March/April 1986 Edition

ReRUN

RUN Programs on Disk

*For the C-64 and C-128**

The Commodore C-128/C-64 Home Computing Guide
RUN
U.S.A. $2.95
CANADA $3.95
April 1986 A CW Publication
POWER UP YOUR HOME FOR SECURITY, SAVINGS AND CONVENIENCE
LOAD AND RUN YOUR PROGRAMS IN A SINGLE KEYSTROKE!
ADD FOUR SUPER UTILITIES TO YOUR CP/M LIBRARY
SING ALONG WITH YOUR C-64
PLUS:
—RUN Script, Part II
—Terminal Programs Explained
RUN SCRIPT 64 RUN's POWERFUL NEW WORD PROCESSOR
Commodore C-128/C-64 Home Computing
RUN
128 PRODUCTIVITY
YOUR COMMODORE TO WORK!
March 1986 A CWC
COMPUTING WORKSHOP
Get On-Line...
There!
YOUR
NG SKILLS
RIES ON BASIC
SPREADSHEET
DATABASE
CREATE YOUR OWN K
FOR FAST AND
NUMERIC DATA E
PLUS:
A GALLERY OF COMMODORE CHARACTERS

Plus: Bonus Programs!

*128 mode programs included

Introduction

March-April 86 ReRUN

In this edition of ReRUN, we are implementing a new menu feature that C-64 and C-128 owners will enjoy. Now, by simply running a menu program first, you can get a directory of all the programs on your disk. From there, you just select the program you want, press a key, and it will automatically load and run.

These menus will handle both Basic and machine language programs. Feel free to place our automenu programs (MENU 128 and MENU 64) on your own disks to make your computing activities easier.

This ReRUN edition contains all of the March and April programs from *RUN* magazine, as well as two bonus programs. (In the brief program descriptions below, the article title is in parentheses if it differs from the filename listed in the directory.) Productivity has emerged as a main focus for home computer use, and we are encouraging this with RUN-SCRIPT (RUN Script 64), our new word processor for the C-64.

Command-driven and written in machine language, it boasts 70 different features for creating, modifying, saving, appending and printing out your text files, which may be any type available on the C-64: program, sequential or user. Because of this versatility, you can use RUN Script 64's text files with other word processors and with terminal programs, such as our own Runterm Plus.

For those of you who wish your C-64 had a number pad like the C-128, here it is. KEYPAD 64 (Create Your Own Keypad) turns part of your regular keyboard into a numeric keypad, which you can call in and out of action as needed while typing in Basic program listings.

Add AUTOBOOT 64 (Auto-Boot) to your repertoire of disk utilities. This program lets you add an automatic load and run feature to individual programs. You'll find this comes in handy, for example, with RUN Script 64. You simply prepare BOOT RUN-SCRIPT as an autobooting program. It will save you that extra step of typing RUN each time.

AUTONUMBER 64 (Automatic Line Numbers) adds to your C-64 an automatic line-numbering feature like that of the C-128.

SONGFEST is a revised version of the musical editing and keyboard program that appeared as a bonus on the Winter ReRUN. With it, we include a new song, MR SANDMAN, for your enjoyment.

ML TYPIST 64 (ML Perfect Typist) is our standard machine language entry program for those of you who would like to take a stab at typing in the hexadecimal machine language program listings that we occasionally publish.

A regular feature of ReRUN has been a bonus program. This edition gives you two! METRONOME 64 is perfect for those of you who teach music or play a musical instrument. This program actually changes your C-64 into a metronome that you can see and hear as you practice your instrument.

Our second bonus program is specifically for those looking for C-128 mode software. GRADEBOOK 128 (C-128 Gradebook) is an outstanding teacher's tool. It's actually a customized 80-column spreadsheet that lets you keep track of all of your class rosters, grades and averages. You can also make printouts from this data.

That's all, folks. Enjoy this edition of ReRUN, and I'll see you next time with programs from the May and June issues of *RUN* magazine.

Margaret Morabito
Technical Manager
RUN *magazine*

Directory

1	MENU 128	C-128 MODE	BASIC
3	MENU 64	C-64	BASIC
5	BOOT RUNSCRIPT	C-64	BASIC
*5	RUNSCRIPT	C-64	ML
5	COPY CHAR SET	C-64	BASIC
*5	C64 CHAR SET	C-64	ML
5	RUNSCRIPT PARAMS	C-64	BASIC
16	DEFINE MACROS	C-64	BASIC
21	KEYPAD 64	C-64	BASIC
21	KEYPAD TYPIST 64	C-64	BASIC
23	SONGFEST 64	C-64	BASIC
+23	MR SANDMAN	C-64	SEQUENTIAL FILE
34	AUTOBOOT 64	C-64	BASIC
36	AUTONUMBER 64	C-64	BASIC
38	ML TYPIST 64	C-64	BASIC
£41	METRONOME 64	C-64	BASIC
£44	GRADEBOOK 128	C-128 MODE	BASIC (in 80 columns)

* RUNSCRIPT and C64 CHAR SET are machine language programs that can only be loaded by running BOOT RUNSCRIPT.

+ MR SANDMAN is a song that can only be loaded from within SONGFEST 64.

£ Bonus Programs!

How To Load

space

This ReRUN provides menu programs for both C-64 and C-128 users. C-64 users should type LOAD "MENU64" and RUN. C-128 users need only press the shift and run/stop keys. These menus will display all of the programs on the disk and let you run them with a single key press.

If you don't want to use the menu programs, then follow these guidelines for loading individual programs.

C-64:

To load a C-64 program written in Basic, type:

```
LOAD "PROGRAM NAME",8
```

and then press the return key. The drive will whir while the screen prints LOADING, and then READY, with a flashing cursor beneath. Type RUN and press the return key. The program will then begin.

Machine language (ML) programs will be loaded with LOAD "PROGRAM NAME",8,1. When these occur, they will be clearly marked for you on the directory page.

C-128 in C-64 mode:

All C-64 programs can be run on the C-128 as long as your computer is in C-64 mode. This is accomplished by typing GO64 after powering up your C-128. Answer Y and press the return key in response to the "Are you sure?" prompt. When in C-64 mode, use the instructions above for loading C-64 programs.

C-128 in C-128 mode:

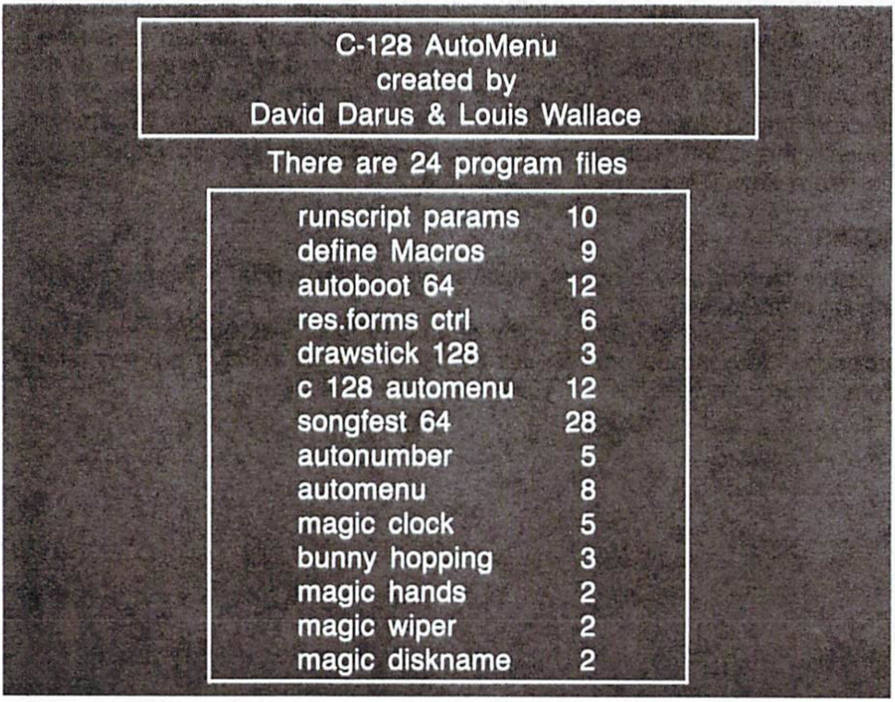
All C-128 mode programs are clearly labeled on the directory page. Your C-128 must be in C-128 mode to run these programs. All programs will work in both 40 and 80 columns, unless they are specifically marked otherwise.

To load a C-128 mode program, just press f2 and type the name of the program that you wish to load. Use the exact name that appears on the directory page. When the program has loaded, a READY prompt will appear. Beneath this is the flashing cursor where you should type RUN and press the return key. The program will then run.

NOTES:

Before loading any program, always refer to the article in this booklet for special instructions.

It is a good idea to make a copy of your ReRUN disk before running the programs. You can accomplish this by using the disk backup programs that Commodore provides for the 1541 and 1571 disk drives.



```
            C-128 AutoMenu
              created by
      David Darus & Louis Wallace

       There are 24 program files

         runscript params    10
         define Macros        9
         autoboot 64         12
         res.forms ctrl       6
         drawstick 128        3
         c 128 automenu      12
         songfest 64         28
         autonumber           5
         automenu             8
         magic clock          5
         bunny hopping        3
         magic hands          2
         magic wiper          2
         magic diskname       2
```

C-128 AutoMenu

By David Darus and Louis Wallace

RUN It Right

C-128 (in 128 mode; 40/80 columns)
1541 or 1571 disk drive

The C-128 AutoMenu is designed to make it easy for you to load and run programs on the C-128. It allows you to quickly and easily look through a disk's available program files and load and run them by simply pressing a key.

You can save C-128 AutoMenu to any of your frequently used disks to save time later. In fact, by using the C-128's autoboot capability, you can set up each C-128 disk so that AutoMenu is automatically booted at powerup. (For a program that allows you to make self-booting program disks, see "Disk Commands on the C-128" in *RUN*'s December 1985 issue.)

When you run or autoboot C-128 AutoMenu, it first checks to see if you are in 40- or 80-column mode, then formats its output accordingly. It will go to the disk for a minute and read in all the names and sizes of the program files on the disk. It will ignore sequential, user and relative files, since they are not executable programs.

Once the program names are in memory, it will list them in a window on the screen, with the program size on the far-right side of the window. You can use your cursor keys to move up or down through the list while highlighting the program name under the cursor. If you wish, you can return to the top of the list by pressing the clr/home key.

Once you've found the program you want, simply press the return key while the cursor is on the program name, and your choice will be loaded and run if it is a C-128-compatible Basic program.

AutoMenu checks the load address of the file you choose; if the address isn't correct for a C-128 Basic program, AutoMenu compares it to the load address for C-64 Basic programs. If AutoMenu finds that the program loads at 2049 decimal, which is where the C-64 Basic programs start, it tells you that this may be a C-64 program and asks if you still wish to load and run it. If you reply yes, it will load it to the proper start of Basic for the C-128 and run it. (Note: Not all C-64 Basic programs will run unmodified in C-128 mode.)

If the load address is incorrect for C-128 or C-64 Basic, AutoMenu assumes the program is in machine language and asks if you wish to boot the program. Booting allows machine language programs to be loaded and activated all at once. It will only work on those machine language files whose start address is the same as the load address. In other words, if the load address is 32768, booting will load it to 32768 and attempt to start it with a SYS 32768. Machine language files that require starting from an address different from the load address cannot be executed this way.

If the program you want is not on the disk you're viewing, simply remove the current disk from the drive (do not remove a disk while the drive is spinning) and insert another disk. Then press the stop key. AutoMenu will load the directory of the new disk and display it for you.

Well, that's all there is to it. You'll find this program will add new dimensions to the use of your disk directory. 🅁

Auto Menu

By Joe W. Rocke

RUN It Right

C-64; C-128 (in C-64 mode)

Auto Menu is a disk-based utility that takes the hassle out of loading a program. The Load and Run commands become a menu-driven operation. With a single keystroke, you select the program to be loaded. Your C-64 then takes over, automatically loading and running the selection. Even the most inexperienced newcomer can load a program.

