

Special Edition

REF RUN

PRODUCTIVITY PAK II

RUN's Most Productive Programs

For the C-64 and C-128

New Updates Plus Brand New Releases!

MENU * RUNSCRIPT * REM
INDER * GRADEBOOK * M
ENU 64 * RUNSCRIPT * 165
0 AUTODIAL * 1670 AUTOD
IAL * RUNTERM PLUS *
LOGO FOR KIDS * CONST
RUCTION SET * DISK BAC
KUP * DATAFILE * DFPRIN
T * DFCALC * MENU * RUNS

Introduction

Welcome to RUN's Productivity Pak II, 1986

Well, here it is—our second annual Productivity Pak disk! Last year's edition was a huge success, so we decided to give you another special ReRUN disk devoted specifically to productive application uses for your C-64 and C-128. This Productivity Pak disk is *packed!*

If you're a regular reader of *RUN* magazine, you may recognize some of the filenames on the directory page. We asked some of our best programmers, who have in past years written top-notch applications for *RUN*, to update and improve their creations for this ReRUN disk. You should be pleased with the work that they've put into these new versions. Of course, Productivity Pak II also includes new programs that have never before been published.

Let's begin by taking a look at RUN Script 2.40, by Robert Rockefeller. He has added dozens of new features to his C-64 word processor, and, by special

request, has written a C-128 version for us. C-128 users will like the 80-column display and faster loading times from the 1571 disk drive, as well as the graphics capabilities. Both C-64 and C-128 versions have powerful features that will keep you hoping for months to come.

The C-64 database management system, Datafile, has been completely updated and improved for this ReRUN disk by author Mike Konshak. Ever since *RUN* published the program in 1984, Mike has been busy providing excellent quality updates and service for Datafile users, and now he surpasses himself.

RUN's telecommunications program for the C-64, Runterm Plus, has become very popular with 64 users. Written by Robert Sims, this powerful and easy-to-use terminal program makes telecommunicating a breeze. For this ReRUN disk, we've tapped the resources of *RUN*'s new telecommunications columnist,

David Bradley, to write autodialing routines that give Runterm Plus even more flexibility.

C-128 users will be interested in Reminder 128, a new program that serves as a useful desktop accessory to help you keep track of your appointments. This program creates a separate file on disk, so be sure to copy Reminder 128 to another disk before running it!

Gradebook 128 is a never-before-published classroom management tool for teachers that received lots of applause as a bonus program on our March/April ReRUN disk. It's one of the best programs available anywhere for handling your grade-keeping and analysis chores.

Education is one of the top home applications for C-64 users, and we at *RUN* provide strong support in this area. Youngsters will enjoy learning the basic principles of LOGO programming with this expanded version of Turtle-Tutor for Tykes, otherwise known as Elmer.

For you adventurers in programming who have lacked the expertise to create multiple graphics screens that can be accessed and scrolled at will, take

a look at Construction Set, making its debut on this ReRUN disk. This graphics construction tool gives you the ability to create high-quality, sophisticated scrolling graphics screens for your own programs.

Productivity Pak II has programs and applications that you'll be using over and over again. Not only are the applications of superior quality, but they're also those that you are most likely to use on a daily basis. Furthermore, we've included some utility programs, such as Menu 64, Menu 128 and Disk Backup, which will make all of your computing activities much easier. Menu 64 and Menu 128 can be copied onto your disks for easy and quick access to your favorite programs. Disk Backup will let you make backup copies of your noncopy-protected disks, such as this ReRUN disk.

We hope you have fun with Productivity Pak II, and, as a wise man once said, "Don't work too hard!"

Margaret Morabito
Technical Manager
RUN magazine

Directory

| Page | Article | Disk Filename | File Type |
|-----------------------|------------------------------|------------------|------------|
| C-128 PROGRAMS | | | |
| 1 | C-128 AutoMenu | MENU 128 | BASIC |
| 3 | RUN Script 128, Version 2.40 | RS128 | BASIC |
| | | OB.RS128 2.40 | ML |
| | | C128 CHAR SET | ML |
| | | OB.RS NMI | ML |
| 25 | Reminder 128 | REMINDER 128 | BASIC |
| 27 | Gradebook 128 | GRADEBOOK 128 | BASIC |
| C-64 PROGRAMS | | | |
| 32 | AutoMenu (64) | MENU 64 | BASIC |
| 3 | RUN Script 64, version 2.40 | RS64 | BASIC |
| | | OB.RS64 2.40 | ML |
| 34 | Runterm Plus | 1650 AUTODIAL | BASIC |
| | | 1670 AUTODIAL | BASIC |
| | | RUNTERM PLUS | ML |
| 43 | Turtle-Tutor for Tykes | LOGO FOR KIDS | BASIC |
| 48 | Construction Set | CONSTRUCTION SET | ML |
| 56 | Disk Backup | BACKUP | ML |
| 59 | Datafile 3.6 | DATAFILE | BASIC |
| | | DFPRINT | BASIC |
| | | DFCALC | BASIC |
| | | DPJ SSORT | ML |
| | | DPJ DOS5.1 | ML |
| | | DFJ MAIL LIST | SEQUENTIAL |
| | | RPJ MAIL LIST | SEQUENTIAL |
| | | MLJ MAIL LIST | SEQUENTIAL |
| | | DFJ INVENTORY | SEQUENTIAL |

NOTE: All indented file names are loaded from within other programs. Do not load these as stand-alone programs!

How To Load

Loading from Menu

This ReRUN disk provides menu programs for both C-64 and C-128 users. These menus will display all of the programs on the disk and let you run them with a single key press.

To get started, C-64 users should type LOAD "MENU 64",8 and press the return key. When you get the Ready prompt, the menu is loaded and you should type RUN to see a list of the programs on your disk.

Commodore-128 users can simply press the shift and run/stop keys to load and run MENU 128.

Loading from Keyboard

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

C-64:

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

```
LOAD "DISK FILENAME",8
```

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

To load a C-64 program written in machine language (ML), type:

```
LOAD "DISK FILENAME",8,1
```

Some ML programs will run automatically when they are loaded. If not, then you should refer to the article in this booklet for exact instructions.

C-128:

All C-64 programs can be run on the C-128 as long as your computer is in C-64 mode.

All C-128 programs are clearly labeled on the directory page. Your C-128 *must* be in C-128 mode to run these programs.

To load a C-128-mode program, press the F2 key, type the disk filename and then press the return key. When the program has loaded, type RUN.

Making Copies of ReRUN Disks

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

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

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

For your convenience, this ReRUN disk contains a backup program that will copy all of the programs for you. You should plan on making a couple of backup copies of your ReRUN disk and then put away the original in a safe place. Once you've made your copies, you can always free up disk space on one of your copies by scratching (erasing) certain programs. See the article, "Disk Backup," elsewhere in this booklet for further instructions.

ReRUN Staff

Technical Manager/Editor: *Margaret Morabito*

Technical Editor: *Tim Walsh*

Managing Editor/Production: *Swain Pratt*

Copy Editor: *Peg LePage*

Proofreader: *Harold Bjornsen*

Design and Layout: *Karla M. Whitney*

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

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. ☐

RUN Script 128 and RUN Script 64: Version 2.40

By Robert Rockefeller

RUN It Right

C-64 or C-128; disk drive; printer

RUN Script 128 and RUN Script 64 (version 2.40) are enhanced versions of the RUN Script 64 word processor (version 1.0), which appeared in the March and April 1986 issues of *RUN*. All the features that were in version 1.0 are still present in version 2.40, but many have been improved. In addition, version 2.40 adds 18 new dot commands and about 30 other new functions.

LOADING PROCEDURE

First read carefully the instructions on how to load. C-64 users must load and run RS64, a Basic program that automatically loads RUN Script's machine language

program. Just wait a moment while it loads and runs. Similarly, C-128 users must load and run RS128.

As you're reading this documentation, keep in mind that when two names of keys are separated by a slash, the keys should be pressed simultaneously. When they are separated by a comma, the keys should be pressed one after the other.

EDITING FUNCTIONS

RUN Script 2.40 offers many editing functions, which I tried to keep as intuitive as possible. Learning to use these functions is easy because most of the editing keys work almost exactly the way they do in Basic.

Reading your text on the screen is also easy, since RUN

Script contains a word-wrap feature that keeps a word from being broken when it's too long to fit into the remaining space on a screen line.

RUN Script uses one line at the top of the screen to display prompts, operation and error messages. This leaves 24 lines free for displaying text. About 35,000 bytes are available for storing text in the C-64 version, and about 68,000 bytes in the C-128 version.

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 of this function will abort it.

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.

MOVING THE CURSOR

In RUN Script 2.40, the cursor can be moved anywhere within the text area. Seven keys and key combinations function as cursor controls. These include

the four cursor keys (eight on the C-128), the home key, the CTRL/back-arrow combination and the shift/return combination. 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 or the (shifted) cursor-up key, you can scroll vertically through the text area. By pressing the cursor-right-and-left key, you can move the cursor horizontally.

With RUN Script's word-wrap feature, a given screen line may have anywhere from one to 39 spaces at the end of it. The empty spaces appear as small dots that don't exist in the text area in memory, but are printed to the screen to pad out the line. When moving the cursor horizontally, you'll find it will skip over these small dots.

Shift/return—This key combination moves the cursor to the start of the next line.

Home—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 end of the text.

INSERTING TEXT

There are three ways to insert text with RUN Script:

Shift/INST—Simultaneously pressing the shift key and the INST key inserts one space at the cursor position.

CTRL/I—Pressing the CTRL key and the I key 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 will be displayed on the status line.

Run-stop—When you need to insert text at the start of a long document, CTRL/I and shift/INST are too slow, since either one has to move the entire text area to insert each character. To circumvent this problem, press the run-stop key to insert a block of 200 spaces. To insert more than 200 spaces, 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 inserted spaces. Just position the cursor at the beginning of the block and press shift/run-stop.

DELETING TEXT

The four methods of deleting text with RUN Script are detailed below:

DEL—Pressing the key that's marked DEL deletes the character to the left of the cursor and moves the cursor left one column.

F7—Pressing F7 also deletes one character, but it's the one *under* the cursor, and the cursor does not move.

