across.

Sound familiar? Well, the cure is Screen Genie, a utility I wrote to eliminate the tedious pencil and graph paper calculations that are traditionally needed to determine screen and color addresses for graphics layouts. It'll be especially valuable to those who want to use X and Y coordinates for "homing" movable characters to a desired location.

Screen Genie superimposes a solid cursor on your screen display, then reports information about the graphics character under the cursor, including its screen-RAM and color-RAM ad-

resses and its row and column positions. Screen Genie won't disrupt your display in any way, and you can move the cursor wherever you want.

UNCORKING THE BOTTLE

Load and run Screen Genie to install it in memory, where it'll be ready when you need it. It can be activated from Direct mode by typing in the SYS 49152 command, or from Program mode by including the SYS 49152 command in your program listing. Either way, an information line containing the following labels will appear at the top of your screen display:

SCN= CLR= X= Y= CHAR=

SCN= is the current screen-RAM address of the cursor, and CLR= is its corresponding color-RAM address; X= and Y= are the cursor's row and column positions, ranging from 0 to 24 and 0 to 39, respectively; and CHAR= is the screen-display code of the graphics character under the cursor.

The colors in the information line will be the same as the colors in your screen display, and if your display is done in reverse characters, the characters in the information line will be in reverse, also.

Only the cursor and return keys are active while Screen Genie is in use. As you move the cursor around with the cursor keys, the information line will be updated continually. Keep a

sheet of paper and a pencil handy, so you can copy down the addresses you need for your screen layout.

To exit Screen Genie and return to the mode you were in before, tap the return key. The screen will revert to its former state, and then, if Screen Genie was activated from Program mode, execution will continue with the statement following the SYS command. ■

Outline 128

By Gene Towe

RUN It Right

C-128

Do you have trouble organizing your thoughts? Or perhaps you finish preparing a long outline and then realize the third topic should actually be a subtopic of the second. What you need is an outline processor, a program that's a cross between a word processor and a database manager, with a few new twists thrown in. An outline processor, sometimes known as a thought processor, lets you enter a list of topics, rearrange and edit them, and, finally, print the outline in the traditional hierarchical form.

I've written just such a program in Basic 7.0 for the C-128 in 80-Column mode. Called Outline 128, it offers the same major features found in commercial outline processors for the most popular computers, including the C-64, the IBM PC and the Macintosh.

CREATING AN OUTLINE

To use Outline 128, load the program and run it. When the title screen appears, press the return key to go to the main screen, which is divided into two parts. The upper box displays the current status of the program, and the remainder displays the level of the outline being worked on. The title of the current level is displayed, followed by each of its subtopics.

Up to 15 subtopics can be entered for a topic, with a maximum of 1000 topics for all levels. If a topic has less than 15 subtopics, one additional subtopic number is displayed so you can add to that outline level.

Outline 128 offers several editing keys. Cursor-up and cursor-down move the topic pointer at the left of the topics or subtopics to the level you want to edit. The F1 key advances to the level below the current one, the F3 key returns to the parent of the level displayed, F5 inserts a new topic or subtopic immediately preced-

ing the current one, and F7 deletes the current topic or subtopic. The escape key displays the command menu, which I'll explain later.

You enter and edit topics with the usual typing keys, including the letters, numbers, punctuation marks and the return key. If you're creating a topic or subtopic, your input appears in the upper box. If the topic or subtopic already contains text, the text is displayed in the upper box for you to edit. To edit the text, start by pressing the return key. If you're entering new text, start with any typing key other than the return key.

OUTLINE 128 COMMANDS

As mentioned earlier, pressing the escape key displays a list of Outline 128 commands, which appear in the upper box of the main screen. To activate a command, press its first letter. To return to Edit mode, press any other key.

Load Outline loads a previously saved outline from disk. You must enter the filename of the outline and the device number of the drive the disk is in. If you have a dual drive, you can specify the drive by prefacing the filename with the drive number.

Save Outline saves the outline you're working on to disk. This command asks for a file-

name and device number, just as the load command does. Be sure to have a formatted work disk in the drive and be careful not to accidentally enter a filename you've already used, because the previous file will be overwritten during the save.

Disk Directory displays the directory of the disk in the drive you specify. To pause the display, use the no-scroll key.

Print Outline prints an outline to the screen, the printer or the disk. It's printed to disk as a sequential file. You must enter a title to appear at the top of the outline, and you can choose to start the printing at the top or at the level that's currently displayed. The outline can be printed with or without topic numbers, and you can specify any indentation up to 65 characters.