The C-64 wedge and similar utilities provide shortcuts in typing the Load command. However, it's still up to you to remember and correctly type the program name. Everyone who uses a disk system has been confronted with a File Not Found message because of a typing error. Most of us have to load and read the disk directory unless a crib sheet of program names is handy.

SIMPLIFIED OPERATION

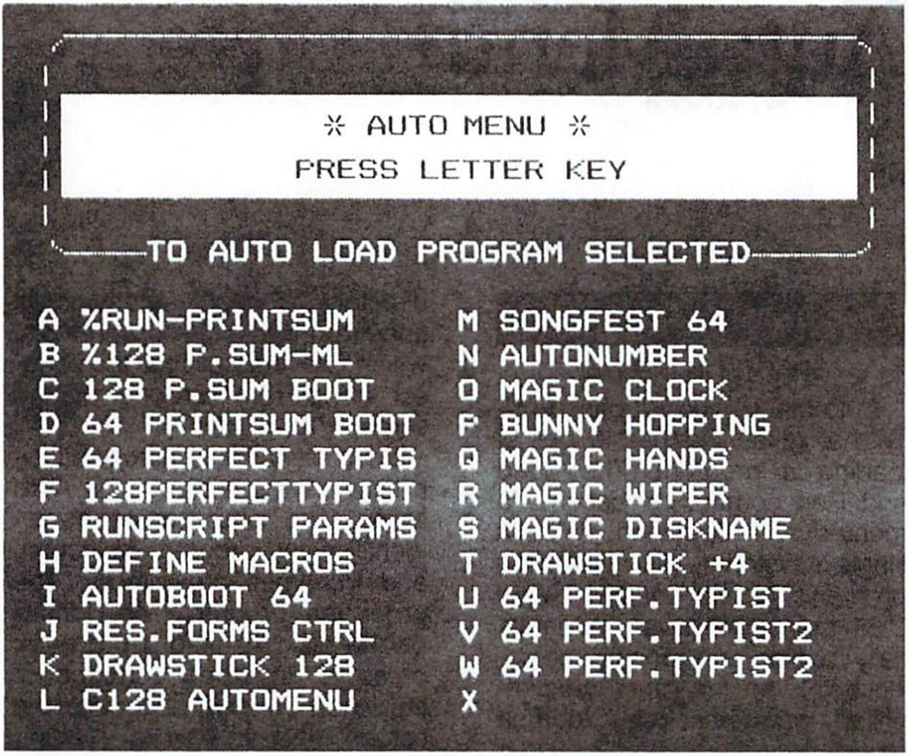
Auto Menu lists the disk directory in menu format. The disk directory is read automatically and listed in double-column form, which prevents all but the longest listing from scrolling off the screen.

Each filename is preceded by a letter that is assigned by Auto Menu. The letter serves as an identifier for program selection, and filenames are listed in the order the programs are stored on the disk. To load a program, press its filename's corresponding letter. This will also clear the screen and display a loading message. A typical message display is as follows:

```
LOAD "PROGRAM NAM*",8,1
SEARCHING FOR PROGRAM NAM*
LOADING
```

No further keyboard input is necessary after the initial menu selection. The loading operation takes place automatically, and, upon its completion, the computer automatically initiates a Run command. Finally, the menu program is removed from memory with a New command.

To facilitate the auto-run operation, the asterisk (*) pattern-matching format is used in Auto



```
              * AUTO MENU *
            PRESS LETTER KEY

   TO AUTO LOAD PROGRAM SELECTED

A %RUN-PRINTSUM       M SONGFEST 64
B %128 P.SUM-ML       N AUTONUMBER
C 128 P.SUM BOOT      O MAGIC CLOCK
D 64 PRINTSUM BOOT    P BUNNY HOPPING
E 64 PERFECT TYPIS    Q MAGIC HANDS
F 128PERFECTTYPIST    R MAGIC WIPER
G RUNSCRIPT PARAMS    S MAGIC DISKNAME
H DEFINE MACROS       T DRAWSTICK +4
I AUTOBOOT 64         U 64 PERF.TYPIST
J RES.FORMS CTRL      V 64 PERF.TYPIST2
K DRAWSTICK 128       W 64 PERF.TYPIST2
L C128 AUTOMENU       X
```

Menu's loading instruction, as described in the 1541 user's guide. Pattern matching simply means that the drive will load the first program that has a name matching the letters in the Load instruction. In this program, the first 12 letters of program names are used. As it's unlikely that a disk will have two programs with the same name, this pattern-matching format should not pose a problem.

Save Auto Menu on each of your disks. When you want to use it, load and run it. The automated operation sure beats typing in loading commands! 🅁

RUN Script 64

By Robert Rockefeller

RUN It Right

C-64; C-128 (in C-64 mode)
Disk drive

RUN Script 64 is a word processor designed specifically for the readers of *RUN* magazine. It's easy to use and, since it's written entirely in machine language, it is fast. Also, you can use this program with Runterm Plus (see the December 1985 and January 1986 issues) and other terminal programs.

RUN Script 64 offers about 70 functions, many of which you'll already know, and the rest of which I tried to keep as intuitive as possible. Learning to use the program is easy because most of the editing keys function almost exactly the way they do in Basic.

Reading your text is also easy, since RUN Script 64 contains a word-wrap feature, which works as follows. If a word is too long to fit into the remaining space on a screen line, the word will not be broken; instead, it will wrap around and be displayed in its entirety at the beginning of the next line.

RUN Script uses one line at the top of the screen to display prompts and operation and error messages. (For example, when RUN Script is in Insert mode, the *ins* message is displayed in this status line.)

This leaves 24 lines free for displaying text. About 40,000 bytes are available for storing text.

ABORTING OPERATIONS

You may abort almost any operation, including saving and loading text, by pressing the CTRL key along with the Commodore key. About the only operation you can't stop is the replace-string function, once replacing has begun; but pressing the CTRL/Commodore combination during the input stage will abort this function.

In RUN Script, the restore key acts as a panic button, taking you out of whatever mode you're in and returning you to normal Text mode. You needn't press the run/stop key with the restore key.

MOVING THE CURSOR

The cursor is restricted to the text area, which means it can only be moved around within text that has already been typed. Seven keys function as cursor controls. These include the four cursor keys, the home key, the CTRL/back-arrow key and the shift/return keys. To use the CTRL/back-arrow, press the CTRL and back-arrow keys together; likewise, press the shift and return keys simultaneously for shift/return.

Cursor keys—By pressing the cursor-down key and shifted cursor-up key, you can scroll up and down through the text area. These two keys function the same as in Basic, except that when you scroll up or down with RUN Script, you can only go up to the beginning or down to the bottom of the text; you can never scroll the text entirely off the screen.

By pressing the cursor-right key or (shift) cursor-left key, you can move the cursor horizontally through the text area. Again with a slight difference, the two keys function as in Basic. The difference is due to RUN Script's word-wrap feature, where a given screen line may have anywhere from one to 39 spaces at the end of it. The extra spaces don't exist in the text area, but are printed to the screen to pad out the line. When moving the cursor horizontally, you'll find it will skip over these invisible spaces. It's impossible to overwrite them.

Shift-Return—This key combination moves the cursor to the start of the next line, just as in Basic.

Home key—Pressing the home key once moves the cursor to the upper-left corner of the screen. Pressing it twice moves the cursor to the start of the manuscript.

CTRL/back-arrow—Pressing the CTRL key with the back-arrow key moves the cursor to the bottom-left corner of the screen, and pressing this combination twice moves the cursor to the text's end.

INSERTING TEXT

There are three methods that you can use to insert text in RUN Script 64.

Shift-INST—Simultaneously pressing the shift key and the INST key inserts one space at the cursor position, exactly as in Basic.

CTRL-I—Pressing CTRL-I toggles you in and out of Insert mode. Whenever you press a key in this mode, a character is inserted at the cursor position. You'll know when you're in Insert mode because an *ins* message is displayed on the status line.

Run/Stop—When you need to insert a lot of text at the start of a long manuscript, CTRL-I and shift-INST are too slow, each requiring the movement of almost the entire text just to insert one character. When a lot of text must be entered, the C-64's 6510 microprocessor simply cannot keep up with a fast typist. Pressing the run/stop key inserts a block of 200 spaces, indicated by solid lines, over which you can enter text. If you wish to insert more than 200 characters, press the run/stop key continuously until the insert block is large enough to suit your needs.

Shift-Run/Stop—This key combination is the functional opposite of the run/stop key, deleting a block of unwanted solid lines. To eliminate these lines, simply position the cursor at the beginning of the block and press the shifted run/stop key.

DELETING TEXT

The three methods of deleting text are detailed below.

DEL—Pressing the key marked DEL deletes the character to the left of the cursor and moves the cursor left one column. This key functions as in Basic.

f8—Pressing f8 also deletes one character, but it's the one *under* the cursor, which does not move.

f7—Sometimes you want to delete several characters or sentences. Doing this one character at a time takes too long. That's when you need a block-delete function. Position the cursor over the first character of the section of text you wish to delete and press f7. A delete-block message is then printed on the status line at the top of the screen. Then move the cursor to the *last* character of the block and press the return key. The block of text will be removed, but not lost.

The deleted block is copied to a buffer at the top of memory. If you decide it shouldn't have been deleted after all, press f6 twice, quickly. (A Select Insert Point message will be printed on the status line.) Move the cursor to the point where the text was deleted and hit the return key. The text will be restored exactly as it was before you deleted it.

If you accidentally hit f7, you can abort without deleting any text by simultaneously pressing the CTRL and Commodore keys.

MOVING AND COPYING TEXT

f5—To move a block of text, position the cursor over the first character of the block and press f5. (A Move Block message will be displayed.) Then move the cursor to the last character of the block and hit the return key. The text is deleted and copied to the same buffer the f7 function uses. (A Select Insert Point message will now appear.) Position the cursor where you want the block inserted, press the return key and the block of text is moved.

f6—There are two ways of copying text blocks by pressing f6. By pressing this key twice in rapid succession, you may insert the contents of the text buffer into your manuscript. After you've pressed f6 twice, a Select Insert Point message will appear. When you move the cursor to the insert point and press the return key, the contents of the text buffer are inserted. This feature allows you to make multiple copies of the same text block or to recover a deleted block of text as explained above.

To copy a portion of text to another position, press f6 once after positioning the cursor over the first character of the block to be copied. (A Copy Block message appears on the status line.) Next, move the cursor to the last character of the block to be copied and hit the return key. (A Select Insert Point message appears.) Move the cursor to the insert position, hit the return key, and the block is copied.

SEARCH AND REPLACE

f2—To search for a string between the cursor position and the end of the text, press f2 once, and a "search string?" prompt appears. Type in the string you're searching for, beginning and ending it with any extra character (for example, :color:); then press the return key. If a match is found, the cursor appears over it. If no match is found, a String Not Found message is printed.

After entering a search string, you may search for multiple occurrences of the same string by pressing f2 twice in rapid succession. This bypasses the subroutine used to select a new search string. (You must move the cursor beyond the first letter of the string found in order that the next occurrence be located.)

f4—Use f4 to replace strings within the text area. There are two modes of operation for replacing text. If you press f4 twice, quickly, you will be asked to input a search string and a replace string. Remember, the first and last characters are ignored. After you've done this, all occurrences of the search string from the cursor position to the end of text will be replaced by the replace string. If you press f4 only once, a search string and replace string will be requested as above; but before each search string is replaced, it will be displayed, and you may elect to replace this occurrence or bypass it.

CHANGING COLORS

To change your display's colors, use the appropriate key combination.

CTRL-1—to change the text color.

CTRL-2—to change the screen color.

CTRL-3—to change the color of the status line.

CTRL-4—to change the border color.