F8—Pressing F8 deletes a block of text. Position the cursor over the first character of the section of text you wish to delete and press F8. A "delete block" message will appear on the status line at the top of the screen. Then move the cursor to the *last* character of the block you want deleted and press the return key. The block of text will be removed, but not yet lost, in case you change your mind.

The deleted block is copied to a buffer at the top of memory. To retrieve the block, press F6 twice, quickly, before pressing any other key. A "select insert point" message will then appear on the status line. Move the cursor to where the text was deleted and press the return key again. The text will be restored exactly as it was.

F8,F8—Pressing F8,F8 will delete all text from the cursor position to the end of the text area. You will be prompted to answer

y/n before erasing is performed. You can abort both F8 key functions 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 appear. Then move the cursor to the last character of the block and hit the return key. The text will be deleted and a "select insert point" message will now appear. Position the cursor where you want the text inserted, press the return key, and the block of text will appear in the new position.

F6—To copy a portion of the text in memory to another position, press F6 once after positioning the cursor over the first character of the block to be copied. A "copy block message" will appear 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 will be displayed. Move the cursor to the position where you want the block inserted, hit the return key, and the block will appear in the new location.

F6,F6—To insert the contents of the text buffer into your doc-

ument, press F6 twice. After a "select insert point" message appears, move the cursor to the insert point and press the return key. The contents of the buffer will be inserted. This feature allows you to make multiple copies of the same text block or to recover a deleted block as explained above.

SEARCH AND SEARCH-AND-REPLACE

(Note to RUN Script 1.0 users: Unlike RUN Script 1.0, RUN Script 2.40 does not require a null element for the first and last characters in search-and-replace strings.)

F2—To search for a string between the cursor position and the end of the text, press F2 once and a "search string?" prompt will appear. Type in the string you wish to find, then press the return key. If a match is found, the cursor will appear over the match. If no match is found, a "string not found" message will be displayed. After entering a search string, you can search for multiple occurrences of the string by pressing F2 twice in rapid succession.

F4—Use F4 to replace strings within the text area. A search string and replace string will be requested as above, but at each occurrence of the search string, the string will be displayed so

you can elect to replace it or bypass it.

F4,F4—This prompts you to input a search string and a replace string. After you've done this, all the occurrences of the search string from the cursor position to the end of text will be replaced by the replace string.

CHANGING DISPLAY COLORS

To change the colors of the RUN Script screen display, use the following key combinations:

CTRL/1—to change the text color.

CTRL/2—to change the background color. Changes C-128 screen color.

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

CTRL/4—to change the border color (C-64 version only).

MISCELLANEOUS EDIT FUNCTIONS

Return—Pressing the return key when entering text signals the end of a paragraph. When you press the return key, a left-arrow character with a bent-up tail will appear at the cursor position, marking the end of the paragraph.

CTRL/x—This combination interchanges the two characters to the left of the cursor—a fast

method of correcting transposed letters.

CTRL/6—This combination toggles RUN Script 2.40 in and out of Shift Lock mode. When the "***cap***" message is displayed on the status line, all alphabetic characters typed will be capitalized. This enables you to enter capital letters and numbers, without having to use the shift key.

CTRL/=—If you should want to replace the line-padding dots with some other character, press the CTRL and equals-sign keys simultaneously. A "***chr***" message will appear on the status line. Then type the replacement character, and it will automatically replace the dots. Once this is done, you cannot retrieve the dots. The closest you can come is to repeat the command and press the period key, or, if you prefer blank spaces, the space bar.

HELP—(C-128 version only). Moves the cursor to the other text area when the display is split. If it's in area 1, it'll move to area 2, and vice versa.

CTRL/z—Prints five spaces to the screen. This function is useful for indenting paragraphs.

CTRL/u—Enables you to select a new character to be inserted into the text area when you press the run-stop key. Type CTRL/u, then any character. The default character is the space.

F1—Pressing the F1 key after a prompt for a filename activates a screen-read feature.

Somewhere at the start of a document, within the first 256 characters, create a comment with the .cm dot command. Following the .cm command, type the document's filename surrounded by double quotes, such as .cm "0:filename".

When you save the file with the F1,@ or F1,s command, if you press the F1 key after the "filename ?" prompt appears, RUN Script will read the filename from the screen into the input line. This saves you from having to type the filename every time.

THE F1 KEY

When you press F1, a "command ?" prompt will appear. You then enter the letter corresponding to the function (as listed below) that you wish to execute.

d—F1,d selects the device with which all saving and loading will take place. This may be device 8 or 9, the disk drive. The default device is number 8.

s—F1,s saves text to the selected device.

@—F1,@ saves with replace. The old file is automatically scratched before the new file is saved.

I—F1,I loads a text file from the disk.

a—F1,a is an append function for combining a text file on disk with the one in memory. The file coming from disk is appended starting at the cursor position and replaces any text that follows the cursor. (This append function is not intended for use with sequential files.)

t—F1,t selects whether characters will be output and input in ASCII or Commodore ASCII. RUN Script 2.40 uses Commodore ASCII as its default mode. When you press F1,t, the prompt "use true ascii (y/n)?" appears. True ASCII is used by most non-Commodore printers, and some of its values are different from the ASCII values Commodore printers use.

Also, when telecomputing with mainframes, you may find that the mainframe uses true ASCII. In this case, RUN Script can do conversions between the two types of ASCII, using the w and r commands (see below).

w—F1,w writes the text in memory to disk as either a sequential or a user file. If you want to save it as a sequential file, you must append the string ,s,w to the filename when entering the filename at the prompt. If you want to save the text as a user file, it's necessary to append ,u,w to the filename.

With the w function, you can write the text to disk using either

Commodore ASCII or true ASCII, depending on how RUN Script is configured. (See the **t** function, above.) Also, if you change file types between a read and a write, you need to rename the file. Finally, you can do a save-with-replace by inserting **@0:** before the filename. This is not recommended with the 1541, because of its DOS bug.

r—F1,r loads a sequential or user file from disk. Just enter the filename at the prompt. It's not necessary to append anything to it. If RUN Script 2.40 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.

\$—F1,\$ lists the disk directory. You can pause directory scrolling by pressing the Commodore key and stop the scrolling by pressing the run-stop key. To return to Edit mode, press the CTRL and Commodore keys simultaneously.

>—F1,> issues a disk drive command.

<—F1,< reads the disk error channel.

f—F1,f reports the number of free bytes remaining.

x—F1,x exits to Basic.

p—F1,p prints text. After selecting the print function, you'll

be asked how many copies to make. You can choose up to 99. 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 disk, output begins immediately and continues until all the text has been output. If you don't have a printer, you can output your text to disk and then borrow a printer to print it out.

If you select the printer or screen as the output device, you can output one page at a time. Simply answer **n** for no at the "continuous (y/n) ?" prompt. A "next output (c/p/s) ?" prompt will then be displayed before each page is output.

If you enter **s** for screen at this point, the next page will be printed to the screen. If you enter **p** for printer, the next page will be sent to your printer. If you enter **c** at the prompt, output from that point onward will be continuous, and the "next output (c/p/s) ?" prompt will no longer appear.

Output to the screen lets you see where the page breaks occur. I recommend that you run through your text this way before printing it. When printing text to the screen, you can pause the output by pressing the Commodore key.

You can terminate output at any time, regardless of the output device, by simultaneously pressing the CTRL and Commodore keys.

O—F1,0 loads a new character set. In Special Graphics mode, the business-graphics characters of this set would be printed using the printer's Dot Graphics mode.

1-9—F1,1 loads a new print set number 1. F1,2 through F1,6 load new print set numbers 2-6. F1,7 through F1,9 load new print set numbers 7-9, but work only in the C-128 version.

Cursor-up and **cursor-down**—F1 with the vertical cursor controls selects the Fast Scroll mode, which rapidly moves the text up or down. You leave this mode by pressing any other key, or abort it with the Commodore and CTRL keys.

Cursor-left and **cursor-right**—F1 with the horizontal cursor controls selects the Page mode. In this mode, pressing the cursor-right key scrolls the text upward 24 lines to display the next screen. Pressing the cursor-left combination scrolls the screen downward 24 lines to display the previous screen. Exit this mode by pressing any other key, or abort it by simultaneously pressing the Commodore and CTRL keys.

c—F1,c changes the case of all text from the cursor position to the document end. Uppercase

letters become lowercase, and vice versa.

R—F1,R saves a copy of the table of redefined characters to the current save device. First, re-define characters with the .dc command (see the dot commands, below); then print the file to allow the .dc commands to execute; then save the table. Now you can load the saved characters at any time with the .lr command.

o—F1,o prints a document with one copy, output to printer device 4 and continuous output.

Z—F1,Z (C-128 version only) swaps text memory in RUN Script's text area in bank 1 with the memory in bank 0. Pressing F1,Z again restores the original text. You can keep one text file in bank 1, one in bank 0, and switch between them with F1,Z. This permits up to 68000 bytes of text to be stored in memory.

M—F1,M selects the macro characters that should be turned off before printing the left margin, a header or a footer. This prevents problems such as underlining in the left margin. RUN Script 2.40 expects the same macro letter to be used to activate and deactivate printer functions. For example, if underlining is turned on with uppercase U, a lowercase u is needed to cancel it, and vice versa.

At the prompt, just enter the macro characters—up to 13 of

them—that need to be cancelled. Only enter the macros that activate functions.

m—F1,m loads a set of macros as defined with the Define Macros program (see discussion of this below). If you're using a C-128, you must run Define Macros in C-64 mode, not 128 mode. However, the macros table it creates can be used with the C-128 version of RUN Script.

g—F1,g enables and disables the Special Graphics mode.

G—F1,G selects the type of printer. If you have a MPS-803 or compatible, MPS-801, 1515, 1525 or MPS-1000 (Commodore mode), you answer yes at the prompt. Answer no if you don't have one of these printers. This function determines how RUN Script will print dot-graphics characters.

C—F1,C selects the secondary address used to print dot graphics and sends macro command strings to the printer. Unlike the .ca dot command, this function changes the permanent secondary address inside RUN Script.

S—F1,S splits the RUN Script text display into two separate areas. Type F1 then S, and the prompt "how many text areas (1/2) ?" will appear. If you answer 1, RUN Script will remain unchanged.

If you enter 2, the prompt "size of area two ?" will appear. You may choose any number from 1

to 18, and text area 2 will be allocated that many kilobytes of memory. Since all of the current text is erased before the splitting occurs, there is a third prompt in this function, "erase text (y/n) ?". If you enter y for yes, the text area will split, and the cursor will appear at the start of text area 1.