If you call for an indentation that exceeds 65 characters, the affected topics and subtopics won't print out, but will be represented by three periods. You can get them to print by moving to the desired level and printing from there. Topics and subtopics automatically word-wrap as they're printed.

Clear Memory clears the current outline from memory.

Move Topic moves a topic or a subtopic and its subtopics from one point to another within the outline. The source for the block

move is the topic pointed at when you select the command. Use the cursor keys and F1 and F3 to place the cursor just below where you want the block to go, then press return to move the

block. Pressing the escape key aborts the command.

Quit exits the program. The character color is set to white and the function keys are returned to their default values. ■

Pop-up Directory Window

By David Morris

RUN It Right

C-64

If you ever need to access a disk directory on your C-64 while writing or running a program, then this handy utility is for you. I know what you're probably thinking—you've already seen scores of "directory readers" in magazines and elsewhere. Well, this program is different, because it leaves the screen "behind" the window intact.

Directory Window is a machine language, interrupt-driven program that resides at memory address \$C000, where it's safe from everything but programs that occupy the same location. Furthermore, it's compatible with most machine language programs that don't use this area of memory.

After you run Directory Window, the only way it can be disabled is with SYS 49152 or

by resetting the computer. Run-stop/restore works as usual, except that it doesn't disable Directory Window or reset the screen background, border and text colors to their default values.

After Directory Window reads its data into memory, you're prompted to specify a solid or clear window. If you choose a solid window, you can pick its color; if you choose a clear window, you can select the color of the text within the window. These settings can be changed at any time. The program then displays the SYS command to activate and deactivate Directory Window.

After activating it, make sure a disk is in the drive, then press control/D. A window will appear in the center of the screen and display the disk directory, ten lines at a time. If you have more than ten files on the disk, press the space bar to display the next ten. To close the window, press the run-stop key. The window will

disappear, and the screen behind the window will be restored as it was, including the original colors and cursor position.

If you want to specify the background, border and text colors for the screen behind the window, set them in the usual manner. As I indicated, run-stop/restore won't change those colors to the defaults.

You can access Directory Window while another program is running. The other program pauses while the directory is read and

displayed; then, when you close the window, it continues where it left off.

Note: Some machine language programs, particularly those that alter the vectors at \$0300-\$030B, may not be compatible with Directory Window. Also, Directory Window is incompatible with the Epyx Fast Load cartridge.

There you have it—a fast, easy-to-use program for displaying your disk directory at the touch of a key. ■

Array Arranger

By Jonathan Holuta

RUN It Right

C-64

If you've ever written a program that uses first and last names, you may have been faced with some dilemmas. Should you store the names in an array? Should the first and last names be separated? Should the last name be first? What if you wanted to sort them by last name, but print with the first name first?

Array Arranger is a powerful utility program that will help solve such problems by letting you store the first and last names, in either order, in one element of an array, and then "switch" the order at your command. It's a machine language program that borrows no memory from Basic, and it uses no intermediate strings that can cause "garbage collection" delays.

SEQUENCE SWITCHES

Let's say you've stored a list of 100 names in an array called

N\$(100), with the last name first for easy sorting. Here are a few sample elements:

N\$(1) = "JONES WILLIAM"

N\$(2) = "SMITH ROBERT"

N\$(3) = "JOHNSON CLIFFORD"

...

N\$(99) = "JEFFERSON ERNESTINE"

N\$(100) = "THOMPSON HARLAN"

You'd like to print them out with the first name first, but you don't want to change the array. All you have to do is include the following in your program:

```
SYS 49152,A$,N$(X)
```

and then print A\$. If X = 1, variable A\$ will hold the switched name and then print WILLIAM JONES. You can use a loop to print the entire list, first name first, without having to change the names in the N\$ array.

CASE AND LENGTH SWITCHES

Now, let's suppose you want to print the names in upper- and lowercase, but they're stored in uppercase only. This time include:

SYS 49159,A\$,N\$(X)

and print A\$. Like magic, A\$ now contains the switched name with the first and last names capitalized (Harlan Thompson, if X=100). Any middle initials will also be capitalized.

This function of the program is handy for mailing lists, databases, customer lists, and the like. The program will also shorten names for you, if your printing format only allows a certain number of characters in a column. The default length is 25. If you have space for only 17 characters, include:

POKE 49411,(L+1)

where L is the length you don't want to exceed—in this case, 17. Then include:

SYS 49159,A\$,N\$(X)

Variable A\$ will now be E. Jefferson, if X=99, but N\$(99) will remain JEFFERSON ERNESTINE. If you don't want the capitalization feature, include the POKE 49411,(L+1), but change the SYS command to SYS 49152, A\$,N\$(X). The name will still be switched and shortened, but without the case change.