MISCELLANEOUS EDIT FUNCTIONS

Return—Pressing the return key when entering text signals the end of a paragraph. You don't need to press the return key after every line, as on an electric typewriter. When you press the return key with RUN Script, a left-arrow character will appear at the cursor position, marking the end of the paragraph.

CTRL-X—This keystroke interchanges the two characters to the left of the cursor—a fast method of correcting transposed letters.

CTRL-↑—This key toggles RUN Script 64 in and out of Caps Lock mode. When the *cap* message is displayed on the status line, all alphabetic characters typed will be capitalized, regardless of the position of the shift key. This allows entry of capital letters and numbers at the same time, without having to touch the shift key.

CTRL-=—Because RUN Script 64 uses word-wrap, it is often hard to tell exactly how many real spaces follow a word that appears at the end of a screen line. To remedy this, it's possible to select some character other than the space character to fill out screen lines when a word must be moved to the next line to avoid splitting. Press the CTRL and equals-sign key. (A *chr* message will appear on the status line.) Then type the character to be used as the filler character. The screen will then be redisplayed with your filler character in place of spaces. Personally, I favor the graphics character obtained by simultaneously pressing the Commodore and + keys.

THE F1 KEY

The f1 key acts as a command key to access various functions, such as saving, loading text, and so on. Press f1 once, and a "command ?" prompt will appear. Enter a letter corresponding to the function you wish to execute, as listed below.

d—Selects the device with which all saving and loading takes place. This may be device 8 or 9, the disk. The default device is number 8, the disk drive.

@—Saves with replace. (The old file is automatically scratched before the new file is saved. This totally bypasses the DOS bug built into the 1541.)

s—Saves text to device.

l—Loads a text file from the device.

a—Loads the text starting at the cursor position rather than at the start-of-the-text area, as in "l." (This append function is not intended for use with sequential files.)

t—Selects whether characters will be output and input as true ASCII or as Commodore ASCII. Some non-Commodore printers use different ASCII values for some characters than the ones Commodore uses. For example, the true ASCII decimal value of the letter "a" is 97. The Commodore ASCII value for "a" is 41.

Also, when telecomputing with some distant mainframes, you may find that the mainframe uses true ASCII rather than Commodore ASCII. In this case, RUN Script can do conversions between the two types of ASCII using the "w" and "r" commands. (RUN Script uses Commodore ASCII as the default mode.)

w—Writes the text in memory to disk as either a sequential or a user file. If you wish text saved as a sequential file, you must append the string ",s,w" to the filename when entering the filename at the prompt. (The quotation marks are not part of the string.)

If you wish the text saved as a user file, you must append ",u,w" to the filename. In addition, the text can be written to disk using either Commodore ASCII or true ASCII, depending on how RUN Script is configured. See the "t" function, above. (If you change filetypes between a read and a write, you will need to rename the file. Also, you may use Save with Replace by inserting @0: before the filename. This is not a good idea, though, because of the DOS bug.)

r—Loads a sequential or user file from disk. Simply enter the filename at the prompt. It is not necessary to append anything to the filename. If RUN Script 64 is in true ASCII mode, the file loaded must be a true ASCII file; otherwise, upper- and lowercase letters will be reversed. If RUN Script is in Commodore ASCII (normal) mode, the file must be a Commodore ASCII file.

The "r" and "w" functions are included mainly to allow RUN Script to write and read files from other word processors.

$—Lists the disk directory. You can pause the directory listing by pressing the Commodore key and stop the listing by pressing the run/stop key. In order to return to Edit mode, press the CTRL and Commodore keys simultaneously.

>—Issues a disk command.

<—Reads the disk error channel.

f—Reports the number of free bytes remaining.

x—Exits to Basic.

p—Prints text. After selecting the Print function, you'll be asked how many copies to make. (Up to 99 copies are allowed.) Next, you'll be asked to select the output device. Output may be directed to device 3, the screen; device 4 or 5, the printer; or device 8 or 9, the disk drive.

If you choose the disk, output begins immediately and continues until all text has been output. If you don't have a printer, you might want to output your text to disk and then borrow a friend's printer to print out your text.

If you select the printer or screen as the output device, a special output mode is possible. A "continuous (y/n) ?" prompt will appear. If you answer "n" for no, a "next output (c/p/s) ?" prompt will appear before each page is output. This gives you the option of feeding single sheets to your printer.

If you enter "c" (continuous) in response to the prompt, output from that point on will be continuous, and the "next output (c/p/s) ?" prompt will no longer be displayed before each page is output. If you enter "s" for screen, the next page will be printed to the screen. If "p" for printer is entered, the next page will be sent to your printer. This allows you greater control and flexibility in printing.

When text is being printed to the screen, you may pause the output by pressing the Commodore key. You may terminate the output at any time, regardless of the output device, by simultaneously pressing the CTRL and Commodore keys. Output to the screen allows you to see where the page breaks occur. It is recommended that you perform this operation before printing your text.

DOT COMMANDS

RUN Script is a post-formatted word processor. This means that your screen display will vary somewhat from your printout. The text is not formatted until it is printed. To specify output format, RUN Script has 21 dot commands, so called because each command must be preceded by a dot (i.e., a period). These commands are embedded in the text to specify margin widths, define headers and footers, and so on.

A dot command is executed only when text is being printed, and only text following the dot command is affected. For example, if margins aren't set until half a page of text has been entered, the margins in the first half will probably be incorrect in width.

For dot commands to be interpreted correctly, certain procedures must be followed. First, the line immediately preceding a line of one or more dot commands must end with a carriage return. Second, the line of dot commands must start at the extreme left of the screen in the first screen column. Dot commands cannot be preceded by spaces or any other character.

Third, there may be more than one dot command per line, but commands must not be separated by spaces. Fourth, each string of dot commands must be terminated by a carriage return. Here is an example.

```
.pw80.pl66.lm6.rm6.tm4.bm4.lj
```

Each dot command is discussed below.

.lj—Specifies that text will be left-justified when printed.

.rj—Indicates that printed text is to be right-justified.

.cn—Specifies that text is to be centered between the left and right margins. Your text must be entered on the next line below the .cn command.

.pl—Sets the page length. It must be followed by a number from 1–240, indicating how many lines must be output to print one full page. For standard-size paper (8½ × 11) with a printer that prints six lines per inch, the setting would be 66. Therefore, you would enter .pl66.

.pw—Sets the page width, defined as the maximum number of characters that may be printed on one line. Most printers print ten characters per inch, which, using standard-size paper, gives 80 characters per line. If your printer has multiple character sets, with different character densities, you'll have to adjust page width accordingly in order to use one of the alternate character sets. Example—.pw80.

.lm—Sets the left-margin width. The .lm directive must be followed by a number from 1–240 that indicates the width. Example—.lm8.

.rm--Sets the right-margin width. The .rm directive must be followed by a number from 1–240. Example—.rm8.

If the sum of the left and right margins is greater than the page width, a margin error will occur.

.tm—Sets the top-margin depth. The .tm directive must be followed by a number from 1–240. Example—.tm6.

.bm—Sets the bottom-margin depth. The .bm directive must be followed by a number from 1–240. Example—.bm6.

The top-margin depth plus the bottom-margin depth must be less than the page length.

.hd—Defines a header to be printed at the top of every page. It may be followed by up to 255 characters and must be terminated with a carriage return. This means that the carriage-return character cannot be part of the header string and that no dot commands can follow the .hd directive, since they would be interpreted as being part of the header string.

For this reason, and to enhance readability, I recommend that the .hd and .ft (see below) dot commands appear on lines by themselves. One character is given special significance when the header or footer is being printed. When the # character is encountered, the current page number will be printed.

The .lj, .rj, .cn, .pw, .lm and .rm dot commands have no effect on the header when it is printed. The header string is printed exactly as defined, starting at the first column on the page and continuing until it has been printed in its entirety. If, for example, you wish to center a title, the title must be preceded by the correct number of spaces to cause it to be printed at the center of the page.

Macro characters (see below) may be embedded in the header string if desired. The header will be printed at the line position equal to the .tm setting plus 1. If the top margin is set to six, the header, if defined, will be printed on the seventh line. Example—.hd RUN Script 64.

.ft—Defines a footer to be printed at the bottom of every page. The .ft directive works exactly like the .hd directive. The footer, if defined, will be printed at the line position equal to the .pl setting minus the .bm setting minus 1. Example—.ft page number.

.hs—Defines the number of lines to be left between the header and the main body of text. It must be followed by a number. Example—.hs2.

.fs—Defines the number of lines to be left between the footer and the main body of text. Example—.fs2.

.ls—Sets the line spacing. One or more blank lines may be printed between each line of text. If .ls is set to 1, one blank line will be printed after each line of text. Example—.ls1.

.l+—This command causes text to be indented from the current left-margin setting. For example, if the left margin is set to .lm8, and you then execute .l+3, text will henceforth be indented as though .lm11 had been executed. To cancel an indent, simply use .l+0 or .l–0.

.l–—Allows you to have certain text, such as a subheading, print to the left of the left margin. This is accomplished with .l– and is called an outdent. For example, if the left margin setting is .lm8 and .l–3 is executed, text will begin printing at the sixth column, just as though .lm5 had been executed. Outdents are cancelled by executing .l–0 or .l+0.

.fp—Forces a new page. When this command is executed, no more text will be output to the current page. If a footer was defined, the footer will be printed, then the bottom margin and a new page will be started. Example—.fp.

.fl—This command lets you

link files to be printed. The command must be followed by a device number and a filename, separated by a comma. When the .fl directive is executed, the file will be loaded from the specified device and begin printing. The permissible device numbers are 1 for cassette, and 8 or 9 for disk. If two disk drives are used, one manuscript can be over 340,000 characters long. Example—.fl8,next file.

.p#—Sets the page number of the next page to be output. Example—.p#45.

.lf—Indicates that a line-feed character is to be printed after every carriage return. Some non-Commodore printers require this. Example—.lf.

.cm—This is a do-nothing dot command. Its only purpose is to let you leave a comment for yourself that will not be printed when text is output. If you are in the habit of keeping all your old letters on disk or cassette, you can leave a comment to yourself with the date and other information. Example—.cm July 19, 1985 RUN Script 64 article.

ERROR MESSAGES

Macro Not Defined—This error message will be displayed during output if an undefined macro character was encountered. This usually means that you made a typing error when entering the macro character. The cursor will be displayed on top of the incorrect character.

Macro characters are a means of customizing RUN Script 64 to take advantage of special features of non-Commodore printers. It is not necessary to understand macro characters to use RUN Script 64. They are explained in Part 2, when I describe how to customize this word processor to your own liking.

Illegal Quantity—This message will be displayed if a dot command's numeric argument is too large or too small. The cursor will appear near the illegal number.

Output error—This error is displayed when a hardware error occurs during printing. If output is to the printer, the error could result from the printer being out of paper or not turned on.

Syntax error—This error occurs if RUN Script 64 isn't able to recognize a dot command. The cursor will be near the offending command.

Margin error—This error results when the margins have been set to illegal values. Check the dot commands preceding the cursor position.

Text Area Full—This message lets you know that the text area is full. At this point, you must save your text and erase what is on the screen to continue.

Device Not Present—This is caused when the device referenced in input/output operation did not respond. Check to see if the correct device number was used and make sure the device is turned on.

File Not Found—This signifies that an attempt was made to load a file that was not found on the load device.

String Not Found—A match was not found when searching for the search string.

Save error—This means that a status error occurred while the program was attempting to save the manuscript.

Load error—This means that a status error occurred while the program was attempting to load a text file.

THE BOOT PROGRAM

A boot program is one that loads other programs. In RUN Script 64's case, a boot program is required because the machine language program must be loaded, as well as a character-set program; thus, two programs must be loaded by the boot program. You have the option of using alternate character sets.

If the boot program is loaded from disk, all the necessary files must be on the same disk with the boot program; the programs can be saved on disk in any order.

In order to use RUN Script 64, just load and run the boot program, BOOT RUNSCRIPT. This will automatically load the C-64 character set and the machine code. If you are making backup disks of this program, be sure to copy the program COPY CHAR SET. When run, this will automatically create a copy of the C-64 character set on your backup disk.