RUN Script can automatically load a text file into text area 2—to display a help file, for instance—when you first boot up the program. Another possible application is to load in a list of items, from which you could quickly locate items with the fast scroll.

You'll have to create your own help file, but if you want to make use of the auto-load feature, just split the text area with F1,S, execute any other configuration commands you want, then save RUN Script with the F1,q command. Make sure you name the file OB.RS128 2.40. Then set the string variable HLP\$ in the boot program equal to the name of your help file.

A—F1,A provides movement between the two text areas. At the prompt, enter either 1 or 2, depending on which area you wish to access.

T—F1,T selects the secondary address used to print text to the printer. Unlike the .ta dot command, this function changes the permanent secondary address inside RUN Script.

L—F1,L specifies whether a

line-feed character should be printed after each carriage-return character. Unlike the .lf dot command, this function changes a permanent flag inside of RUN Script.

q—F1,q saves a copy of RUN Script, including any printer macros, the current color settings, the current settings from the F1,S; F1,T; F1,G; F1,C; F1,L; F1,M; F1,d; F1,t; F1,g; CTRL/= and CTRL/u commands, and the settings from the .gc, .gb, .ge, .bs and .ff dot commands. To customize RUN Script 2.40, use these commands to configure it, then save it with F1,q.

DEFINING PRINTER MACROS

Note: The following discussion refers to a Basic program called Define Macros, which is not on this Productivity Pak II disk. If you are interested in the features it makes possible, you can obtain the Define Macros program from the April 1986 issue of RUN (p. 46) or the March/April edition of ReRUN.

Many Commodore owners use non-Commodore printers such as the Okimate 10 or Epson MX-80. These printers often have desirable features, like italic character sets and the ability to do underlining. The printer-macro feature of RUN Script enables you to customize your copy of

this word processor so you can take full advantage of whatever capabilities your printer may possess.

Printer macro character strings are always sent to the printer when output is to the screen, rather than being ignored. This is necessary when output is being switched between the screen and printer. An idiosyncrasy of RUN Script 2.40 is that the printer must be turned on when output to the screen is taking place, because a file is always opened to the printer when you select output to the screen. If RUN Script ever seems to "hang up" mysteriously during a printout, check your printer.

You may select any upper- or lowercase alphabetic character to be a macro character. You then create a table of printer macros with the Define Macros program. 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 and 88) to start printing double-width characters. With Define Macros, you can select a character—D, for instance—to represent this two-character string. Then, when D is encountered

during printing, the decimal sequence 27,88 will be sent to the printer to produce double-width characters. You could define another character, perhaps d, to represent the sequence to stop printing double-width characters.

This macro feature is most useful for printing titles and sub-headings. To create a double-width heading, first place the cursor in front of the heading, then press the F3 key. A `"*mac"` message will appear on the status line. Next, 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 nonalphabetic character, the operation will abort.

Your table can consist of 52 different macro definitions, each of which can be from one to 20 characters long. I've allocated exactly 500 bytes in memory for the complete macro table.

CUSTOMIZING MACROS

Before running Define Macros, make a list of the alphabetic characters you want 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.

First you'll 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 * key.) You'll 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'll have defined one macro. The prompt, "finished all definitions (y/n)?" will then appear. If you have more macro definitions to enter, type n and press the return key.

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, then prompt you to save the table to disk and provide the proper device number.

When you're in RUN Script 2.40, you can easily load in your macro set by pressing F1, followed by m. (See the description

of this procedure earlier in this documentation.)

DOT COMMANDS

RUN Script is a post-formatted word processor. This means that the text is not formatted until it's printed, so your screen display will vary from your printout. To specify output format, RUN Script 2.40 has about 40 dot commands, so called because each command must be preceded by a dot (a period). These commands are embedded in the text to specify margin widths, define headers and footers, and so forth.

A dot command is executed when text is printed, and only text following the dot command is affected. For example, if you don't place the dot commands to set margins until you're halfway through a page of text, the margins in the first half will have the default width when printed.

Four steps must be followed for dot commands to be interpreted correctly. First, the line immediately preceding a line of one or more dot commands must end with a return. Second, the line of dot commands must start in the first screen column. Third, multiple dot commands in a string must not be separated by spaces. Fourth, each string of dot commands must end with a carriage return.

Following is an example of a string of dot commands:

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

The dot commands are:

- .lj**—Left-justifies printed text.
- .rj**—Right-justifies printed text.
- .cn**—Centers printed text between the left and right margins. Your text must begin on the next line below the **.cn** command.
- .pl**—Sets the page length. This command must be followed by a number from 1–240, indicating the number of lines you want to constitute a full page. For standard-size, 8½-by-11 paper 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 the page width accordingly to use an alternate character set. Example: **.pw80**.
- .lm**—Sets the left-margin width. The **.lm** directive must be followed by a number from 1–240. Example: **.lm8**.
- .rm**—Sets the right-margin width. The **.rm** directive must also be followed by a number from 1 to 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 to 240. Example: `.tm6`.

.bm—Sets the bottom-margin depth. The `.bm` directive must be followed by a number from 1 to 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. The command may be followed by up to 255 characters and must end 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 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. The `#` character following the `.hd` (or `.ft`) has special significance. Entered just once at the beginning of a document, `.hd#` automatically prints the page number of each page.

The `.lj`, `.rj`, `.cn`, `.pw`, `.lm` and `.rm` dot commands have no effect on the header. The header string is printed exactly as defined, starting in 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.

You may embed macro characters (see discussion above) in the header string if you wish. The header will be printed at the line position equal to the `.tm` setting plus 1. So, if you set the top margin to 6, the header will be printed on the seventh line. Example: `.hd RUN Script 2.40`.

.ft—Defines a footer to be printed at the bottom of every page. The `.ft` directive works exactly like that of the header. The footer 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. The command 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. You can print one or more blank lines between each line of text. For double-spacing (one blank line between lines of text), set `.ls` to 1. Example: `.ls1`.

.l+—Indents text from the current left-margin setting. For ex-

ample, if the left margin is set with `.lm8` and you execute `.l+3`, text will henceforth be indented as though you'd set `.lm` at 11. To cancel an indent, use `.l+0` or `.l-0`.

.l—“Outdents” text, such as a subheading, to the left of the left margin. 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 `.lm` had been set at 5. Outdents are canceled with `.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 and bottom margin will be printed, and then a new page will be started. Example: `.fp`.

.fl—Links files to be printed. The command must be followed by a device number and a file-name, separated by a comma. When the `.fl` directive is executed, the specified file will be loaded from the specified device and begin printing. The permissible device numbers are 8 or 9 for disk. If two disk drives are used, one document could even slightly exceed 340,000 characters in length. Example: `.fl8, next file`.

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

.lf—Prints a line-feed character

after every carriage return. Some non-Commodore printers require this. Example: `.lf`. (In version 2.40, `.lf1` enables line feeds; `.lf0` disables them.)

.cm—A handy dot command that lets you leave a comment for yourself that won't be printed. For instance, if you're in the habit of keeping all your old letters on disk, you can leave comments to yourself with dates and other information. Example: `.cm July 19, 1985 RUN Script 64 article`.

.r+—Indents the right margin. This code must be followed by a number from 0 to 255. For example, if you set the right-margin width to 10 spaces with `.rm10`, and then execute `.r+5`, the effect is the same as if you had executed `.rm15`. A right indent is canceled by executing `.r+0` or `.r-0`. Example: `.r+5`.

.r-—Makes a right outdent. The command must be followed by a number from 0 to 255. For instance, if you've set the right margin to 10 with `.rm10`, and then execute `.r-5`, the effect is the same as if you had originally executed `.rm5`. A right outdent is canceled by executing `.r+0` or `.r-0`. Example: `.r-3`.

.bj—Stands for “both justify” and prints the ensuing text with both the left and right margins justified. It does this by inserting extra spaces between words so that each line is flush with the left and right margins. You can-

cel the .bj command by executing an .lj, .rj or .cn command. Example: .bj.

.pr—Sends a sequence of up to 98 bytes to the printer using a specified secondary address. The secondary address must immediately follow the command; then the bytes to be sent to the printer, separated by commas, must follow after the secondary address.

Example: .pr2,36,36,36,46,57,57. This example will send bytes 36, 36, 36, 46, 57 and 57 to the printer using secondary address 2. On the MPS-1000 printer, this would set up a formatting string.

Example: .pr5,27,69. Here, bytes 27 and 69 will be sent to the printer using secondary address 5.

.ta—Stands for "text address," and must be followed by a number from 0 to 31. The number is the secondary address that will be used to print the ensuing text, and it lasts for only one printout. You can set the default (permanent) text secondary address with the F1,T command. The .ta command is useful with printers, such as the MPS-1000, that have features that can be accessed only through special secondary addresses.

Example: .ta1. This example sets the print secondary address to 1. On the MPS-1000, text printed using secondary address

1 is formatted according to a previously-defined format string (which can be sent to the printer with the .pr command).

.ca—Defines a new, temporary secondary address. This is the secondary address that will be used to send macro strings to the printer and to print dot-graphics characters. You can set the default (permanent) secondary address with the F1,C command. Example: .ca5.

.dc—Stands for "define character." This command takes three forms that enable you to redefine any character on the keyboard except @, to a total of 127 characters, then to undefine them. The total is enough to redefine the entire keyboard if you wish. All redefined characters are erased before each printout.

Example: .dca,66. To define a character, follow the .dc command with the character to be redefined, a comma and the new decimal value of the character. The Commodore ASCII decimal value for the letter "a" is 65. Here, the value has been changed to 66, which is the value for "b." Therefore, at print-out time, all the a's in the document will print out as b's.

Example: .dca-. The minus sign following the character undefines that character. This example would undefine the character "a", removing it from the table of redefined characters.

Example: `.dc@`. This would erase the entire table of redefined characters, effectively undefining all of them.

.lr—Stands for "load redefined characters." This command is used only after you've performed three steps. First, you must define a number of characters with `.dc` commands. Second, you must print the current document; third, you must save the table of redefined characters to disk with the `F1,R` command. Then you can use `.lr` to load the saved characters during a later printout. This is useful when you have to redefine many characters—to access special characters on a printer, for example. The `.lr` command must be followed by a device number, a comma and a filename.