ARRAY SWITCHES

The real power of this program lies in its third type of switching. Let's say your list of names is stored first name first, and you

want to call a routine to sort by last name. The entire array must be switched before calling the sort routine. This program can do it. Just include:

SYS 49730,E,N\$(0)

where E is the number of elements in the array. Before you can blink, the entire array will be switched, and you can call your sort routine. To switch the array back, just use another SYS 49730,E,N\$(0), and instantly the entire array will revert to its former state.

This is a powerful command, because it actually changes the names in memory. If your name strings are read from a disk file or typed in as user input, the actual text is stored from the top of memory down (which is usually transparent to the user). If your strings are read from Data statements, the actual text is stored in the program. By entering the SYS 49730,E,N\$(0) command after an array of strings is read from Data statements and then listing the program, you can see that the actual program text has been switched. A line that was:

500 DATA JONES WILLIAM,SMITH
ROBERT,JOHNSON CLIFFORD

will now be:

500 DATA WILLIAM JONES,ROBERT
SMITH,CLIFFORD JOHNSON

PROGRAM DESCRIPTION

Array Arranger 1 pokes the numbers in the Data statements directly into memory. Be sure to save the listing before you run it, because, if the checksum is correct, the Basic portion will be erased, leaving only the machine language in memory. The machine language remains there until the computer is turned off or another machine language

program is loaded over it.

Array Arranger 2, written entirely in Basic, demonstrates all of the program's functions. Load and run it after running Array Arranger 1.

I hope you enjoy using the Array Arranger. I wrote it for some personal applications, but, since I've learned so many things from *RUN*, I wanted to share it with other readers. ■

DFCopy

By Paul Turner



C64

RUN's Datafile programs are among the most powerful and easiest-to-use database management programs available.

In using Datafile, I've often needed special-purpose files composed of only some of the records in my Datafile files—for instance, a file of zip codes copied from an address file. So I wrote my own program, called DFCopy, to create them. I intend it to work in conjunction with the DF series of programs to provide additional data-management flexibility.

As I indicated, DFCopy creates a new file by copying records from a Datafile file that already exists. It copies only those records from the source file that meet certain specifications you input. It doesn't alter the source file in any way, nor does it alter the records it copies. It just moves them.

The beauty of DFCopy is that it eliminates any need to alter a

source file's structure to tailor its data to special needs or to reenter data that you've already typed into the source file.

DFCOPYING

To use DFCopy, first enter `LOAD"DFCOPY",8` and then `RUN`. The first screen that appears is an introduction. Next, the program asks for the name of the Datafile file you want to use as a source from which to copy data. Your response doesn't need to include the special characters, `DF`, that precede the filename. If you press return without entering a name, the program goes back to the introductory screen.

Assuming you have entered the name of a source file, the computer reads that file into memory and displays a list of its field names and lengths, then asks you to specify the field you want to work with. A return without specifying a field brings you back to the screen for naming the source file.

Assuming you want to pro-

ceed, enter the field number. Then a list of logical operators will appear. DFCopy will use the operators that you specify for identifying data in the source file to be copied to the new file.

You'll find the operators listed in Table 1, along with their results for a sample run of DFCopy. The table assumes that the source file contains addresses and that its zip-code fields are the ones to be copied. It assumes further that all the zip codes in the source file are a full five digits long.

Back to the operation screen, enter the numbers for the logical operators you want to use (or zero to return to the field-selection screen). Any operators you choose will be listed at the top of the screen for reference.

Then you'll be asked to input the constant data—that is, the data strings DFCopy should search for in the source fields. The maximum length for each string is XXX. A return with no input of constant data brings you back to the field-selection display.

Refer again to the table, and note that the data string you input can be shorter than the length of the fields it'll be compared to. Just fill out the string with spaces. (That's what the # symbols in the table represent.) I coded the comparison this way to provide the greatest amount

of flexibility in record selection.

After you've specified the operators and data strings you want to use, the program lists them for your final okay. An R for reject retrieves the operations-available screen; an A for accept starts construction of your new file. If DFCopy doesn't find any fields that meet your criteria, a message to that effect appears and the program returns to the field-selection screen.