RUN Script is set up to expect a Commodore printer connected to the serial bus. A non-Commodore printer with a printer interface that emulates a Commodore printer will also work. However, you cannot use an RS-232 printer with it. If you have an RS-232 printer, you can output text to disk and later use a Basic program to print it.

If you don't have a Commodore-compatible printer or interface, you can still use RUN Script by putting your printer interface into Lock mode and using the RUN Script f1-t function to output true ASCII.

In Part 2, I will explain how to customize RUN Script so you can take full advantage of your printer's features. R

RUN Script 64 Part 2

By Robert Rockefeller

RUN It Right

C-64; C-128 (in C-64 mode)
Disk drive

As I said in Part 1, if you have a Commodore-compatible printer, it's not necessary for you to customize RUN Script 64. For those with other types of printers, I will explain about RUN Script 64's printer-macro facility, and then the customizing procedure.

DEFINING PRINTER MACROS

Many C-64 owners use non-Commodore printers such as the Okimate 10 or Epson MX-80. Such printers often have many desirable features, like italic character sets, the ability to do underlining, and so on. The printer-macro feature of RUN Script 64 was designed to allow you to customize your copy of this word processor so you can take full advantage of whatever capabilities your printer may possess.

You may select any upper- or lowercase alphabetic character to be a macro character. You then create a table of printer macros with an easy-to-use Basic program called Define Macros. Each macro character represents a string of user-defined characters. When a macro character is encountered during printing, this string, rather than the macro character itself, will be sent to the output device.

For example, let's say you own a printer that requires the sequence ESC X (decimal values 27,88) to start printing double-width characters. With Define Macros, you could select a character—D, for instance—to represent this two-character string. Then, when D is encountered during printing, the decimal sequence 27,88 would be sent to the printer, resulting in double-width characters. You could then define another character, perhaps d, to represent the correct sequence to stop printing double-width characters.

This macro feature is most useful when you are printing titles and subheadings. To create a double-width heading, first place the cursor in front of the heading. Next, press the f3 key. A *mac* message will appear on the status line. Press the upper- or lowercase alphabetic character you've chosen to activate the double-width capability (in my example, D). Finally, move the cursor to the end of the heading, press f3 again, and press the key you've chosen to deactivate the double-width feature (d, in my example). That's all there is to it! If you press any non-alphabetic character, the operation will be aborted.

Your table can be composed of 52 different macro definitions, each of which can be from one to 20 characters long. Exactly 500 bytes have been allocated for the complete macro table.

CUSTOMIZING PROCEDURE

You can create your own personalized copy of RUN Script, with a custom-character set, default screen colors of your choice, printer parameters you design, and the macro table described above.

Step 1. You should prepare a new disk to hold this customized version. Load the BOOT RUNSCRIPT program. If you've decided to make use of the printer macro feature, change line 60 to:

```
60 MAC$="RUNSCRIPT MACROS"
```

If you're going to use a custom character set (see Step 4), make the variable CHAR$ in line 100 equal the name of your character set. For example, if you've saved a character set under the filename "Big Letters," then change line 100 to:

```
100 CHAR$="BIG LETTERS"
```

Now, save the modified Boot program to your new disk.

Step 2. Step two sets the default parameters of the RUN Script 64 machine language program. First, execute this line of Basic in Immediate mode:

```
POKE 43,1: POKE 44,3*16:
POKE 3*4096,0: NEW
```

This raises the start-of-Basic memory to allow room for the RUN Script 64 machine language program. Now, load RUN Script 64 with:

```
LOAD"0:RUNSCRIPT",8,1
```

After loading, type:

```
NEW <return>
```

Now, load the RUN Script Params program with a normal Basic load, then run it.

Now, you must select the screen colors that RUN Script 64 will use when first starting up.

You will also be allowed to change the variable that sets the highest address used by RUN Script 64. You may set this to a lower value if you wish to have a machine language program (such as a fast-load program) stored in high memory.

Next, you will be prompted to select the default save-device number, which is used for all saving and loading. Then you will be allowed to specify the type of ASCII that RUN Script will use when powering up. If you intend to use true ASCII, answer "true" at the prompt. You can then specify whether the linefeed character (decimal value 10) will be sent after each carriage return character (decimal value 13). Some non-Commodore printers require linefeeds to move to the next line.

The final parameter, secondary address, is very important to users of non-Commodore printers. Those who own Commodore printers and Commodore-compatible interfaces can leave the secondary address at 7 to specify upper- and lowercase printing.

If your printer requires a parallel interface, check your user's manual for the secondary address that passes characters unchanged to the printer. This is the secondary address that should be used to send macro strings to the printer.

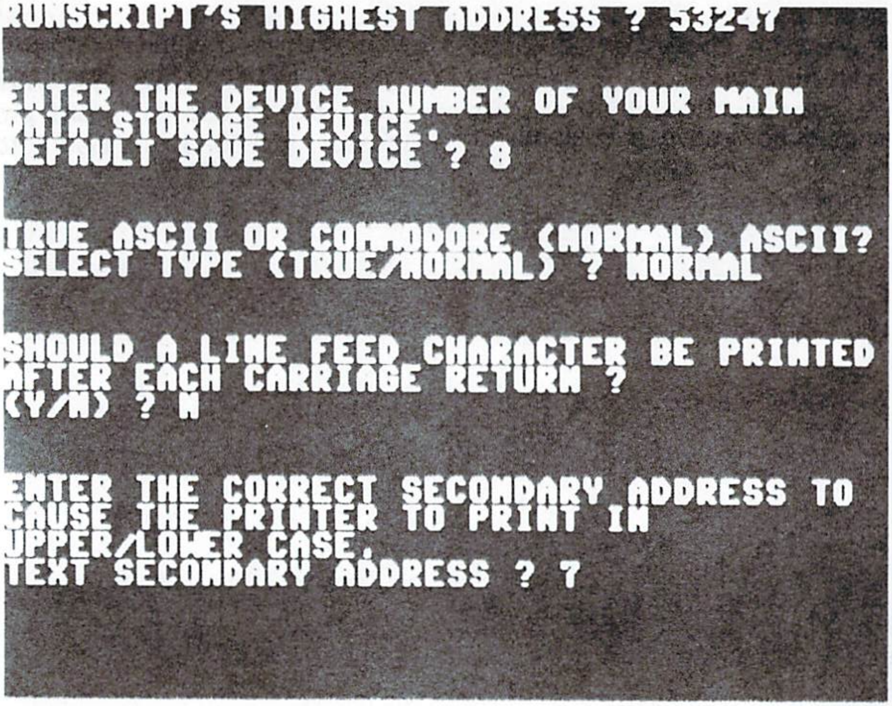
You can also lock your interface into a mode in which characters received by the interface are passed to the printer unchanged. You must do this before starting RUN Script 64, and RUN Script 64 must be set to use true ASCII if this method is used. Also, once your interface is locked, you needn't specify a secondary address. Check your printer interface manual to see if it has this capability.

An advantage to using RUN Script 64 over other word processors is that you can print Commodore graphics characters if you have a graphics interface along with a non-Commodore printer.

After you've set the parameters, the RUN Script machine language program will automatically be saved. Be sure to use your new disk for this.

Step 3. If, for some reason, you have decided not to make use of the printer-macro feature, skip this step and go to the next. Otherwise, load the Define Macros program.

Before running Define Macros, make a list of the alphabetic characters you want to use to represent the various functions your printer can handle. These will be your macro characters. Beside each macro character, write the decimal values of the character sequence that must be sent to your printer to implement the function each individual macro character represents. Then run the Define Macros program.



```
RUNSCRIPT'S HIGHEST ADDRESS ? 53247

ENTER THE DEVICE NUMBER OF YOUR MAIN
DATA STORAGE DEVICE.
DEFAULT SAVE DEVICE ? 8

TRUE ASCII OR COMMODORE (NORMAL) ASCII?
SELECT TYPE (TRUE/NORMAL) ? NORMAL

SHOULD A LINE FEED CHARACTER BE PRINTED
AFTER EACH CARRIAGE RETURN ?
(Y/N) ? N

ENTER THE CORRECT SECONDARY ADDRESS TO
CAUSE THE PRINTER TO PRINT IN
UPPER/LOWER CASE.
TEXT SECONDARY ADDRESS ? 7
```

You will first be prompted to select a macro character. Enter any upper- or lowercase character from A to Z. (If you make a mistake and wish to cancel a macro definition, use the asterisk.) You will then be asked how many characters will be represented by the macro character you've entered. Count them from your list and enter the total.

Next, enter the decimal value of each character in the string, starting with the first and continuing until all have been entered. Once you've done this, you will have defined one macro. The prompt, "Finished all definitions (Y/N)?" will then be displayed. If you have more macro definitions to enter, type N and press return.

After you've entered your list of macro definitions, press Y at the prompt. Within seconds, the program will create the table of macro definitions. The program will then prompt you to save the macro table to disk and provide the proper device number.

Step 4. Finally, you must save a character set on your new disk for RUN Script 64. Load the COPY CHAR SET program and run it. It will automatically create a copy of the C-64 upper- and lowercase character set onto your new disk.

That's all there is to it. You should now have a customized version of RUN Script 64. When you wish to use this word processor, just load the Boot program and run it. 🅁

Create Your Own Keypad

By James E. Borden

RUN It Right

C-64; C-128 (in C-64 mode)

RUN's checksum program for the Commodore 64, 64 Perfect Typist, was first published in *RUN*'s September 1985 issue. The checksum ensures accurate entry of *RUN*'s listings. To complement 64 Perfect Typist, you have the Keypad 64 program, which allows fast entry of numeric data. You may use this keypad alone or with the Perfect Typist.

USING THE NUMERIC KEYPAD

When you run the Keypad 64 program, the start-up screen will be displayed with a SYS address, which is used to wedge the keypad in place. Write down the keypad address in case you need it.

To toggle the keypad on, hold down the CTRL key and press the Commodore key (repeat this procedure when you wish to return to the normal keyboard). You should hear a click, and the cursor will blink about twice as fast as normally, signifying that you are in the Keypad mode. Press the M key, which is now used for 0—you'll hear a click, and a 0 will appear on the screen. (When using the numeric keypad, the C-64's sound capabilities will be restricted to this clicking.) Try pressing various keys—you'll find that only the keypad keys click. To return to the normal keyboard, press the CTRL and Commodore keys again (the cursor will resume its normal speed).

The keypad keys will be easier to find and remember if they're marked with small stick-on numbers (available at art supply stores), as shown in Photo 1. You could also print out small labels and cover them with clear glue to prevent the ink from being rubbed off.

Pressing the run/stop with the restore key will kill the Keypad wedge, and you'll have to use the SYS displayed on the start-up screen to wedge the keypad back in. (Run/stop and restore will *not* affect 64 Perfect Typist.)

BONUS FOR 64 PERFECT TYPIST USERS

As previously mentioned, the numeric keypad is intended mainly for entering data. To help you do this, a very simple auto-numbering routine is provided for Data statements.

First, you must Poke the line number increment into memory location 2. For example, if you want the line numbers of your Data statements to increase by 5, use

```
POKE 2,5 {RETURN}
```

The default value is 10. The last line number entered into the Basic interpreter is considered to be the old line number (this location is also used for Poke and other commands).

The increment is added to that value and the new line number, followed by the Data statement, is printed on the screen. For this reason, *always* check to see if the correct line number is printed on the screen before you enter the line. (You could accidentally press the return key twice, thus skipping a line number.) As long as you enter the lines one after the other, you shouldn't have any problems.

To use auto-data numbering, type in all lines up to the first group of Data statements. Next, enter

```
POKE 2,#
```

(where # is the increment you want) and press the return key.