Example: `.lr8,filename`. This example would load the file of redefined characters named "filename" from device 8, the disk drive.

.lc—Stands for "load characters," and enables you to load a new character set or a new print set during printout. The `.lc` command must be followed by a number from 0 to 9 (0–6 for a C-64) that specifies the set to be loaded. A 0 loads a new set onto the screen; the numbers 1 through 9 load a new set into the printer. After the number comes a comma, followed by the device number from which the

set is to be loaded, then another comma and the filename of the character or print set.

Example: `.lc0,8,special set`. Here, the new set would go to the screen from device number 8, the disk drive.

.el—Stands for "empty lines," and prints the specified number of carriage returns. If the number of empty lines to be printed is greater than the number of remaining lines on the page, a force page is executed instead.

Example: `.el10`. Here, ten carriage returns will be printed.

.st—Stands for "stop." This command works only with the printer or the monitor, not the disk drive.

Example: `.st`. When not followed by any parameters, the `.st` command terminates Continuous mode during printout and starts Single Sheet mode after the current page is done. The command may be placed anywhere within the text area.

Example: `.st5`. When followed by a parameter, `.st` interrupts continuous output at a specified page number, anywhere from 1 to 65535, and enters Single Sheet mode. In this example, if continuous output has been selected with `F1,p` or `F1,o`, the first four pages will print in Continuous mode, then at page 5 the "next output (c/p/s) ?" prompt will appear.

.po—Stands for "page order,"

and must be followed by three numbers, separated by commas. This command enables you to print on both sides of a page or in a number of columns, as in a newsletter. Any number of columns can be printed across one page—2, 3, 4 or even more.

The `.po` command requires a little advance planning and preparation on your part, and it works only when continuous output is selected. I'll explain through examples.

Example: `.po1,2,1` and `.po2,2,1`. This combination of `.po` settings lets you print on both sides of the page, with text extending across the full width of the page (not multiple columns).

The first setting is for the first pass. The first parameter in this setting sends the first page to the printer or disk; the second parameter specifies that every second page after that will also go to the printer or disk. For example, the odd-numbered pages might be printed and the even-numbered pages not—a phenomenon I call "page cycle." The last parameter in the first setting indicates how many pages will be output before the page number is incremented. It should equal the number of columns across the page.

The second setting is for the second pass. Notice that the last two parameters are unchanged. The new first parameter, how-

ever, now sends the even-numbered pages to the printer. You print on the same paper as in the first pass, but now you use the back side. You can even define different headers or footers for the odd- and even-numbered pages—to place page numbers on opposite sides of the page, perhaps.

Example: `.po20,1,1.st21`. You can combine the `.po` and `.st` commands to print only one page out of a document. In this example, pages 1–19 will not be printed, but page 20 will. After page 20, the `.st` command will bring up the "next output ?" prompt, at which point you can abort the print operation.

.ff—Stands for "form feed." The `.ff` command must be followed by number 0 or 1. If you select 1, the bottom margin will be printed by sending the form-feed character to the printer. The Default mode, `.ff0`, prints the bottom margin with carriage returns. Example: `.ff1`.

.fc—Stands for "force conditionally," and must be followed by a number from 0 to 255. If less than the specified number of lines remain on a page when `.fc` is executed, no more text will be printed on that page. Instead, a force page will be executed, printing the footer (if any) and bottom margin immediately and resuming text output on the following page.

Example: .fc20. If less than 20 lines remain when .fc20 is executed, the page will be forced.

.dg—Stands for "define graphics character." This command lets you change any character, in any print set or the screen character set, without resorting to a character-set editor. Example: .dg0,1,255,0,0,255,0,0,255,0.

The command must be followed by ten byte numbers, separated by commas. The first byte number specifies the set in which the character to be changed is situated. As always, 0 specifies the screen character set, 1–9 specify a print character set.

The second byte is the screen-code value of the character to be changed. The 1 in the example selects the "a" character.

The last eight bytes define the character itself. The sample setting would change the "a" into three parallel bars.

.gc—Stands for "graphics characters." The .gc command is followed by two parameters, separated by a comma. The parameter settings are saved within RUN Script, and are permanent until the computer is turned off, or until you execute another .gc command. Example: .gc1,6. This is the default setting, which is correct for the MPS-803 printer.

The first parameter is a number, 0 or 1, that specifies the orientation of dot-graphics char-

acters printed when you select either a print set or Special Graphics mode. Only one setting is correct for any one printer. If you select the wrong setting, all characters printed in Graphics mode will be printed upside down. The only way to determine the correct setting for your printer is by experimentation.

The second parameter sets the width of Graphics Mode characters in dots. Standard Commodore characters are eight dots wide. So, if you set this parameter to 6, only the first six dots of each character will be printed. If you set this parameter to 10, ten dots will be printed—eight dots of character and two dots of space.

Many non-Commodore printers accommodate three dot densities in Dot Graphics mode: single density with 480 dots per line, double density with 960 dots per line and quadruple density with 1920 dots per line. If you wanted to print 80 characters per line, how many dots wide must each character be for each of these densities?

Single Density mode: $480 \div 80 = 6$ dots wide.

Double Density mode: $960 \div 80 = 12$ dots wide.

Quadruple Density mode: $1920 \div 80 = 24$ dots wide.

Obviously, none of these are correct. To use Single Density

mode, you need a print set only six dots wide. If you use Double Density mode, you can print all eight dots of each Commodore character, but four dots spacing will remain between each dot-graphics character.

You can use double-density with a dot width of ten if you're using the elite (96 characters per line) character set for printing normal text. There'll be two dots of space between each dot-graphics character, but you may find this acceptable.

Quadruple Density mode will not work properly, regardless of characters per line.

To make use of the full eight-dot width of a Commodore character and print 80 characters per line, you need a printer with a dot density of $8 \times 80 = 640$ dots per line. Use a dot width of eight if you have such a printer.

In general, use the following formula to calculate the correct dot width: $\text{dot width} = \frac{\text{dots per line}}{\text{characters per line}}$. If the result is a dot width of less than eight, you'll need a print set that uses only that many columns of a Commodore character.

There's a problem with printing dot-graphics characters with MPS-803-compatible printers. As I said, standard Commodore characters are eight dots wide, and they're also eight dots high.

Unfortunately, the MPS-803, 1515, 1525, MPS-801 and MPS-1000 (Commodore mode) need characters that are only six dots wide and seven dots high. There's no way around it—they need special print sets that use only the left six dots and the top seven dots of Commodore characters. Hopefully, *RUN* readers who design such print sets will share them with others.

.gb—Stands for "graphics begin," and defines the string of bytes that is sent to the printer to activate Graphics mode. The string is sent before each character is printed in Dot Graphics mode. Example: `.gb8`.

The MPS-803 and its compatibles require just one character, `CHR$(8)`, to activate graphics mode. This character will put the printer into Graphics mode until it receives a byte value of less than 128, at which point it will return to normal Text mode. Most other printers handle graphics differently, however, and require that more than one character be sent.

Example: `.gb27,75,6,0`. These four characters will tell the printer to print the next six bytes in Dot Graphics mode. Then *RUN* Script will send the left six dot columns of a character.

Remember that the string must tell the printer how many bytes

of graphics data will be sent. If the dot width is set to 6, six bytes will be sent; if the dot width is set to 10, ten bytes will be sent. On most printers (not the MPS-803), if the dot width is changed, you'll have to redefine `.gb` also.

Example: `.gb27,76,8,0`. This example selects the Double Density Graphics mode. With the dot width set to 8, dot-graphics characters will print with a density of 120 characters per line.

.ge—Stands for "graphics end." The string of bytes following the `.ge` command will be sent to the printer after each dot-graphics character. Some printers require this command to return to Text mode.

Example: `.ge15`. This setting, the default, is correct for MPS-803 printers.

Example: `.ge`. This is the correct setting for most non-Commodore printers. No bytes will be sent after the graphics bytes.

ERROR MESSAGES

Macro Not Defined—An undefined macro character has been encountered. This usually means that you made a typing error when entering the macro character. The cursor will rest on the incorrect character.

Macro characters are a means of customizing RUN Script to take advantage of special features of non-Commodore printers. I explain macro definition

later in this documentation. However, it's not necessary to understand macro characters to use RUN Script.

Illegal Quantity—A dot command's numeric argument is too large or too small. The cursor will appear near the illegal number.

Output Error—A hardware error has occurred during printing. If output is to the printer, the error could mean the printer is out of paper or not turned on.

Syntax Error—RUN Script is unable to recognize a dot command. The cursor will be near the offending command.

Margin Error—The margins have been set to illegal values. Check the dot commands preceding the cursor position.

Text Area Full—The text area is full. At this point, you must save your text and erase what's on the screen to continue.

Device Not Present—The device referenced in the input/output operation does not respond. Check to see if you used the correct device number and make sure the device is turned on.

File Not Found—The file you attempted to load was not found in the load device.

String Not Found—A match was not found during a string search.

Save Error—A status error occurred while the program was attempting to save text.

Load Error—A status error oc-

curred while the program was attempting to load a text file.

PRINTERS

RUN Script is set up to expect a Commodore printer connected to the serial bus. A combination of a non-Commodore printer and an interface that emulates a Commodore printer will also work.

If your system includes neither of these, you can still use RUN Script by putting the printer interface into Lock mode and using the RUN Script F1,t function to output true ASCII. Another method is to use the Define Macros program (see previous discussion) to create a customized set of macros that RUN Script can use to control your printer.

Note that you cannot use an RS-232 printer with RUN Script. If you have such a printer, you must output your text to disk and then use a Basic program to print it.

DAISYWHEEL PRINTERS

RUN Script 2.40 can do underlining and double-strike characters with daisywheel printers that recognize the back-space character. Type F3, then the (character to start underlining; type F3, followed by) to end underlining. Type F3, then [to start double-strike; type F3, followed by] to end double-strike. (See also the .bs command.)

To change the print wheel,

type F3, then *. When RUN Script encounters the reversed * character, it will stop until you press a key.

GRAPHICS MODE

Most dot-matrix printers can print dot-addressed graphics and ordinary text on the same line, and RUN Script 2.40 takes advantage of this feature. When you put RUN Script into a graphics mode, instead of outputting normal text, it uses the printer's dot-graphics capability to print each character. This allows printers such as an MPS-803 to print italics, boldface or underlining, and print foreign character sets such as French, Russian or Greek—in fact, almost any kind of text.