If DFCopy does find some fields that meet your criteria, it asks for a name for the new file. A return with no name input brings back the field-selection screen. If you do name the file, DFCopy writes the file to disk. Then the field-selection screen comes back, giving you an opportunity to create another special file from the same source file.

Two special files can be built from a source file before you must read the source file into memory again. This limit lessens the chance of the computer going into "garbage collection."

DATAFILE 3.6

Those who've upgraded to Datafile 3.6 (see *RUN*, January 1987) and want to use the DOS 5.1 Wedge should remove the REM statements in lines 110, 120 and 300 of DFCopy. Also, be sure that the wedge is on the same work disk as DFCopy. ■

The Money Program

By Michael McKellips

RUN It Right

C-64

The October 1987 Resource Center contains a program, called Money, that works on the C-64 and was written and donated by Michael McKellips of the Lexington School, 1130 W. Co. Rd. B, Roseville, Minnesota. McKellips has been using C-64s in his special-education class-

room for the past four years, but he's had difficulty locating software for moderately retarded students. To solve this problem, he started writing his own.

His Money program provides practice in recognizing coins and counting change. It's appropriate for elementary-age students and foreign-born adults who are trying to learn our currency, as well as for its intended audience, the moderately retarded. ■

Fifteen Puzzle

By Mike Buckley

RUN It Right

C-64; mouse

If you're always itching for a new challenge, you'll want to try Fifteen Puzzle, my mouse-driven C-64 version of Sam Loyd's famous sliding-block puzzle. Later, after you've solved the puzzle a few times, you'll have fun giving it a special twist and trying it on unsuspecting friends and relatives.

When you run Fifteen Puzzle, two arrays appear on the screen, each consisting of 16 blocks arranged in four rows and four columns. Fifteen of the blocks are numbered, 1-15, and the other one is blank.

The small array on the right is the "target," with the numbers appearing in proper numerical order. In the large array on the left, the numbers are scrambled. The object of the game is to "slide" the blocks in the scrambled array up and down and left and right until they're in the same order as in the target array, and to move as few blocks as pos-

sible in the process. Each time you move a block, your score increases.

Here's how to play. Move the mouse so the arrow points to any number in the scrambled array that's in the same row or column as the empty space. Then click the left joystick button, and the number will slide toward the space, shoving any numbers in between with it. The empty space will end up under the pointer.

Keep moving numbers until the two arrays are alike, at which point the game is over. You can slide as many as three numbers with one click, but try to move as few as possible, because your score will increase by one for each number you move. Your current score and your best (lowest) previous score are displayed as you work. When you've finished the puzzle, you can click the left button to play again or the right button to quit.

Fifteen Puzzle is written in Basic, with a machine language subroutine to take care of the

mouse and its pointer. The machine language code and sprite parameters are poked into memory from Data statements, so be sure to heed the usual warning about saving the program before running it.

LOYD'S HOAX

With the program as written, each game starts with a different scrambled array, and each puzzle is solvable. However, if you change line 390 so it reads `FORL=GTO62` instead of `FORL=1TO62`, the first puzzle will be solvable and the second will be impossible.

Demonstrate Fifteen Puzzle to a friend, then play the first game and challenge him or her to get a better score on the second. No matter how high (poor) your score is, your friend won't be able to beat it!

If this seems unfair, consider the historical precedent. Around 1872, Sam Loyd, the great American puzzlist, drove the country crazy with his Fourteen-Fifteen puzzle. The antecedent to mine, it also was a square box containing 15 numbered blocks and one blank. The numbered blocks were in numerical order except for the 14 and 15, which were reversed. The object was to slide the blocks around to reach a configuration similar to the initial one, but with the 14 and 15 in proper sequence.

The craze died down when mathematicians proved that the task was impossible.

To use a simpler example, consider a smaller box, with only three numbered blocks. The 24 possible arrangements of the blocks fall into two distinct sets: one in which the numbers run clockwise and one in which they run counterclockwise. It's evident that you can't get from one set to the other without taking the blocks out of the box. In fact, you can't get from one configuration to another if they differ by an interchange of two blocks or any odd number of such exchanges. Loyd's puzzle, although it has almost 21 trillion possible arrangements of the blocks, falls in this category.

An even number of swaps maintains the "parity" of the configuration. In my program, the 62 switches performed in lines 390-400 guarantee not only a solution, but a better-than-50-percent chance that no number in the scrambled array will start off in its target position. The programming change I suggested results in only 61 switches, making the second puzzle unsolvable.