The routine will only print a line number if the cursor is in the first column. Type in the first Data statement as usual. This will set up the proper line number. After pressing the return key and making sure that the checksum is correct, press the CTRL and shift keys. You'll see the next line number, followed by the word "Data" and a space. You can then type in the data for that line and hit the return key. If the checksum is correct, just press CTRL with the shift key again, for the next line number. Continue this process until the group of Data statements has been entered.

The numeric keypad and the auto-data line numbering should make typing *RUN* listings even easier. R

Songfest

By Joe Kaczynski

RUN It Right

C-64; C-128 (in C-64 mode)

Songfest is designed for the broadest of audiences, ranging from the musical novice to the professional musician. This program allows you to utilize completely the little-understood SID (sound interface device) chip.

Now, your Commodore can use up to three voices to play any song. You need only take the sheet music of your favorite song and, following simple menu-driven questions, enter that song. You can then play it back, modify it to your individual taste, if desired, and then save it to the disk for playback at any time.

Since songs require an unusually large amount of space and are sequential in nature, you must use disks to hold them. I suggest you start by formatting a new disk before you proceed with this program.

As you may already know, the Commodore 64 SID chip is the heart of the 64's music capabilities. This highly sophisticated integrated circuit makes the Commodore a unique instrument indeed. It provides the 64 with a three-voice (three-instrument) and full eight-octave electronic music synthesizer.

More information on the capabilities of the 64 and the SID chip may be found in various chapters of the *Commodore 64 Programmer's Reference Guide*. You may want to refer to this guide if you wish to do some additional experimentation with various parameters involved with the attack/decay and sustain/release (A/D and S/R) settings. Fortunately, you won't need to read any more than this article to utilize to the full your 64's music capabilities.

Songfest has many features that make it very easy to use. To get an accurate reproduction of a particular song, you must enter each note. However, you must also realize that music is made up of a complex combination of notes, rests, tempos, rhythms and octaves. All of these combine to make a song. The correct combination of all these variables is available to everyone, directly from sheet music. Songfest makes it easy to set this data into your 64 in the appropriate form.

When you have entered Songfest into your 64 and saved it, run the program. You will see the following:

```
Welcome to Songfest
Menu
R  = read preprogrammed song from disk
P  = play song once
C  = continuous play; space bar to end
I  = input notes of your own song
A  = add notes to song
M  = modify waveform, A/D, S/R
S  = sound change during playback
E  = error correction after input
K  = keyboard function
Q  = quit and exit program
```

Any time you wish to leave Songfest and return to Basic, simply press Q (quit and exit program) while viewing the main menu.

MENU EXPLANATION

R—read preprogrammed song from disk. If you wish to load a preprogrammed song that you have previously saved, you select this option on the main menu. (We have included Mr. Sandman, written in three voices, for you to play.) When you select this option, be sure to select the number of voices in which you saved the song.

There is, of course, no way for Songfest to know how many voices your own creation used. When the screen displays the question, "How Many Voices Would You Like?", you are simply being asked how many voices were actually stored to the disk when you last saved the song. If you try to read more voices than you actually stored, you'll get an error. Similarly, if you try to read fewer voices than you wrote, you will also get an error.

You will then be asked for the name of the song. Type in its name exactly as it appears in the disk directory, but without the quotes. Once your song is loaded, it will begin playing automatically. When it is finished playing, you may. . .

P—play song once or C—continuous play. These two menu options seem clear enough, but there are a few things you might like to know about them. First, you can play back any song in the number of voices in which it was recorded (saved). You can also play back any song in fewer voices, but you can't play back a song in more voices than it was saved in (those voices are just not there). If you choose C, pressing the space bar at any time will end the song and return you to the main menu.

I—input notes of your own song. When this option is selected, the screen clears and displays the following messages, which will enable you to set up all of the necessary parameters

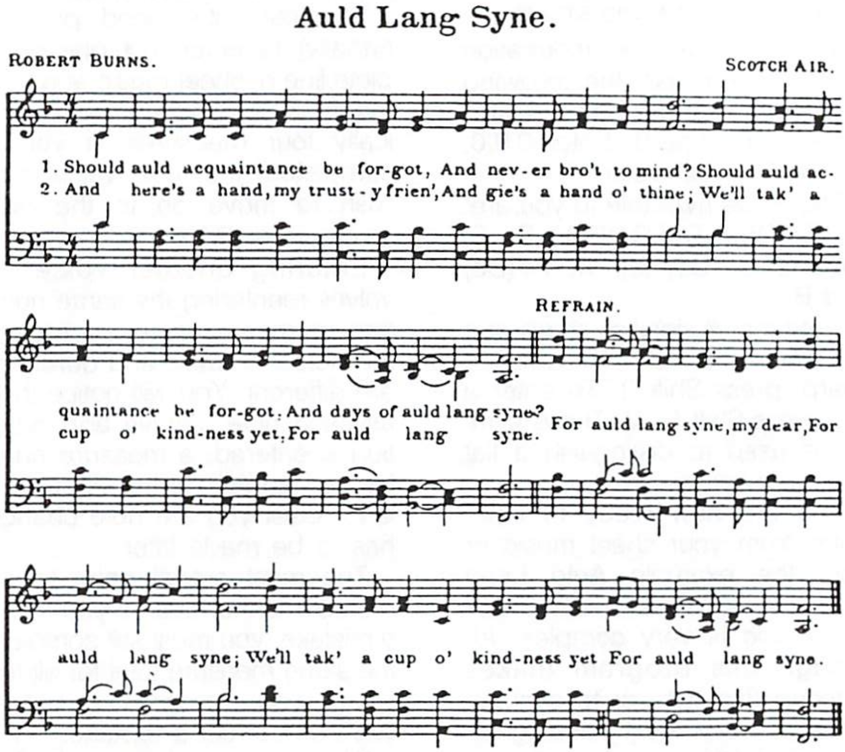

Auld Lang Syne.
ROBERT BURNS.
SCOTCH AIR.
1. Should auld acquaintance be for-got, And nev-er bro't to mind? Should auld ac-
2. And here's a hand, my trust-y frien', And gie's a hand o' thine; We'll tak' a
REFRAIN.
quaintance be for-got, And days of auld lang syne?
cup o' kind-ness yet, For auld lang syne.
For auld lang syne, my dear, For
auld lang syne; We'll tak' a cup o' kind-ness yet For auld lang syne.

Figure 1. Auld Lang Syne sheet music.

critical to the types of voicings your songs will play in. Parameters for voice 0 are now:

W/F—0, A/D—0, S/R—0
Select waveform (17,33,65)
Select A/D (0-255)
Select S/R (0-255)

Also, if the waveform 65 is selected, the following additional message appears:

High Pulse (0-15); a suggestion is 8
Low Pulse (0-255); a suggestion is 128

When you initially enter your songs, I suggest you use the following values. Later, you may wish to experiment with other values. Incidentally, if you only wish to change one of these parameters in a voice, you may press the return key until you get to the specific parameter you wish to change.

```
Voice 0 W/F = 17 A/D = 9 S/R = 240
Voice 1 W/F = 33 A/D = 10 S/R = 120
Voice 2 W/F = 33 A/D = 10 S/R = 85
```

When the above information has been entered, the following message will appear: "Enter Notes For Voice 0, Enter 0,0,0, To Go To Next Voice."

The notes available to you are: C, C#(Db), D, D#(Eb), E, F, F#(Gb), G, G#(Ab), A, A#(Bb) and B.

Note that # signifies sharp and b denotes a flat. To enter a sharp, press Shift 3. To enter a flat, press Shift 6 (&). The & symbol is used to distinguish a flat from a lowercase b.

You are now ready to input notes from your sheet music or from the example Auld Lang Syne (Figure 1). As indicated earlier, music is very complex. Although this program makes entering this data quite straightforward, you must take great care in entering each note, octave and duration, and you must enter them in just that particular format—note,octave,duration.

Specifically, the format must be a note (which may be a letter by itself or a letter and a sharp or flat; refer to Figure 3) followed by a comma, a number followed by a comma and another number. For example: C#,5,4.

A comma must be used to separate off each entry, or it will be rejected and you'll be prompted with a Redo From Start error message. You may then reinput the line where the error occurred.

Generally, it's good practice (initially) to enter just one complete line of sheet music at a time (as in my example). This is typically four measures. If you've completed the first voice and wish to move on to the next voice, enter 0,0,0.

Entering another voice involves reentering the same number of measures, even though the notes, octaves and durations are different. You will notice that, as each note, octave and duration is entered, a measure number will appear to the right. This is to assist you if a note change has to be made later.

The most critical point to remember is that even if you make a mistake, you must still complete the same measure total for all the voices you are entering before you can correct a mistake.

When you make a mistake, simply jot down the measure number where the mistake occurred. If you don't complete the proper measure count for all voices, the entire series must be redone from the last "A = Add Notes To Song" selection.

It's a good practice to write down the final measure number before you go on to the next voice. When you complete the next voice, you can then com-

NOTE TYPE	DURATION	EXAMPLE
Whole note	16 beats	𝅝
Dotted half note	12 beats	𝅗𝅥.
Half note	8 beats	𝅗𝅥
Dotted Quarter note	6 beats	♩.
Quarter note	4 beats	♩
Dotted Eighth note	3 beats	♪.
Eighth note	2 beats	♪
Dotted sixteenth note	1.5 beats	𝅘𝅥𝅯.
Sixteenth note	1 beat	𝅘𝅥𝅯
Whole rest	16 beats	𝄻
Half rest	8 beats	𝄼
Quarter rest	4 beats	𝄽
Eighth rest	2 beats	𝄾

Figure 2. Songfest Reference Guide.

pare and be sure that the measure numbers are the same. Therefore, until you become more familiar with what you're doing, only enter a maximum of four measures at a time for all the desired voices.

Now use the A option to continue your input. This way, if you make an error, you will only have to enter, at most, four measures of the song. When the selected number of voices have been properly entered, play the song to make sure you find it pleasing. If you are satisfied, press the A option again and continue inputting your song.

If you feel a note is incorrect, use the E option. You will be asked which voice contains the error (0, 1 or 2) and for the measure number just before the note to be changed.

Up until the time you press the return key, you may edit as always by pressing the delete key and retyping the data. Once you press the return key, your data will be saved within Songfest. Changes at this point require the use of the E option. Duration errors are basically fatal, because they offset the entire song's timing and measure count, therefore requiring reentry of the entire song.

If you discover an error in a note or octave, mark down the measure number that's beside the incorrect entry. Continue entering until you have entered all notes up to the previous voice's measure number. (Remember, all voices must have the same number of measures.) If you've decided to enter all three voices, all three voices must be at the same measure number before you have a correct song.

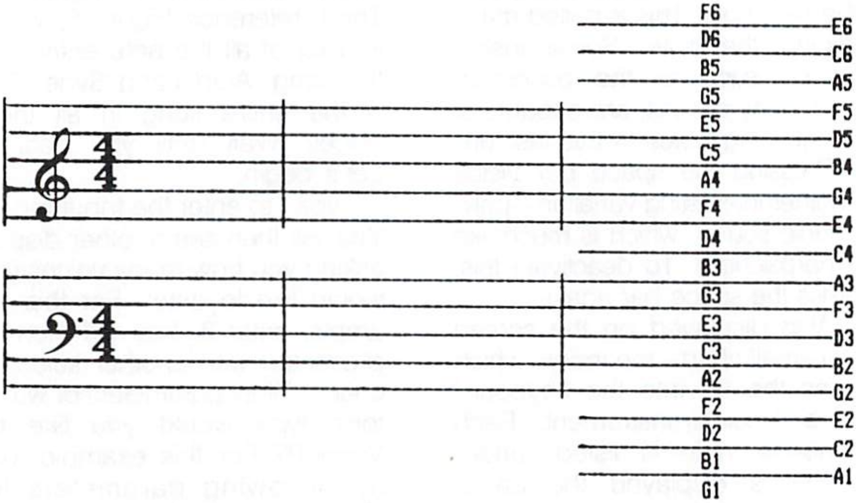
This will return you to the main menu, where you can select the E option. This option will ask you which voice you would like to change. You will then be asked for the measure number of the incorrect note. After Songfest finds the note, you will be asked to reenter the note and the octave.