There are two ways to create graphics with RUN Script. One is through a "print set," which is half of a normal character set. A character set contains 256 characters, 128 non-reversed and 128 reversed, so a print set contains 128 characters. How does it work?

You select a print set with the F3 key, the same key that selects macro characters. After you press F3, choose a numeral from 0 to 9. The numeral will appear at the cursor position in the text, in reverse field.

Pressing 0 selects normal Text mode, which is the default; pressing 1-9 enables the Graphics

mode. Within that mode, 1 specifies the first print set, 2 specifies the second set, and so on, up to the maximum number of sets your computer can have (six for the C-64, nine for the C-128).

For example, say you chose print set 3 and the next character to be printed is an a. The letter a is the second character in the standard Commodore character set (see the *C-64 Programmer's Reference Guide*, page 376, for a listing of the character set), so RUN Script will go to set 3, take the second character of that set, and print it using dot-addressed graphics.

Only six print sets are available in the C-64 version of RUN Script 2.40, but you can still press the 7, 8 or 9 key after the F3 key. If you press 7 or 8, RUN Script will use the non-reversed characters of the character set as a print set; if you press 9, it'll print the reversed characters.

Print sets and the character set are loaded into memory by the boot program, which also loads the RUN Script machine language program and initializes the RUN Script system.

You also can print graphics with the Special Graphics mode. It's activated by the F1 key, then g, then answering y at the prompt "enable special graphics (y/n) ?". You disable the Special Graphics mode by answering n.

When the Special Graphics mode is functioning, the busi-

ness-graphics characters of the character set, which are accessed by simultaneously pressing the Commodore logo key and any other key, will be printed using the printer's Dot Graphics mode. This occurs only when normal Text mode is selected. All other characters will be printed as normal text characters.

Since RUN Script's character set is used to display text on the screen, this mode has the advantage that characters will appear on the screen exactly as they will be printed. Of course, to get characters other than the Commodore business-graphics characters, you have to alter the relevant characters with the .dg command or with a character-editor program.

Printing RUN Script graphics requires an MPS-803, 1515, 1525, MPS-1000 in Commodore mode, or other compatible printer. RUN Script 2.40 is designed to be used with these printers. However, a 1526 or MPS-802 printer won't work with RUN Script graphics.

If you have a non-Commodore printer that's not compatible with the MPS-803, you'll have to customize RUN Script 2.40 to the printer with the various customization commands available. (See the F1,q command, above.)

For more information about the Graphics mode, see the .gc, .gb, .ge and .dg dot commands I mentioned earlier. ☐

Reminder 128

By Bob Guerra and Jim Richards

RUN It Right

C-128 (in 80-column mode); disk drive

If you have a lot of important dates to remember, you know how useful a desk calendar or pocket date book can be. Reminder 128 is an electronic desk calendar that improves on the pencil-and-paper versions. It eliminates thumbing through pages by letting you store up to 100 dates and then search for upcoming events by typing in the current date.

The first time you boot Reminder 128 and type in the current date, the program automatically creates a relative file called REMFILE for storing your reminders. Once this has been done, you can press any key to access the main screen and search the file. Since the file will be new at this point, the program will say you have no messages. A menu above the message area will display your options: Add, Delete, View, Sort, Print and Exit.

To make a selection, use the left and right cursor keys to move the highlight onto the option you want and press the return key. The highlight even wraps around from one side to the other.

ADDING REMINDERS

The first thing you'll want to do is add some reminders to the file. Select Add from the menu and enter the event date at the prompt. To help prevent typing in invalid information, the program accepts at this point only numeric input and real dates. For example, you can't list the date of someone's birthday as 11/31/86, since November has only 30 days; or if you try to schedule an appointment for February 29, 1987, the program will remind you that February has only 28 days in 1987. Try making it 1988 (a leap year), however, and it'll work fine.

After you type in the event date, you must specify the number of days in advance that you want to be alerted to the event. Each time you use Reminder 128, the pro-

gram retrieves only those events that are upcoming within the specified number of days.

Next, type in a message of up to 56 characters (with no commas or colons). Messages can be either one-time reminders that are automatically deleted once the date has passed, or annual reminders, for birthdays and such, that are automatically updated for the next year and written back onto the disk. To designate a reminder as annual, all you have to do is begin the message with an asterisk.

Once you've typed several reminders into your file, you can check to see if they're really there with the View option on the main menu. This displays all the reminders in your file along with their record numbers. It's a good idea to keep your reminder file on a backup disk.


OTHER OPTIONS

If you decide to eliminate a reminder, do it with the Delete option. Once you've typed in the record number to tell the program which reminder to erase, that reminder will appear on the command/menu line at the top of the screen. To proceed with the deletion, press the return key. However, if you have sec-

ond thoughts you can abort the operation by pressing the escape key. To use file space efficiently, record numbers that have been freed by the delete process are the first ones filled when you add more reminders.

The Sort option arranges all the reminders in REMFILE in ascending chronological order. Although the program can find the reminders for any given day regardless of their order in the file, I find it reassuring to see them printed out in chronological order. If you don't care, you can forget about the Sort option.

If you want a hard copy of your reminders, select Print from the menu. A small pull-down menu will appear, offering the choice of "Today's," for a print-out of today's reminders only, or "All" for a listing of your complete reminder file. To move the highlight from one selection to the other, use the up-and-down cursor key, and then press the return key to begin printing.

When you're done using Reminder 128, leave the program by selecting Exit from the menu and answering Y to the prompt, "Leave Program, Are You Sure?" If, at the last instant, you remember an upcoming appointment you forgot to add, enter an N to return to the menu. 

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 re-allocates memory so that space is allowed for the number of students in the file plus 5; the re-

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 en-

tered 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,

| STUDENT NAME | AVERAGE | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|----------------------|---------|----|-----|----|----|----|----|----|----|-----|----|----|
| 1 ABBOTT, LOUIS | 73.6 | C | 49 | 95 | 79 | 79 | 89 | 79 | 49 | 49 | 79 | 49 |
| 2 ADAMS, BETTY | 72.3 | C | 68 | 68 | 59 | 29 | 63 | 66 | 64 | 68 | 89 | 58 |
| 3 ALBERTSON, EDWARD | 71.7 | C | 56 | 38 | 39 | 96 | 59 | 25 | 77 | 95 | 79 | 78 |
| 4 APFLEBEE, ELLEN | 88.6 | A- | 100 | 65 | 65 | 98 | 98 | 59 | 99 | 100 | 98 | 98 |
| 5 BAKER, DARLENE | 62.0 | D | 49 | 59 | 68 | 59 | 49 | 46 | 59 | 93 | 68 | 68 |
| 6 BJORNSEN, HAROLD | 77.6 | C+ | 98 | 49 | 67 | 38 | 64 | 68 | 69 | 84 | 92 | 93 |
| 7 BLAKEMAN, KEN | 79.3 | C+ | 26 | 76 | 49 | 94 | 99 | 49 | 98 | 99 | 87 | 98 |
| 8 BOYD, DENNIS | 79.6 | C+ | 93 | 29 | 92 | 94 | 87 | 66 | 68 | 98 | 69 | 68 |
| 9 CANTERBERRY, FRED | 79.4 | C+ | 98 | 48 | 89 | 68 | 67 | 95 | 22 | 97 | 81 | 72 |
| 10 CLEAVER, THEODORE | 78.3 | C+ | 56 | 98 | 98 | 49 | 59 | 68 | 37 | 68 | 95 | 94 |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |

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. ☐

Auto Menu

By Joe W. Roche

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 | MENU 128 | N | LOGO FOR KIDS |
| B | RS128 | O | CONSTRUCTION SET |
| C | OB.RS128 2.40 | P | BACKUP |
| D | C128 CHAR SET | Q | DATAFILE |
| E | OB.RS NMI | R | DFPRINT |
| F | REMINDER 128 | S | DFCALC |
| G | GRADEBOOK 128 | T | DP] SSORT |
| H | MENU 64 | U | DP] DOS5.1 |
| I | RS64 | V | DF] MAIL LIST |
| J | OB.RS64 2.40 | W | RP] MAIL LIST |
| K | 1650 AUTODIAL | X | ML] MAIL LIST |
| L | 1670 AUTODIAL | Y | DF] INVENTORY |
| M | RUNTERM PLUS | Z | |

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's Great Communicator— Runterm Plus

By Robert Sims

RUN It Right

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

Runterm Plus is a full-featured terminal program that now has auto-dialing capabilities. There are two boot programs that will now automatically dial your modem and load and run Runterm Plus.

These boot programs are written for the Commodore 1650 and compatible modems as well as for the 1670. When you run either one, you'll be prompted to type in the telephone number of the on-line service you wish to access. These auto-dialer programs have been tested successfully with Uninet and Telenet, but they do not work with Tymnet.

Once you've made a connection, Runterm Plus will automatically load and run. Be sure you

have a copy of Runterm Plus on the same disk on which you have the auto-dial programs.

When you get into Runterm Plus, select from the main menu whichever service you need. The instructions in this article tell you how the terminal program works.

Runterm Plus will communicate with any computer that uses either standard ASCII or Commodore ASCII data format. The program can transfer files by capture buffer, as well as by automatic protocols, which ensure error-free transmission. It also has one feature that lets you view high-resolution graphics and another that lets you play games through your modem.

This program is set up to work with the VIC-Modem, the 1650 Auto-Modem and other modems that are compatible with these.

Runterm Plus includes automatic file-transfer routines and modem game features, as well as the standard ASCII and full-duplex capability that you'll need to call the national networks. You will be able to upload and download program files with this terminal.

From the opening screen, press CMD-M to display the menu of features available.

CONTROL CODES

National services and some local bulletin boards require that you enter a log-on code, which tells the host computer that you are ready to go. CompuServe, for example, waits for a Control-C (ASCII 3) or a Return. If you are calling Delphi via Tymnet, you'll be asked to enter your terminal identifier (you type A).

In addition to the log-on code, you will need to send and receive other special control codes. The most common codes are composed of the first 27 characters in the ASCII character set—Control-A through Control-Z and Escape.