PLAY THIS GAME NOW

If you're the kind of person who likes to tackle six impossible jobs before breakfast, give some

thought to this puzzle array:

P L A Y
T H I S
G A M E
N W O

The previous discussion would suggest that it's hopeless to try to correct the transposition in the bottom line. If you make the following changes to the program,

they will enable you to research the matter:

```
310 DIMA$(4,4),T$(4,4):T$="PLAY  
THISGAMENOW "
```

```
390 :
```

```
400 A$="PLAYTHISGAMENWO "
```

If you do achieve the impossible, write to me care of *RUN* and let me know how many moves it took and what they were. ■

Cursor Commotion

By Mark Joerger

RUN It Right

C64

Not only does this little program, QuickDoc, provide a quick and easy way to type in and save short messages, but you can animate them, too.

Once you run QuickDoc, you can enter text, graphics, color changes, cursor moves, even inserts and deletes, and save them all as part of the program. Just enter RUN 25 and start typing. Change colors, type over old text, make frames or boxes in Reverse mode, move text around, use Commodore-key graphics. Press F7 when you're finished, and watch the action.

Then, when you're ready to save your document, place a formatted disk in the disk drive, press the run-stop key and give QuickDoc a new descriptive filename. When your friend or relative loads and runs that version of QuickDoc, your message will

be printed to the screen, one character at a time, just as you entered it.

Be careful when you make changes to QuickDoc, though. If you make the program too long, there'll be interference to the text you enter. Safe changes include the screen color Poke in line 50, the border color Pokes in lines 10 and 25, the cursor character (C\$) in line 50 and the printing speed (currently 52) in line 20.

The Dimension statement in line 25 provides room for 10,000 characters of information. You can increase this amount if you need to write a longer document.

In addition to leaving notes for others, you might want to use QuickDoc for on-disk documentation of your programs, lo-res graphics art, tutorials, disk letters, an on-screen home-message board, or any number of other purposes. Your imagination is the limit! Experiment and have a good time—that's what home computers are all about, right? ■

Talking Data Statements

By Eugene Cozzi

RUN It Right

C-64; Covox Talking Demo disk

With this program and a copy of Covox's Talking Demo Disk, your computer will "speak" a program's Data statements to you as you check its listing. (To get a copy of the Talking Demo Disk, send \$5 along with your order to Covox, Inc., 675-D Conger St., Eugene, OR 97402. The disk contains several other demo programs and comes with a 28-page booklet.)

Follow these steps:

1. Copy the VMX and Cash files from the demo disk to your work disk.

2. Load the VMX file with LOAD "VMX",8,1, then type in the SYS 49152:NEW command.

3. Load TALKING DATA ST. and save it on your work disk with the filename Talker.

4. Load the program containing the Data statements you want Talker to recite, then, in Direct mode, enter:

```
M = PEEK(45) + PEEK(46) * 256 - 2:  
POKE44,M/256:POKE43,M - PEEK  
(44)*256
```

Now load Talker and type in

```
POKE43,1:POKE44,8
```

5. Load in the Cash file, using the command FIND "CASH",8.

6. Type in RUN 60000—you'll hear your computer reciting the Data statements to you. Note that Talker only recites numbers from 0 to 999 and the last three digits in larger numbers. Negative numbers end the program.

7. Now, whenever you want your computer to read Data statements, just load the VMX program (step 2), and then follow steps 4, 5, 6 and 7. ■

Disk and File Librarian

By Jay Machielse

RUN It Right

C-64; C-128

This ReRUN bonus program, Disk and File Librarian, will create a catalog of your disks and files and then tell you on which disk a specific program is located. It will also print, either to the screen or to your printer, an alphabetical list of all the files on all your disks, with disk IDs, or an alphabetical list of all the files on a particular disk. If you use the program in the C-128's 80-Column mode, the Fast command will speed up operation.

To use Disk and File Librarian, load it and then place a formatted work disk in the drive. Next, type in RUN to write a sequential file called Categories to the work disk. This file will hold your library of filenames, with the names divided into categories that you input. For instance, I use "C-64" and "C-128" for my

Commodore 64 and 128 programs, respectively.

CREATING AND MODIFYING A LIST

When the Categories file is on your work disk, input a category name. Then select either the Category Maintenance option to add another category or delete one, or the Existing Categories option to input filenames into your category or view the filenames already there.

If you're at the Main menu and want to enter filenames, choose the Add/Delete Files option. A submenu will appear that offers the following ways to input filenames:

1. Manually, one at a time.
2. Semi-automatically, in which filenames are read off a disk and you can accept or skip each one in turn as it's displayed.
3. Automatically, with all the filenames from a disk input at once.