M—modify waveform. After working with Songfest for a while, you may wish to do some experimentation of your own. This option gives you the capability of playing back your song in just about any combination of waveform and attack/decay and sustain/release parameters possible on your 64. The M option will first tell you what the present settings are and then ask you to input your new selections. The A/D and S/R parameters are explained fully in the sound section of your user's manual.

The suggested waveforms are: a triangle, which is 17; a sawtooth, which is 33; and a pulse, which is 65. You may also select different settings for the A/D and S/R parameters.

These parameters all affect the types of sounds the SID chip produces. By careful experimentation and selection, you'll be able to reproduce various musical instruments. The possibilities are virtually endless for the types of sounds you can create, including those unique to the SID chip itself.

S—sound change during playback. This menu selection lets you experiment with various sounds without really understanding anything about waveforms or A/D and S/R settings, which you may not have had time to learn yet. Once you've loaded a song, either from the disk or the keyboard, press S. Pressing f1, f3,



NOTE AND OCTAVE IDENTIFICATION CHART

This is a sample staff with all the notes labeled. The letter indicates the note name and the number indicates the octave. These are the values you will be using during the Add or Input mode in Songfest. Simply find the placement of the note from your sheet music by using this guide. The duration may be found by referring to the note type chart on the previous page.

Figure 2. Songfest Reference Guide continued.

f5 or f7 will play the song with different preselected voice parameters. If, after experimentation on your own, you like the sound you have created, you may change these parameters yourself. They are located in lines 920-970 of the program.

K—keyboard synthesizer. The keyboard synthesizer is another great feature of Songfest. Basically, what the K selection yields is a screen of help text, which provides you with various alternatives for selecting, tuning and playing your new musical instrument, the Commodore 64.

The f1, f3, f5 and f7 function keys select the octave in which you wish your instrument to play. (F1 is the highest octave; f7, the lowest.)

The f2, f4 and f6 keys select the waveform or instrument type you wish your Commodore to be. You'll find f2 to be a smooth, mellow sound, while f6 tends to be a bit twangy.

Pressing f8 allows any note to continue to play until you press the next note. This is called maintaining the note. Some instruments, such as the accordian and many organs, are capable of maintaining notes in this fashion.

Pressing the space bar yields another interesting variation—polyphonic sound, which is much like a harpsichord. To deactivate this, press the space bar again.

Also displayed on the screen is a small chart—the index, which turns the 64 into the keyboard of a musical instrument. Each available note is listed, under which is displayed the corresponding letter that you must type to produce this sound. To return to Songfest, you may press Z at any time.

Q—quit and exit program. This selection, of course, returns you to Commodore Basic. If you accidentally press Q while inputting or playing your song, you can recover by immediately entering GOTO 40 in the Direct mode.

SONGFEST SAMPLE APPLICATION

The following section will assume that you have an understanding of the basics of music.

Relax, now, and I'll take you through a brief example of entering music into Songfest.

First, be prepared to reference Figure 2, the Songfest Reference Guide. Second, take a look at the sample song on the sheet music provided in Figure 1. Third, reference Figure 3, which is a list of all the note entries for the song, Auld Lang Syne. This is the entire song in all three voices. Wait until you hear it! Let's begin.

Type I to enter the Input mode. You will then see another display asking you how many voices you would like to enter. For this example, enter 3. You will then be presented with another selection choice. What parameters or waveform type would you like for Voice 0? For this example, use the following parameters for Voices 0, 1 and 2 as indicated below:

Voice 0	W/F	=	17
Voice 0	A/D	=	9
Voice 0	S/R	=	240
Voice 1	W/F	=	33
Voice 1	A/D	=	10
Voice 1	S/R	=	120
Voice 2	W/F	=	33
Voice 2	A/D	=	10
Voice 2	S/R	=	85

Of course, you may change these parameters during the playback if you are not happy with them.

After selecting these parameters, which basically select the "instruments" that will play the song you're about to enter, you will be presented with the note input questions. From here, you just need to enter the data listed in Figure 3.

But why don't you look at this whole process a little closer? If you look at the sheet music, you'll notice a time signature of 4/4. This means that you have to treat each measure as if it had 16 beats. For a 4/4 time signature, all (and I do mean *all*) measures must have a total of 16 beats. On the other hand, a 3/4 time signature would have 12 beats per measure.

The funny-looking & sign on the left of the music is called a G clef. To the right of that is a flat designation (b) on the staff line B (remember, Every Good Boy Does Fine). This tells you that every time you encounter the note B, you must enter it flat (Shift 6). Another interesting way to input B flat is to enter A sharp (A#).

You may always use this conversion if it's easier for you to remember. To visualize this a little better, you might picture a flat as being ½ step below a given note and a sharp as being ½ step above a given note. Therefore, an A sharp is equal to a B flat. Similarly, an F# is equal to a G flat. All right, let's get back to the sheet music.

Knowing that each measure has to have 16 beats because of the time signature 4/4, let's take a look at the first measure in Auld Lang Syne. There's only one note there, and it's a quarter note, which, according to your reference chart, is only supposed to get four beats.

What to do now? Well, this happens sometimes, but, remember, all measures with this 4/4 time signature must have 16 beats, or the song is not going to sound right. Here you can utilize a "rest" to fill the unused beats in the measure. Enter S,4,12 as your first entry. Then enter C,4,4 for the quarter note. That gives you a total of 16 beats, and you're in business.

You can enter as many notes as you'd like (up to 90 measures) in each voice, but I strongly suggest that, until you are thoroughly familiar with the program, you enter only four measures at a time. Enter the four measures, then enter 0,0,0 to move on to the next voice, and repeat this procedure until all three voices are complete. Play the song to make sure it's all right. Now use the A option to continue inputting. After you've finished, sit back and listen to what you have accomplished. It sounds great, doesn't it?

HINTS FOR ADVANCED USERS

To suit your own taste, you may change the number of beats that each note normally gets, but only if you change all others accordingly. For example, if you purchase some sheet music, enter it into Songfest and find that the

```
Voice 0: W/F=17  A/D=9   S/R=240
Voice 1: W/F=33  A/D=10  S/R=120
Voice 2: W/F=33  A/D=10  S/R=85

Voice 0
N, O, D
-------
S, 4, 12
C, 4, 4

F, 4, 6
F, 4, 2
F, 4, 4
A, 4, 4

G, 4, 6
F, 4, 2
G, 4, 4
A, 4, 4

F, 4, 6
F, 4, 2
A, 4, 4
C, 5, 4

D, 5, 12
D, 5, 4

C, 5, 6
A, 4, 2
A, 4, 4
F, 4, 4

G, 4, 6
F, 4, 2
G, 4, 4
A, 4, 4

F, 4, 6
D, 4, 2
D, 4, 4
C, 4, 4

F, 4, 12
D, 5, 4

C, 5, 6
A, 4, 2
A, 4, 4
F, 4, 4

G, 4, 6
F, 4, 2
G, 4, 4
D, 5, 4

Voice 0
N, O, D
-------
C, 5, 6
A, 4, 2
A, 4, 4
C, 5, 4

D, 5, 12
D, 5, 4

C, 5, 6
A, 4, 2
A, 4, 4
F, 4, 4

G, 4, 6
F, 4, 2
G, 4, 4
A, 4, 2
G, 4, 2

F, 4, 6
D, 4, 2
D, 4, 4
C, 4, 4

F, 4, 12
S, 4, 4

0, 0, 0

Voice 1
N, O, D
-------
S, 4, 12
C, 4, 4

C, 4, 6
C, 4, 2
C, 4, 4
F, 4, 4

E, 4, 6
D, 4, 2
E, 4, 4
E, 4, 4

F, 4, 6
F, 4, 2
F, 4, 4
F, 4, 4

F, 4, 12
F, 4, 4

F, 4, 6
F, 4, 2
F, 4, 4
F, 4, 4

E, 4, 6
D, 4, 2
E, 4, 4
E, 4, 4

D, 4, 6
A, 3, 2
A#,3, 4
A#,3, 4

A, 3, 12
F, 4, 4

F, 4, 8
F, 4, 8

E, 4, 6
D, 4, 2
E, 4, 4
E, 4, 4

Voice 1
N, O, D
-------
F, 4, 8
F, 4, 8

F, 4, 12
F, 4, 4

F, 4, 6
F, 4, 2
F, 4, 4
C, 4, 4

E, 4, 6
D, 4, 2
E, 4, 4
E, 4, 4

D, 4, 6
A, 3, 2
A#,3, 8

A, 3, 12
S, 4, 4

0, 0, 0

Voice 2
N, O, D
-------
S, 4, 12
C, 4, 4

A, 3, 6
A, 3, 2
A, 3, 4
C, 4, 4

C, 4, 6
A, 3, 2
A#,3, 4
C, 4, 4

A, 3, 6
A, 3, 2
C, 4, 4
A, 3, 4

A#,3, 12
A#,3, 4

C, 4, 6
C, 4, 2
C, 4, 4
A, 3, 4

C, 4, 6
A, 3, 2
A#,3, 4
C, 4, 4

A, 3, 6
F, 3, 2
F, 3, 4
E, 3, 4

F, 2, 12
A#,3, 4

A, 3, 6
C, 4, 2
C, 4, 8

C, 4, 6
A, 3, 2
A#,3, 4
A#,3, 4

Voice 2
N, O, D
-------
A, 3, 6
C, 4, 2
C, 4, 4
A, 3, 4

A#,3, 12
D, 4, 4

C, 4, 6
C, 4, 2
C, 4, 4
C, 4, 4

C, 4, 6
C, 4, 2
C, 4, 4
A, 3, 4

A, 3, 6
F, 3, 2
F, 3, 4
E, 3, 4

F, 3, 12
S, 4, 4

0, 0, 0
```

Figure 3. Auld Lang Syne input data.

tempo is too slow or too fast, you can change the speed by altering the note-duration value.

In a song that is playing too quickly, double the duration value of each note. Give a whole note 32 beats, a half note 16 beats, a quarter note eight beats, and so on.

In a song that's being played too slowly, cut the duration in half. Give a whole note eight beats, a half note four beats, and so on. Remember, make sure all notes are treated proportionately. Don't be afraid to experiment. As long as each voice has the same number of measures, there will be no difficulty.

Here's another hint that may clarify some points on A/D and S/R settings. Being an advanced user, you may understand that the actual settings of A/D and S/R are actually controlled by two bytes in the SID chip. The upper nibble of the A/D byte is A, and the lower nibble is D. The same holds true for the S/R byte—S being the high nibble and R being the low nibble.

What this means is that a little calculation is necessary to obtain meaningful values of A/D and S/R. First, select the desired value of decay (D) and release (R) between 0 and 15. Remember, a lower number means a quicker decay or release rate. Now pick a value for attack (A) and sustain (S) from 0 to 15.

The lower the attack number, the faster the note rises to its initial peak. The sustain value determines the amplitude (volume) during the sustain portion of the note.

Now, to calculate the actual value of A/D from 0 to 255, take the value (from 0 to 15) you have selected for the attack and multiply it by 16. Then, add this value to the value you have selected for the decay (from 0 to 15). This will give you the correct number to enter to select A/D. The same procedure applies for sustain and release.

You are now well on your way to becoming a great computer musician. Good luck. R

AutoBoot

By Doug Smoak

RUN It Right

C-64; C-128 (in C-64 mode); disk drive

If you would like your Basic programs to run automatically without typing in RUN, or have machine language programs start without having to remember what SYS address to use, then you'll appreciate AutoBoot. AutoBoot makes self-starting "boots" that load and run Basic or machine language programs from disk.