To generate these codes with Runterm Plus, hold down the CTRL key and press the appropriate key. CTRL-A sends a binary 1, CTRL-T (or the delete key) sends a binary 20, and so on. To send the escape character, hold down the CTRL and press the colon key. (ESC is often used on CompuServe.)

You should also be aware that if you press the home key, a CTRL-S (pause) is sent to the other computer. If you hit this key accidentally, then you should send a CTRL-Q (press the cursor-down key) to resume transmission. Pressing the run/stop key will send a CTRL-C, which is used as a cancel code on many systems.

OTHER SPECIAL KEYS

In the process of communicating with another computer, you will need to perform several auxiliary operations, such as checking the disk directory, capturing data in a buffer and saving it to disk, or preparing to receive a file. These operations are performed using local commands, generated via a combination of the Commodore logo key and letter keys.

For example, you can read the disk directory by holding down the Commodore logo key and pressing the D key. As it does for all local commands, Runterm Plus sends a CTRL-S (ASCII 19) to the other computer; this puts it on hold, so incoming data won't get mixed into the directory contents. The program then retrieves the disk directory and displays it to the screen.

When the last byte of the directory is displayed, Runterm Plus sends the other computer a

CTRL-Q, telling it to resume transmission. All this is done automatically, so you needn't worry about it; I'm telling you this for your information only.

There are two more disk-maintenance commands—Logo-N and Logo-R. Use Logo-N to scratch a disk file. Simply type in the name of the file to be scratched, and Runterm Plus will remove it from the directory. Logo-R is used to rename a file. At the filename prompt, enter the change in this format:

```
newname = oldname
```

and hit the return key.

To see a menu of these local commands and the keys that generate them, hold down the Commodore logo key and Press M when Runterm Plus is running.

BUFFER COMMANDS

The Logo-O combination opens the 29K capture buffer; Logo-C closes it; and Logo-Z resets the pointers to the beginning of the buffer.

To capture characters, you must first open the buffer by pressing Logo-O. When it's full (about 117 blocks), you will see a Buffer Full message mixed in with the incoming characters. Any further data that's transferred will appear on the screen, but will not be captured.

Logo-B allows you to view the buffer contents on screen, and

Logo-P will send the buffer contents to the printer, stripping out screen codes and control codes that your printer can't handle. You can abort either of these operations by pressing CTRL-X. (It may take the program a few seconds to acknowledge your command and stop the operation.) For faster results, hold down both the CTRL key and X until the abort is accomplished.

Two commands allow you to save the buffer contents to disk. Logo-U will save into a disk file all characters in the buffer. The program will ask you to supply the filename and filetype (program or sequential). Logo-S also saves buffer contents, but edits out screen codes and control codes so the saved file can be printed later.

The program handles all characters as Commodore ASCII. If you are calling a standard ASCII database, characters are translated to standard ASCII as they are sent. Incoming characters are converted to Commodore ASCII before they are stored. This means that all text is in a format compatible with your computer, so you don't have to concern yourself with conversions.

FILE AND BUFFER TRANSFERS

There are three ways of transferring files with Runterm Plus.

```

A      View hi-res -- toggle
B      Display buffer to screen
C      Close buffer
D      Display disk directory
G      Play a modem game
H      Change border color
I      Change background color
J      Change text color
K      Redefine DElete key
L      Load a file to buffer
L      Load a file to buf
N      Scratch a file
O      Open the buffer
P      Print buffer contents
Q      Quit the program
R      Rename a disk file
       newname=oldname
S      Save edited buffer
T      Change terminal settings
U      Save buffer unedited
V      Upload buffer

```

Press any key to continue

The first involves loading the file from disk to the program's buffer area and uploading the data from there.

Second, using the XModem protocol, you can transfer files directly to and from your disk. Again, there is no need for conversion; programs transferred by XModem are ready to run when the transfer is done.

The third means of file transfer is via the Bozart transfer protocols.

If you wish to upload via the buffer, use the Logo-L command to load the file into the buffer—it will be loaded over anything that was there before.

To transfer the buffer contents

to the other computer, you have two choices: Logo-V or Logo-Y. Logo-V will send the entire buffer contents without pause.

Logo-Y will ask you to enter a prompt character. The routine will then upload each line of the buffer, pausing after it sends each carriage return. During the pause, Runterm Plus examines incoming characters for the designated prompt character, which signals that the other computer is ready to receive the next line.

This transfer method is specially designed so you can upload messages from your disk to bulletin board systems. Generally, you can compose a message on

your word processor and store it to disk as a CBM ASCII text file. Then, when you're on-line and want to send your message to the bulletin board, load the file using Logo-L and, when the BBS tells you to enter your message, press Logo-Y and supply the prompt character. Runterm Plus will then send the message contents automatically.

To use Logo-Y, you must, of course, know the prompt character being sent by the other computer. In most cases, this prompt character will be the last character in a menu or a start-of-line marker. Before using the Logo-Y command, try to notice which prompt character is being used.

On CompuServe, the prompt is usually a greater-than (>) or a colon (:). On Delphi, the prompt is usually a linefeed (you type CTRL-J when asked to supply the prompt) sent after a carriage return. Other systems simply send the return without a linefeed. If you don't see an obvious prompt character, try the linefeed or carriage return. Better yet, check the service's documentation or ask the SYSOP.

When you initiate either of these uploads, you'll be told that the upload is in process. When the upload is finished, the cursor will return and you'll be told that the transfer is complete. You can

abort these uploads by entering CTRL-X.

SCREEN COMMANDS AND WORD WRAP

Three commands—Logo-H, Logo-I and Logo-J—change the border, background and text colors, respectively. If you are using Commodore ASCII, then you can also use the regular key combinations to change text color, just as you do when the C-64 is in Immediate mode.

Since the C-64 has a 40-column screen, and most telecommunication services use an 80-column format, you'll often find that incoming data will wrap around the screen, leaving the first part of a word dangling on the end of the previous screen line. If this irritates you or makes the words hard to read, then you need Logo-W.

When Runterm Plus boots up, the Logo-W routine is set to eliminate word wrap. The screen is formatted for 40 columns. Broken words are erased from the previous line and moved to the next line for ease of reading.

However, there are times—typing in messages, for example—when you want to use 80-column format to keep up with spacing and the number of characters in a line. To turn off the justification routine, just type Logo-W. Each

time you do this, the program will toggle and tell you its current status.

Some services—Delphi is one—have automatic word wrap. When using such services, you may want to set Runterm Plus to 80-column format (allow word wrap) so that the two justification features will not work against each other.

GRAPHICS AND LOGO COMMANDS

While the C-64's function keys are handy, there aren't enough of them. That's why Runterm Plus uses the logo key command format. As long as you're communicating with another computer that's using standard ASCII, there's no conflict. If, however, you are using Commodore ASCII and want to send the graphics characters represented by the logo and letter key combination, you must use the F8 key to leave Command mode and enter Graphics mode.

The F8 key is a toggle that switches the program between two states. When Runterm Plus is booted, it is in Command mode. This means you can use the logo key to generate local commands. If you are using Commodore ASCII and wish to send graphics characters, then just hit F8 to go into Graphics

mode: hit it again when you need to use commands. Each time you toggle with F8, the program tells you which mode you have selected.

Logo-Q is used to end the program cleanly. If you hit it by mistake, the program allows you to change your mind.

THE OPENING MENU

You can configure Runterm Plus to fit almost any telecommunications format. When you load and run Runterm Plus, you will see an opening screen that gives you six options. You may set the program to call a specific national service, such as CompuServe, Delphi or The Source. If you are calling a BBS, you may choose between standard ASCII or Commodore ASCII. A custom terminal setting is also provided.

Runterm Plus will set up your computer and modem to communicate with the specific service you have selected; you don't have to worry about stop bits, word length or other technical aspects of telecommunications. When using Runterm Plus, be sure to set your modem to the Originate mode.

The selection of any option, except #6, will cause the screen to clear, signifying that Runterm Plus is ready.

If you choose option 6, you'll

be asked to specify the parity, word length, stop bits, duplex and ASCII settings and the deletion character. Once you set these parameters, the screen will clear and you'll be ready to go on-line.

XMODEM TRANSFERS

To transfer a file directly to or from your disk, use the Logo-X command. The sequence for an XModem transfer is as follows.

You select download or upload from the other computer's menus. The other computer will prompt you when it's ready. You then press Logo-X, select X for XModem and type U or D.

If you select U for upload, Runterm Plus will ask you to supply the name of the existing file to be uploaded. You will be told when the transfer is completed, and keyboard control will resume.

If you select D for download, you will be asked for the filename and filetype. (You must not use an existing filename.) The program then creates the new file on the disk. The downloading process then becomes automatic. You need only wait until the program notifies you that the transfer is complete.

If noise on the phone line or some other problem causes the transmission to become garbled, then the garbled portion will be

retransmitted to you. If Runterm Plus and the other computer are unable to complete the transfer, Runterm Plus will abort the transfer and return keyboard control to your computer.

A note about CompuServe: Using the XModem feature of Runterm Plus, you can download all files in CompuServe's Commodore Information Network. The special handshakes required to download programs with the extension .IMG are handled automatically by Runterm Plus.

HIGH-RESOLUTION GRAPHICS SCREENS

Runterm Plus can transfer and display high-resolution graphics screens. To view the screen while you're downloading it, it must be in Doodle format. That is, the file must be constructed so that the high-resolution color area is placed first in the file, followed by a bit-mapped screen.

Although the file will transfer without problems, this hi-res feature will not give you a true copy of a multicolored high-resolution screen, which is longer and in a different format than the regular bit-mapped two-color screen.

There are two ways to use this feature. You can either send a hi-res screen or receive one. Keep in mind, however, that your buffer contents will be overwrit-

ten during this process. Before viewing a screen, save your buffer contents to disk or print them out. If your local bulletin board uses Bozart protocols, you will be able to do graphics downloads from the board, too. Check with your SYSOP.

When Runterm Plus is booted, the hi-res feature is turned off. To activate this feature and view a hi-res screen while downloading, use the Logo-A command before you begin the downloading process. You can toggle on and off the Hi-Res Graphics mode by typing Logo-A.

To transfer a hi-res screen, first type Logo-A, as previously discussed, then, when the other computer is ready, select Logo-X to begin the transfer. Select Bozart protocol and then D for download, and the rest is automatic.