Each filename must include a

disk ID, which is read from the disk, and since Disk and File Librarian uses these IDs in organizing and finding filenames, it's imperative that the ID for each disk be unique. Also note that the program won't accept duplicate filenames that have the same disk ID.

You can delete filenames either individually or all on a disk at once. The latter option is useful when you're reformatting a disk.

When you return to the Main menu, your altered list of filenames is automatically rewritten to the current category file.

The View All File Names option in the Main menu lists your filenames to the screen alphabetically in two columns of 20 filenames each or to your printer in three columns of 60 filenames each. On-screen, you can rapidly page through the filenames.

SEARCHING FOR A FILE

To locate a specific file, use the Search for a File Name option. If you're unsure of the spelling of the filename, input it as best you can. Then, if the program can't find what you typed in, it will drop the last character and try again. It will keep dropping characters until it finds a match or runs out of characters. In the latter case, it will tell you there's no match.

When the program does locate a filename you want, it lists the name at the top of the right column. It also lets you page forward and backward through the file.

The checks for duplicate names and a match in a search use the high/low method. First, the filename in question is checked against the middle filename in the category list to see if it comes before or after the middle filename. This is possible because the list is alphabetical, and it immediately eliminates half of the list from further searching. Then the program checks the filename against the middle filename in the half of the list that's left, thus eliminating another quarter. This process continues until a match is found or the list is exhausted. This technique is much faster than proceeding through the list one filename at a time.

As the program is written, each category can hold up to 775 filenames. However, you can change this number by changing the value of variable MN in line 2450. Just remember that the higher the value you dimension the variable at, the slower the program will be.

When you choose the Add/Delete Files option, the program automatically goes into garbage collection. On a C-64, the time

required for this becomes quite noticeable as the number of names in the current category gets into the 700s. On a C-128, the time remains negligible.

HELPFUL HINTS

If you're running Disk and File Librarian on a C-128 in 80-Column mode, omit the first REM statement in lines 2480, 2642

and 5245 and delete line 5250. If you have a 1541 disk drive, change line 2350 to RETURN to disable the routine for a 1571 that determines if disks are double- or single-sided and sets the drive accordingly. All the program's routines are modular, and you can use them in developing your own programs. ■

Please send me back issues of ReRUN

___ January/February 1986

___ January/February 1987

___ March/April 1986

___ March/April 1987

___ May/June 1986

___ May/June 1987

___ July/August 1986

___ July/August 1987

___ Productivity Pak II

___ September/October 1986

___ November/December 1986

___ Disk version(s) at \$16.47 each*

** Price includes postage and handling. For foreign air mail, please add U.S. \$1.50 per item. Prepayment only.*

☐ Payment Enclosed

☐ MC

☐ VISA

☐ AE

Card #

Exp. Date

Name

Address

City

State

Zip

Signature

ReRUN • 80 Elm Street • Peterborough, NH • 03458

BEAT THE RUSH!

Please send me:

☐ 1 year (6 issues) for \$69.97

☐ November/December 1987 ReRUN disk for \$16.47.*

**Available in December 1987.*

Includes programs for C-64 and C-128 (in both 64 and 128 modes).

Price includes postage and handling. For foreign air mail, please add U.S. \$1.50 per item and \$25 per subscription. Prepayment only. Subscriptions will start with the current issue.

☐ Payment Enclosed

☐ MC

☐ VISA

☐ AE

Card #

Exp. Date

Name

Address

City

State

Zip

Signature

ReRUN • 80 Elm Street • Peterborough, NH • 03458

13 RUN Programs Included on this Disk:

Notepad for 128 ▶ Function Keys Utility
Screen Utility ▶ Game ▶ Array Arranger
Outline Generator ▶ Money Tutor
Disk Librarian

From the September RUN:

- ▶ 128 Notepad
- ▶ Phaser Phire
- ▶ 64 Function Keys
- ▶ Screen Genie
- ▶ Cursor Commotion

From the October RUN:

- ▶ Outline 128
- ▶ Directory Window
- ▶ Array Arranger
- ▶ DFCopy
- ▶ 15 Puzzle
- ▶ Talking Data Statements
- ▶ Money Program

Bonus Program:

- ▶ Disk Librarian

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:

ReRUN • 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 1987 by CW Communications/Peterborough. Unauthorized duplication is a violation of applicable laws.

©Copyright 1987 CW Communications/Peterborough



CW COMMUNICATIONS/PETERBOROUGH