HOW TO USE AUTOBOOT

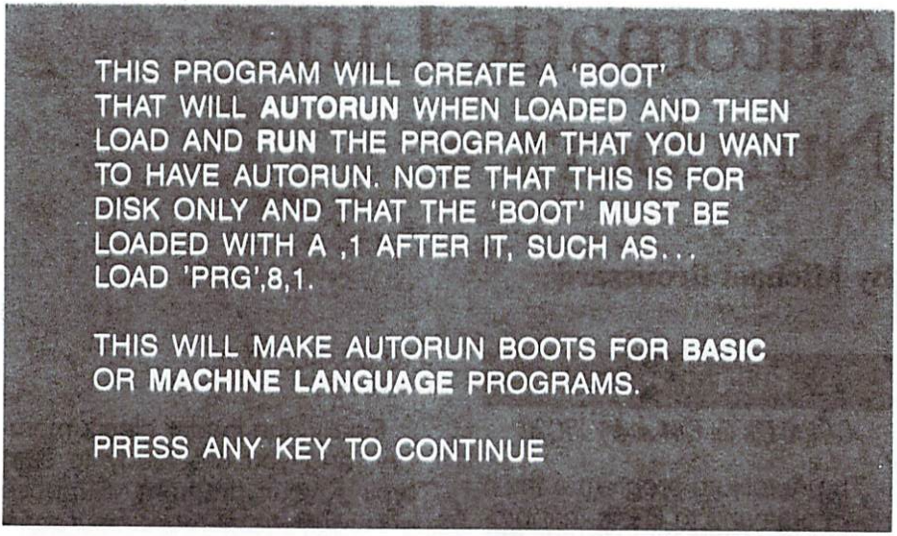
Load and run AutoBoot 64. After running, you may either read instructions or continue with the creation of a boot. After a pause in which the program sets up an array, you must supply the name of the program you want to run automatically. You can use any valid filename and can have a wild card (*) on the end.

You must then supply a name for the boot itself. The best way to do this is first to save the program under a simple name such as A, then call the boot the actual name of the program. Then, when the user loads

```
"program name",8,1
```

the boot will automatically load and run the intended program. Note that the boot *must* be loaded as a non-relocating program file, hence the ,1 after the device number 8. Also note that the disk to which you want the boot saved must be in the drive while AutoBoot is creating the boot. You can transfer the boots with a program such as Copy All or other single-file transfer programs, but this is an unnecessary step.

AutoBoot, as is, will not create boots that display the searching and loading messages for the program to be autorun. I felt this would be confusing to novice users. If you want the messages for your own peace of mind, then substitute the number 128 for the first 0 after the first 169 in line 620. If you want to use a drive with a device number other than 8, then substitute 9s for the two 8s in line 620.



```
THIS PROGRAM WILL CREATE A 'BOOT'
THAT WILL AUTORUN WHEN LOADED AND THEN
LOAD AND RUN THE PROGRAM THAT YOU WANT
TO HAVE AUTORUN. NOTE THAT THIS IS FOR
DISK ONLY AND THAT THE 'BOOT' MUST BE
LOADED WITH A ,1 AFTER IT, SUCH AS...
LOAD 'PRG',8,1.

THIS WILL MAKE AUTORUN BOOTS FOR BASIC
OR MACHINE LANGUAGE PROGRAMS.

PRESS ANY KEY TO CONTINUE
```

Above all, remember that the boots *must* be loaded with a ",1" after the 8, such as

```
LOAD "PROGRAM",8,1
```

I hope you enjoy making your programs run automatically. It adds a professional touch and keeps control in the hands of the programmer; novice users can then be free to interact with the software with a minimum of effort. R

Automatic Line Numbers

By Michael Broussard

RUN It Right

C-64; C-128 (in C-64 mode); VIC-20

The AutoNum program adds a new command to Basic that provides automatic line numbering as you type. Written in machine language, AutoNum is in the form of a Basic loader. The program Pokes the machine language into upper RAM and then deletes itself.

Using AutoNum is easy. Each time you turn on (or reset) your Commodore, you must first load and run the AutoNum loader. The loader clears the screen and prints a message to let you know it is initializing. It then Pokes a Basic wedge into a place in RAM where it won't interfere with the Basic program you are entering. (How this wedge works is explained later.) After a few seconds, the loader prints the following set of abbreviated instructions:

```
TO ENABLE, TYPE:
  AUTO <increment>
TO DISABLE, TYPE:
  AUTO
```

Finally, the loader clears memory, and you are ready to start typing your program. Suppose you wish to number your program by tens. Simply type in:

```
AUTO 10
```

and press the return key. You will get a Ready message back from Basic, and AutoNum will be turned on. Now you type in the first line of your program. Suppose it is:

```
100 PRINT
```

When you type in the line and press the return key, the line becomes part of the Basic program in memory, as usual. In addition, AutoNum prints the next line number:

```
110
```

The cursor is then positioned so that you are ready to type in the next program line.

Continue entering program lines in this fashion until you have typed in the whole program or until you want to enter a direct command, like Save. To perform the latter, simply backspace over the line number that AutoNum printed and then type in your direct command. Or you can simply press the return key after the next AutoNum line number. Either action will disable AutoNum until you type in another line beginning with a line number. Then AutoNum will step in and start printing line numbers again, continuing from the new line number.

To turn off AutoNum completely, simply type:

```
AUTO
```

in Direct mode. (If you type this command next to a line number and press the return key, you will add an unwanted line to your program and AutoNum will still be enabled.)

That's all there is to using the program. The rest of this article is directed toward more advanced programmers and tells how to customize the loader in order to place AutoNum at a particular RAM location. It also explains how AutoNum works.

CUSTOMIZING THE LOADER

As mentioned earlier, AutoNum resides in a part of RAM that won't interfere with Basic programs. On the Commodore 64, it begins at location 49152. On the VIC-20, there is no free RAM area for machine language use, so the loader "steals" a page of Basic program space by moving the top-of-memory pointer down 256 bytes. If you won't be using AutoNum with a Datassette, you can reclaim this program space by deleting line 90 of the loader and changing line 100 to read:

```
100 BASE=828
```

This will cause AutoNum to be Poked into the cassette buffer, which is a great place to put short machine language subroutines as long as you are *not* using a Datassette. (C-64 users can also redirect AutoNum to the cassette buffer by making the same changes.)

For those of you familiar with machine language, you can move AutoNum to any other location in RAM by changing the value assigned to BASE on line 100. (Of course, you must be careful not to place the routine where it interferes with Basic or with the operating system, or you might cause a "crash.")

Have fun with this program. R

ML Perfect Typist For the C-64

By Robert Sims

RUN It Right

C-64; C-128 (in C-64 mode); disk drive

ML Perfect Typist is a checksum for machine language programs that you'll be typing in from *RUN* magazine on the C-64 (or the C-128 in C-64 mode).

When you load and run ML Perfect Typist, the Basic portion of the program Pokes into memory the machine code from the Data statements and executes the code.

TYPING IN MACHINE LANGUAGE LISTINGS

Machine language listings in *RUN* consist of a series of program lines, each of which begins with a four-digit number, the line number, followed by 13 pairs of characters, separated by spaces.

You do not enter the line number; the program enters it for you automatically. The spaces in the lines are for ease of reading only; they also appear automatically.

The pairs of characters are hexadecimal numbers. The first eleven pairs are the program's machine code. The last two pairs in each line are checksum values used by ML Perfect Typist to ensure that you've entered the line correctly.

To type in a machine language listing, first load and run ML Perfect Typist. At the opening menu, select 1 to enter a new program.

You will see a line number on the screen. Find this line number in the listing and type in the characters that follow the line number. ML Perfect Typist automatically takes care of spacing on the screen. If you make a mistake, simply delete your error and retype the character.

When you've typed in the 13 pairs of characters, compare what you see on the screen with the line in the magazine. If they're the same, press the return key. If you find an error, use the delete key to back up to the incorrect character. Type the correct character, then reenter the rest of the line. Check it again, then press the return key.

If the line has been entered correctly, ML Perfect Typist will prompt you with the number of the next line to be entered. If you've made a mistake, Perfect Typist will alert you to this and prompt you to enter the same line again.

The line is not processed until you press the return key and ML Perfect Typist verifies the contents. The line is then converted to binary numbers and stored in its proper place in memory.

SAFETY FEATURES

ML Perfect Typist will accept only the keys 0–9, A–F, delete and return. If, for example, you mean to press the 7 key but hit the T key instead, ML Perfect Typist will ignore the T and wait for you to press a valid key.

In case of power failure or other disaster, it's a very good precaution to save your entries to disk every 50 lines or so.

COMMAND CODES

There are three command codes, generated with the CTRL key: CTRL-A, CTRL-C and CTRL-E.

If you enter CTRL-A (hold down the CTRL key and press A), the program will abort. Only use CTRL-A if you become hopelessly confused while entering a listing. CTRL-A causes ML Perfect Typist to end without saving the entered lines to disk.

Use CTRL-C to cancel entry of a line. When you type CTRL-C, ML Perfect Typist will erase the line and prompt you to enter it again. Only the current line is affected.

HANDLING INCOMPLETE PROGRAMS

CTRL-E has two uses. First, use it if you want to end a session before you've entered the complete listing. Be sure, of course, to have a disk in the drive, or you'll lose the data you've already entered. Instead of entering the next line, press CTRL-E. You'll be asked for a filename. Then a short menu will be displayed. Choose option 2. When you select option 2, ML Perfect Typist will save the incomplete file to disk.

Caution: You *must* use option 2 to save your incomplete file! If you use option 1, you'll never be able to recover your program for further line entry. Also, if you name a file that exists on the disk, ML Perfect Typist will scratch the old file and replace it with the new one.

Later, when you're ready to resume entering the listing, load and run ML Perfect Typist and select 2 from the opening menu. You'll then receive a prompt for the filename. Enter the name of the incomplete file and press the return key. ML Perfect Typist will load that file into memory and then display your next line number. This is a handy feature because it always identifies the next line to be entered. Resume entry by typing in the character pairs in the line whose number is displayed.

STORING A COMPLETED PROGRAM

The second use for CTRL-E comes when you have entered the last line of a listing. When you are prompted with a line number larger than the last line number in the listing, enter CTRL-E and supply the filename. Next, you will see the menu. Select option 1 for storing a completed program, and ML Perfect Typist will store it to disk. Again, you *must* have a disk in the drive before you enter CTRL-E.

CAUTION!

If you follow instructions, ML Perfect Typist will give you an error-free copy of the machine language program you enter. However—keeping in mind the theory that, if anything can go wrong, it will—take precautions. When entering your program lines be sure that you are entering them in the correct sequence. Do this by checking the line number on the screen against the one you are entering from the magazine.

Also, you will notice that all lines, except the last one, are the same length. Therefore, when you have entered all the characters in any line except the last one, your cursor should be in the 40th column on the screen. If it isn't, you must have left out one or more characters. If you press the return key at this point, you should get an error message and a prompt to retype the line.

ML Perfect Typist uses the last two character pairs of each line you enter as a checksum, to see whether you've entered the line correctly. There is a slight chance that, if you leave characters out of a line, ML Perfect Typist will add up the value of the characters, compare them to the last two character pairs, find a match and store an incorrect line. To avoid this, always check that your cursor is in the 40th position before you press the return key.

For the special case of the last line, just check to be sure you've entered the correct number of characters.

If you enter too many characters in a line, ML Perfect Typist will alert you to this and prompt you to enter the same line again. R

A Modern Metronome

By Robert S. Owen

RUN It Right

C-64; C-128 (in C-64 mode)

The metronome is a device that beats time to help a person keep a regular tempo when practicing music. The first metronome of 1816 was a mechanical device invented by a German mechanic named Johann Maelzel. (A similar device is said to have been constructed by Dietrich Winkel in 1815.)

You frequently see that the tempo (speed) marked on musical scores is in the form of MM = 90. The letters MM stand for Maelzel's Metronome; the number 90 stands for 90 beats per minute.

Most mechanical metronomes are powered by a key-wound spring that swings an inverted pendulum. A weight, which may be moved up or down this pendulum, regulates the metronome's speed. As the weight is moved closer to the end (top) of the pendulum, the metronome beats more slowly.