The special Bozart graphics protocol is similar to XModem, but faster, more accurate and more automatic. The protocol switches your computer to Hi-Res mode and back to regular Screen mode. Also, it includes a handshaking feature, which checks to make sure that the incoming file will fit on your disk. (If the file won't fit, Runterm Plus will abort the file transfer and return to Terminal mode.)

In viewing hi-res files, you may experience a few seconds' delay

while the two programs make the initial connection before transferring the file. During this delay, you'll be looking at the bit map of whatever was in the capture buffer. As the transfer progresses, you will see the background color change; then, while the file is being stored to disk, the bit-map screen will appear.

When the transfer is complete, Runterm Plus will pause for three seconds, to allow you to view the hi-res screen, then the screen will clear and you'll be back in Terminal mode.

Keep in mind that the Bozart protocol can only be used to swap files with another terminal program that uses Bozart protocol. Just as you can't speak English and expect to be understood by someone who speaks, say, only French, you can't expect two different transfer protocols to communicate.

Bozart protocol can be used to get an error-free transfer of *any* file, not just graphics files. (The only difference is that you don't use Logo-A to view the nongraphics screen; you go directly to Logo-X for the transfer.)

Although the speed of transfer depends on the file being exchanged and, especially, on the number of repeated characters in the file, as a general rule you can expect to transfer a file 20

percent faster using the Bozart graphics protocols instead of XModem. For example, a typical 9000-byte graphics screen that takes about ten minutes to transfer with XModem will transfer in about seven minutes with Bozart protocol. Faster speeds are not possible, due to the constraints of the 300 bits-per-second speed of the modem.

An interesting side effect of this hi-res capability is that you can get a bit-mapped representation of any nongraphics file being transferred. Use Logo-A before the transfer, just as you would if the file were a graphics screen.

As the file is transferred, you'll see a kind of abstract design being built on your screen. Only actual hi-res screens will show you an organized picture. However, the abstract designs generated by a nongraphics file are more interesting to watch than the constant waiting/transferring messages that normally accompany a file transfer.

CHANGING TERMINAL SETTINGS

There are times when you'll want to change your terminal settings without restarting the program. To redefine your delete key, use Logo-K. The standard

delete character is an ASCII 127. However, some services require you to use a backspace (ASCII 8), and you may need to revert to the Commodore delete character, ASCII 20.

Depending on which service you select from the opening menu, Runterm Plus sets the delete character. For CompuServe and for linkups using Commodore ASCII, the setting is the regular ASCII 20; for Delphi and The Source, the setting is ASCII 127 (true delete).

To reset any or all of your terminal parameters, use Logo-T. This command will review each setting and prompt you to reset it.

MODEM GAMES

The Logo-G command allows you to load and run a special game program while on-line. This feature lets two people with copies of Runterm Plus play a game over their modems during a regular telecommunications session, without breaking the connection or switching programs.

To use this feature, both parties must have a game written specifically for Runterm Plus. Ordinary games will not work, because they load into the same locations in memory where Runterm Plus code resides. ☐

Turtle-Tutor For Tykes

By Peter Crosby

RUN It Right

C-64

Children under six or seven are fascinated by computers, but limited in what they can create with them, since they can't read or handle detail well enough to program. I wrote Elmer the Turtle, an introductory turtle-graphics program, for my own children so they could start to program, and I've found that it can be useful fun for beginning adults, too.

Elmer is a pen-wielding "turtle" who moves about the screen and draws according to a list of instructions you create by selecting options from a menu. The programming is done with only two screens, the second following the first automatically. There's no switching from menu to menu as in more advanced programs like Logo. Eleven simple statements are sufficient to put Elmer

through reasonably complex maneuvers, and four rudimentary editing commands enable you to arrange the program listing.

The program is reasonably crash-proof. If you type in garbage, Elmer just says he doesn't understand and would you please try again. If it does crash, you can usually restart it without losing anything by typing GOTO 700.

The statement list for Elmer is limited to 36 lines so that they'll all fit on one screen. Obviously, after a while you'll want more room. That's when you move on to Logo or regular turtle graphics. Since Elmer's vocabulary and procedures carry over, you'll have a valuable head start.

TALKING TO ELMER

As I mentioned, there can be up to 36 instructions in a list. Each instruction has a line number. After a couple of introductory screens that explain what

| | |
|----------|----------------|
| RIGHT | DELETE |
| UP | CHAR |
| DOWN | COLOR |
| INSERT | TEXT |
| LEFT | BACK TO LINE # |
| PEN UP | STOP |
| PEN DOWN | |

Table 1. Commands for directing Elmer and for editing the instruction list.

| | |
|----------|----------------|
| 0 Black | 8 Orange |
| 1 White | 9 Brown |
| 2 Red | 10 Light Red |
| 3 Cyan | 11 Dark Gray |
| 4 Purple | 12 Medium Gray |
| 5 Green | 13 Light Green |
| 6 Blue | 14 Light Blue |
| 7 Yellow | 15 Light Gray |

Table 2. Color choices.

the program is about, a display appears with an empty list of all the line numbers and, at the bottom, a menu of three choices.

You can go to a line number you specify to write in an instruction; you can type E to make Elmer follow the instructions you've already written; or you can type NEW to clear your list and start afresh. Of course, when you're just beginning to play, only the first choice, writing instructions, is viable.

So, type in the line number you want—probably 1—and hit the return key. The next screen displays all the possible instructions for making Elmer walk and draw, and for editing the list. (See Table 1.)

Choose a direction for Elmer to walk by typing the appropriate word, then tell him how many steps he should take in that di-

rection. You must include the number—he won't understand otherwise. When you press the return key again, your instruction will appear at the specified line number in the list.

To make Elmer draw, tell him PEN DOWN. He'll place his pen on the screen and draw a trail behind him. When you say PEN UP, he'll lift the pen and leave no mark at all. The drawing instructions do not need any following numbers.

Elmer usually uses an asterisk for drawing his trail, because he thinks it looks like a turtle. However, he'll draw with any other character you choose. Just type the instruction CHAR, followed by a space and the character you want. For example, CHAR E makes him leave a trail of Es. You can request any letter, number or punctuation mark on the

| | | | |
|-----|-----------|----------|----------------|
| 1. | PEN DOWN | 19. | |
| 2. | COLOR 0 | 20. | |
| 3. | DOWN 15 | 21. | |
| 4. | COLOR 5 | 22. | |
| 5. | RIGHT 10 | 23. | |
| 6. | UP 14 | 24. | |
| 7. | LEFT 8 | 25. | |
| 8. | DOWN 13 | 26. | |
| 9. | COLOR 1 | 27. | |
| 10. | CHAR X | 28. | |
| 11. | RIGHT 22 | 29. | |
| 12. | COLOR 7 | 30. | |
| 13. | DOWN 5 | 31. | |
| 14. | RIGHT 9 | 32. | |
| 15. | COLOR 14 | 33. | |
| 16. | CHAR . | 34. | |
| 17. | UP 20 | 35. | |
| 18. | | 36. | |
| | RIGHT | LEFT | CHAR |
| | UP | PEN UP | COLOR |
| | DOWN | PEN DOWN | TEXT |
| | INSERT | DELETE | BACK TO LINE # |
| | LINE 18 ? | | |

keyboard except the Commodore graphics symbols.

Elmer can draw in 16 different colors (listed in Table 2). To change color, type COLOR, a space and the number of your choice, 0 to 15. For a monochrome monitor, you can choose from seven shades.

Even though he's just a silly old turtle, Elmer can write messages if you tell him what to say. Type TEXT, a space, then a message from one to ten letters long. If you have a longer message, break it up into a few short ones.

You can make Elmer stop moving anywhere in the list with the instruction STOP. He'll hold

still until you press any key, then continue on. It's a good idea to make STOP the last instruction on a list, so you can see what you and Elmer have created.

CHANGING THE LIST

You can alter Elmer's instructions in various ways after you've written them. As I mentioned earlier, NEW erases the whole list. To erase only one line, specify the line number, then type D for delete. The instruction at that line number will disappear, and all the ones below it will move up.

If you need to change an instruction instead of erasing it, type the new version after des-













| | | | | |
|------------|---|------------|-----|-----------|
| RIGHT |  | 1. RIGHT 4 | E | * * * * |
| LEFT |  | 2. DOWN 3 | | * * |
| UP |  | | | |
| DOWN |  | 1. UP 5 | | 1. UP 5 |
| PEN UP |  | 2. LEFT 2 | 2,I | 2. |
| PEN DOWN |  | 3. STOP | | 3. LEFT 2 |
| CHAR |  | 4. | | 4. STOP |
| TEXT | ABCD* | 1. UP 5 | | 1. UP 5 |
| STOP |  | 2. LEFT 2 | 2,D | 2. STOP |
| | | 3. STOP | | 3. |
| | | 4. | | 4. |
| COLOR 0-15 | | 1. UP 5 | | 1. |
| 0 |  | 2. LEFT 2 | NEW | 2. |
| 6 |  | 3. STOP | | 3. |
| 5 |  | | | |
| 1 |  | | | |

Figure 1. Prompt card of commands for non-readers.

ignating the line number. To add a line between two others, type the number of the second and an I for insert. That line will clear, and its instruction and all those following will move down. Then, by accessing that line number again, you can fill in the blank. By the way, if you should leave blank lines in the list, Elmer won't

mind. He'll just ignore them and plod along.

At any time you're on the second menu screen, you can change the line number you're working at by typing B, for Back to Line #. This recalls the first menu screen, where you can choose another line number.

Although all the commands


are displayed on the screen, I also keep explanatory notes by the computer for the children to refer to. For those who can read, I provide a list of the commands, with a sentence describing the use of each. For non-readers, I've made up a card (see Figure 1) with little descriptive sketches of most of the instructions to Elmer (on the left) and brief examples (on the right) of how the E and editing commands work.

You'll have to explain the E and editing command examples thoroughly at first, but once the child understands, they'll serve as good reminders. In each example, you start with the instructions on the left, then enter the command in the middle to produce the result on the right. I never came up with an illustra-

tion of the Back command. Perhaps you can think of one.

After a while the child won't need the card at all. You'll be amazed at how fast even the very young remember how to spell words they use frequently.

You know, Elmer may not be very smart, but he has endless patience. He never tires of reading your list and walking around the screen following directions. He tells you when he doesn't understand, and gives you as many tries as you need to get it right.