A scale on the pendulum indicates the number of beats per minute. The musician can see this pendulum swing as well as hear the beats while he plays his instrument.

Maelzel's Modern Metronome has all the features of this classic mechanical device: it has a weight to change tempo; the end of the pendulum visibly moves with the beat; and, of course, there is an audible beat.

But Maelzel's Modern Metronome has some extra modern features. The audible beat is slightly accented to match the time signature of the music. Every third beat may be accented for 3/4 time and every fourth beat for 4/4 time. Six-eight time is accented differently than 6/4 time by Maelzel's Modern Metronome. Even the irregular metres of 5/4 and 7/4 time are recognized (one-two-one-two-one-two-three).

This modern metronome also has a clock, which you may set to the real time, so you'll know when to quit practicing and leave for your music lesson.

If you don't set the clock, it operates as a timer. If you load and run Maelzel's Modern Metronome at the start of your practice session, you can keep track of how long you have practiced. This feature is especially helpful if your teacher has you keep a record of practice hours.

USING THE PROGRAM

Load and run Metronome 64, which is very user friendly. You needn't know anything about computers to operate it. You only need to use the return key, the shift and return combination and the number keys to choose tempo and time signature and to set the clock.

You can make no computerized mistakes in using this metronome; it never insults you with computer jargon like ERROR or REDO FROM START. An arrow simply points to the mode the metronome is in; to move the arrow pointer to other modes, press the return key.

Maelzel's Modern Metronome starts itself in the Tempo mode; the arrow pointer points to the word Tempo. Like the classic mechanical device, Maelzel's Modern Metronome allows you to choose any tempo from 40 to 208 beats per minute.

After you have chosen a tempo, the arrow pointer will automatically jump to the Beat mode. If you wish to pass over the Tempo mode without making any changes, press the return key.

The metronome will then be in the Beat mode, from which you select a time signature. Press the return key to go on to the next mode without making any changes to the time signature. When you do make a change in the time signature, the arrow pointer will automatically jump to the next mode, which is the Go mode.

In this mode, press the return key to start the metronome. The metronome will start running, and the arrow will point to the word Stop. To stop the metronome, press the return key.

The arrow pointer will now jump back to the Tempo mode. You may again repeat the above steps. But while in the Tempo mode, you may also press the shift key with the return key to jump to the Clock mode. There, you may either set the clock or press the return key to pass over this mode into the last one, which is Quit.

Use the shift and return keys together to stop the program. If you press the return key, you'll pass over this mode, and be back in Tempo. Note that you can jump up to the Clock and Quit modes only from Tempo. R

Gradebook 128

By Frederick Goddard

RUN It Right

C-128 (in 80-column mode)

Gradebook 128 is a classroom-records management program for teachers that makes extensive use of screen windows for menus, help messages and prompts. As a result, you needn't refer to a user's guide, nor know much about computers.

This program probably has more features and is more user friendly than any other similar program available. It is written for the C-128's 80-column mode, but you don't need an RGBI monitor; any monochrome monitor will do.

Gradebook 128 allows you to enter up to 250 student records per file and up to 99 grade columns per student record. Once entered, the files may be saved to or retrieved from disk. Other features let you:

- Print out your grade files, class rolls or analysis of grade distribution on any letter-quality or dot-matrix printer.
- Correct or edit a student record.
- Add or delete records.
- Sort files by name or grade average.
- Enter new grades by number or letter.
- Average grades using a straight or weighted average, with the option of including or ignoring zero grades.
- Modify a grade column by adding, subtracting, multiplying or dividing by a constant. You may use the results of the modification to create a new grade column, to replace an existing column or to add the results to an existing column.
- Delete one or more of the grade columns.
- Perform an analysis of grade distribution and get a screen display showing the mean, the standard deviation and a bar graph of the distribution. This analysis may also be printed (including the bar graph) as a hard copy on any printer.
- Change numerical values of letter grades and assign letter grades based on numerical average for each student.

After you load and run Gradebook 128, the program asks you to select one of four printer options: Commodore 1525; Epson/Graftrax; Smith Corona TP1; or Other. If you have a Commodore or Epson-compatible printer, choose one of the first two options. For all other printers, the last option generally gives the best results. (Some printers, such as the TP1, do not have the full set of 95 ASCII characters; the TP1 selection is for these printers.)

Next, the program asks you for a secondary address. Some printer interfaces, such as Cardco, require secondary addresses; if this is the case with your printer, enter the required address. Otherwise, press the return key to default to a zero secondary address.

Finally, the program asks if you are using continuous form-feed paper. Answer y or n. If you select n, the program will pause output before each page to allow you to position the paper, and will provide for a one-inch margin at the top of each page.

Once the printer selection is complete, the main menu is displayed at the top of the screen.

MENU OPTIONS EXPLAINED

To select any of the menu's 15 options, press the indicated key. You'll then be provided with prompts that correspond to that function. To scroll through your students' names and grades, use the cursor keys. The f7 key is the stop-and-return-to-menu key. Pressing f7 whenever you are prompted for input will return the program to the main menu.

If you select menu option N to start a new file, you'll be asked for a filename and size. For size, enter a number between 20 and 250. This number does not have to be exact, but should be large enough to allow for all the students whose records you expect to enter into the new file.

Gradebook 128 is designed to use all the available memory for your file, and the number of grade columns available depends on the number of student records in the file. For files with fewer than 60 students, the maximum number of grade columns is 99. For larger files, the number of grade columns is less; with 250 students, you will have only 25 grade columns available.

When you save a file to disk, in addition to the names and grades, Gradebook 128 also saves the actual number of students and grade columns used, along with the numerical values for letter grades you have selected. Each time you load a file from disk, Gradebook 128 reallocates memory so that space is allowed for the number of students in the file plus 5; the remaining memory is allocated to grade columns.

After you determine the filename and size, you are prompted to enter each student's name in a last name-first name format, separated by a comma. When you've entered all the names, press the f7 key.

After you create a new file, it is displayed below the main menu in 15 rows, with seven grade columns. Each student is assigned a number according to his or her order in the file. You may scroll the display up or down using the cursor keys. When you have entered more than seven grade columns, you may scroll through them horizontally. Holding down the cursor key causes a fast scroll.

Also, you may use menu option G to go to a particular row or grade column in a large file. Press G, then enter R and a row number, or C and a column number, and press the return key.

When entering a student's grade with menu option E, you may enter it as a number or a letter. If you enter a letter grade, the program will assign to that student the numerical equivalent.

You may change the values of the numerical equivalents for each letter grade by using the letter grade Values function. The same function allows you to assign letter grades based on the numerical values you have entered and each student's average grade.

Each student is assigned the highest letter grade consistent with his or her numerical average. (If a student has a zero average, he or she will not be assigned a letter grade, and if a particular letter grade is given a value of zero, that letter grade will not be assigned to any student.) These letter grades will be displayed and printed out with the student's numerical average.

After you've entered the students' names and grades, you may correct or edit a name or grade by selecting menu option C. Menu options + and – let you add or delete a student record. (If you want to add more than five records, you must save the file to disk after adding the first five; then reload it, add the next five, and so on. This is because the memory space allocated to student records allows for only five more than the number you entered when the file was first loaded or created.)

The R (reorder) menu option allows you to sort your files in several ways. When you enter R, the screen prompts you to enter either a number from 1–20 or the letter G. Entering the default value of 1 will alphabetize the file, beginning with the first letter in each name field. This is the sort most often used. Entering G causes the file to be sorted by average grade, with highest grades first.

It is possible to sort alphabetically, but beginning with a character other than the first in the name field. For instance, I often enter a student's name as a four-digit student-registration code, followed by the last name, then the first name. Thus, I can sort according to student code by entering a 1 to define the sort field; or I can sort the names alphabetically by entering a 5 to define the sort field.

Menu option A allows you to average grades using a straight average, a weighted average or an average with the lowest grade dropped. You may also elect to include or ignore any zero grades. If you select a weighted average, you'll be prompted to enter the weight for each grade column. These weights will be proportionately adjusted in the case of zero grades if you've chosen to ignore them.

If you choose menu option M, you will first be asked whether you wish to modify or delete a grade column. If you choose the latter, you'll be prompted for the number(s) of the column(s) you wish to delete. You enter these numbers in the same way you enter line numbers to list a Basic program. For example, entering 4-7 deletes these columns, and column 8, if there is one, becomes column 4, and so on.

If you want to modify a grade column, you'll be asked which one and the type of modification. You can modify any grade column by adding, subtracting, multiplying or dividing by a constant. Once you've selected one of these mathematical modifications, you're prompted to determine its use. You may choose to put the results in a new grade column, replace an existing column or add the results of the modification to an existing column.

This function allows you to enter raw scores and then use the Modify option to calculate the adjusted grade according to your formula. (Gradebook 128 limits grade values to the range between −99 and 999.)

The grade-distribution function (menu option D) allows you to calculate the mean and standard deviation of any grade column and to display a bar graph of the distribution. For the bar graph, you must enter the grade column, the grade interval and the top grade.

Menu option P allows you to print either grades or class rolls. If you elect to print rolls, Gradebook 128 will print the names from your file followed by 12 blank grade columns. You may use these rolls for recording attendance, and so on. If you select to print grades, the program will print the students' names followed by the average grade, letter grade, and ten grade columns. You'll then be prompted to select the grade columns you want printed.

Enter column number(s) in the same manner you would select Basic program lines to be listed—for example, 5, 1–, 8–16 and –25 are all legal entries. Only ten columns can be printed at a time, since no more will fit on standard eight-inch-wide paper.

Gradebook 128 also allows you to choose between printing from memory or disk. If you elect to print from memory, the file currently resident in memory is printed. If you elect to print from disk, you're prompted to enter the filenames of those files to be printed. After you've done this, Gradebook 128 will load each file in turn and print out the grades or rolls—go have a coffee break.

I hope you enjoy using Gradebook 128. I'm sure you'll find that it saves you a lot of time in handling your class records. R

Please send me back issues of ReRUN

____ Spring Edition
____ Gamepak
____ Summer Edition
____ Fall Edition
____ Productivity Pak (Disk only)
____ Winter Edition
____ Jan./Feb. 1986 (Disk only)

____ C-64
____ VIC-20

____ Cassette version(s) at $11.47*
____ Disk version(s) at $21.47

** Prices include postage and handling. Foreign Air Mail please add U.S. $1.50 per item and $20 per subscription. Pre-payment only.*

□ Payment Enclosed □ MC □ VISA □ AE

Card # __________ Exp. Date __________

Name __________

Address __________

City __________ State __________ Zip __________

Signature __________

ReRUN • 80 Pine Street • Peterborough, NH • 03458

BEAT THE RUSH!

Please send me:

□ **1 year (6 issues) for $89.97**

□ **May/June ReRUN disk for $21.47.***

****Available in June.***

Programs run on C-64 and C-128 (in C-64 mode).

Price includes postage & handling. Foreign Air Mail please add U.S. $1.50 per item and $20 per subscription. Pre-payment only.

□ Payment Enclosed □ MC □ VISA □ AE

Card # __________ Exp. Date __________

Name __________

Address __________

City __________ State __________ Zip __________

Signature __________

ReRUN • 80 Pine Street • Peterborough, NH • 03458

10 RUN *Programs Included on this Disk:*

Word Processing ▻ Disk Utilities ▻ Music ▻ Programming Aids ▻ Education

From the March RUN:

- ▻ RUN Script 64, Part 1
- ▻ Create Your Own Keypad
- ▻ ML Perfect Typist

From the April RUN:

- ▻ RUN Script 64, Part 2
- ▻ C-128 AutoMenu
- ▻ AutoMenu 64
- ▻ Songfest
- ▻ AutoBoot
- ▻ Automatic Line Numbers

PLUS Bonus Programs:

- ▻ Modern Metronome 64
- ▻ C-128 Gradebook

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 Pine 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.

CW COMMUNICATIONS/PETERBOROUGH