Nowadays, when I'm done using our Commodore, I load in Elmer the Turtle and leave it on. More often than not, some visitor passes by and starts to play. Bingo!—another programmer is born. 

Construction Set

By John Ryan

RUN It Right

C-64; joystick

Construction Set is a programming utility that enables you to "build" colorful, high-speed background graphics on your monitor screen and save them to disk, where they can be used in your Basic and machine language programs.

This menu- and window-driven graphics editor offers joystick control, left and right scroll capability, automatic saves and loads, multicolor support and other options that can be used with the editor alone or with your own programs. It can create and use up to six background screens at once, and is written entirely in machine language for fast and efficient operation.

GETTING STARTED

You can load and run Construction Set from the menu program by pressing the O key. (Or you can load it in Direct mode with the statement LOAD

"CONSTRUCTION SET",8,1. Then type NEW to reset the Basic pointers and SYS 49152 to enter the editor.) A title screen will appear, and with it a window offering three choices:

- 1) ON TO CONSTRUCTION SET
- 2) READ IN ALT CHAR SET
- 3) EXIT TO BASIC

The program can utilize Commodore's built-in character graphics or custom character graphics supplied by you. If you wish to use the Commodore graphics, or if you've constructed screens before, select option 1. You use option 2 to supply the program with a character set of your own. When you select option 2, the program transfers the entire Commodore character set from ROM to RAM, beginning at decimal address 12288 (\$3000 hex). A brief message then appears on the screen advising you to load and run your alternate character Data statements.

It doesn't really matter how you get the redefined character set into memory, just as long as

it begins at 12288 decimal. Any character-editor program that generates Data statements will do the job. One of the easiest methods I've found is to place the screen code of each redefined character at the beginning of its corresponding Data statement. (The character-font editor I use does this automatically, as do many others.)

Let's say, for example, you've redefined the character A to look like a solid block. The resulting Data statements would look something like this:

```
100 DATA 255,255,255,255,255,
      255,255,255
```

To get this redefined A into memory, place its screen code (1) at the beginning of the Data statement:

```
100 DATA 1,255,255,255,255,255,
      255,255,255
```

Using this method for each character, you can easily insert all of your characters into the appropriate memory locations with the following four-liner:

```
10 READ A: IF A1 THEN END
20 FOR I=0 to 7
30 READ B: POKE 12288 + A*8 + I,B
40 NEXT I: GOTO 10
50 REM: <DATA goes here>
```

Whether you use this method or another one, it's important not to hit the run/stop-restore key combination at any time during the character load or run se-

quence, because it'll reset the character pointers installed by Construction Set. Once you've loaded and run the Data statements, type SYS 828 to return to the first screen of the editor.

Exit to Basic, Option 3 on the first menu, is a graceful way of leaving the program without resorting to run/stop-restore.

Option 1 places you in the Editor mode proper, where three choices are available:

- 1) START NEW CONSTRUCTION
- 2) RESUME SAVED WORK
- 3) EXIT TO MAIN MENU

If this is your first time using the program, or if you're otherwise starting new construction, select number 1. You will go right into the editor, where you can begin constructing your graphics ideas.

If you've used Construction Set before, undoubtedly you'll have screen files on disk already. Select option 2 to have BCS load in those files before entering the editor. Loading is automatic and requires no input from you.

CHARACTERS AND COLORS

After you enter the editor, the screen will divide into two sections. The upper half is the actual work area—a graphics screen composed of 16 rows and 40 columns. You'll notice a small flashing cursor in the upper-left corner.

This cursor and the joystick figure prominently in almost every aspect of construction. If your joystick is plugged into port 2, you'll see that you can use it to move the cursor quite easily anywhere within the graphics area.

The bottom half of the screen displays information, including several function key descriptions and, at the very bottom, the command line.

You'll control every aspect of Construction Set from this important line.

Starting at the left end of the command line, notice the character sequence CHAR = . This tells you what character is currently active on the graphics screen. (If you've just booted up, there'll be a space after the equals sign.) Press F5 to select a character to work with.

The cursor will disappear from the graphics screen, and then you can begin searching for a character or symbol by pushing the joystick forward and backward. Graphics characters will scroll by rapidly to the right of the equals sign in CHAR = . When you find a character you want to use, release the pressure on the joystick and press the fire-button. The cursor will reappear on the graphics screen with the newly chosen character under it.

Each character can appear in any of the 16 Commodore colors, blue being the default.

Press F6 to select a different color, and again the cursor will disappear. However, this time pressing the joystick forward and backward will make the character to the right of the equals sign change color. Once you find a color you like, press the fire-button to return to the graphics screen. From then on, that character will always appear in the newly chosen color.

BUILDING A SCREEN

Now that you've selected a character and its color, you can begin experimenting with the graphics screen. Move the joystick to place the cursor at whatever point you choose on the screen; then press the fire-button to place the character you're working with at that point in its appropriate color. You'll notice that the character replaces anything that's already at that location. Once you've finished with that character, press F5 to search for a new character and repeat the process.

In this manner, you can construct a full screen of background graphics in little time. If you're using the built-in Commodore graphics, you'll probably need a while to determine which characters work best for your needs. However, if you're using redefined characters, you'll already have a good idea which ones you want to use.

Should you make a mistake and need to erase something, there are three ways to do it. The most extreme is by pressing F7, which erases everything on the graphics screen. This key is most handy when you're first constructing a background, for each new screen is full of random data.

The second way to erase is to scroll through the entire character set until you find a space character, but this is a time-consuming task. The third way is to press the space bar, which will load a space character (screen code 32) into the working-character register. Since the character you're working with replaces whatever you place it on, a space character effectively erases what's under it. Pressing the space bar doesn't make the cursor disappear, so the effect is immediate.

While you're constructing your screens, you may feel that the cursor is moving too fast to position characters precisely. A fast cursor is great when you want to lay down a lot of identical characters, but clumsy when working with detail. Press F4 to control the speed of the graphics cursor.

The cursor will then disappear from the screen, again cueing you to glance down at the information screen. A prompt will appear below the function-key definitions, asking for an input

between 0 and 6, with 0 being the fastest cursor speed and 6 the slowest. Once you make a selection, the cursor will reappear on the graphics screen.

Sooner or later you'll finish constructing your screen. Then you can decide to save your work and exit or to construct another screen. On the command line, find the word `SCREEN =`, followed by a number ranging from 1 to 6. This is the screen number you're currently working on. Press F2 to select another number.

This time, the graphics cursor will jump from the graphics screen down to the number (or space) following the equals sign. Press the joystick forward to cycle through all the screen choices. When you've found the one you want, press the joystick button and that screen will be displayed on the monitor.

If you've worked on that particular screen before, then all the construction you did will still be there. If it's a new screen, it'll be filled with random data you need to clear by pressing F7.

Anyone who has tried to program games or applications where screen addresses (either color or screen RAM) are used, knows the tedious methods used to calculate screen positions. As an aid to your own programming, Construction Set makes these calculations automatically.

The last two labels on the command line, C.RAM and S.RAM, stand for color RAM and screen RAM, respectively. Each time you move the cursor, these two labels are updated to reflect the precise screen addresses you're working at.

SCROLLING AND COLOR OPTIONS

One of the more powerful options available is a rapid scroll from one screen to another. This background scrolling is fundamental to many arcade games today. Press F3 to enter the Scroll mode. After the cursor disappears,¹ move the joystick either left or right to scroll the entire graphics screen left or right. This mode includes a "wrap" feature so you can scroll continuously. Since this Scroll mode is available to you in your own programs, the scroll function in the Editor mode is just a way to let you preview the effect.

The default colors for the screen and border are black and red, respectively. These can be changed by pressing the S key for screen color or the B key for border color. If you're using multicolor character graphics, you can toggle between standard hi-res and multicolor by pressing the M key. In Multicolor mode, the 1 and 2 keys control color values for background colors 1 and 2.

Construction Set uses colors 0-7 in Hi-Res mode and colors 8-15 in Multicolor mode. Each character appears only in the color you last chose for it.

SAVING SCREENS

There are two ways to save your work using the editor. With the F1 key you can periodically save your screens and character data to disk while you're working on them (handy in case of a sudden power outage). No input is needed from you—the save begins automatically after you press F1. Once the save is complete, you're free to work on your creation again.

If you want to save your work and then exit, press the X key. This save process is exactly the same as if you'd pressed the F1 key, except once the save is complete, the program performs a system cold-start, resetting the computer and leaving you with the familiar Commodore start-up message. This does not erase Construction Set from memory, however, and you can re-enter the program by typing SYS 49152 and hitting the return key.

If, on the other hand, you have a Basic program in memory while using Construction Set, the Save and Exit option will erase it from memory. You can get around this by saving your work with F1, then pressing the run/stop-restore combination when

| Command | Function |
|---------|-----------------------|
| 0 | Load BCS files |
| 1 | Call screen #1 |
| 2 | Call screen #2 |
| 3 | Call screen #3 |
| 4 | Call screen #4 |
| 5 | Call screen #5 |
| 6 | Call screen #6 |
| 7 | Read joystick #2 |
| 8 | Scroll right |
| 9 | Scroll left |
| 10 | Clear graphics screen |

Table 1. Commands for controlling Construction Set from Basic programs.

the save is complete. In any case, make sure the main Construction Set program is on the same disk you're saving to if you're using that disk for constructing screens.

PUTTING IT TOGETHER

Eventually you'll have constructed your backgrounds and will want to incorporate them into your programs. This is where the ease of using Construction Set shines through, as there are no special formulas, language extensions or overlays to remember. You control Construction Set's functions from a Basic pro-

gram by Poking location 820 with a command number, then going to location 52800 with a SYS command. The command numbers and their functions appear in Table 1.

For these commands to work, the first couple of lines in your program must contain the following code (line numbers optional, of course):

```
10 IF X=0 THEN X=1:LOAD
   "CONSTRUCTION SET",8,1: REM
   LOAD IN MAIN CONSTRUCTION
   SET PROGRAM
20 POKE 820,0: SYS 52800: REM
   LOAD IN CONSTRUCTION SET
   FILES FROM DISK
```

These lines load in the main program and your saved screen files. The Construction Set program and the files must be present on the same disk as your own program, or a File Not Found error will occur.

TRYING IT OUT

Now for some examples of using Construction Set from Basic programs. The following one-line program will print all of your backgrounds to the screen:

```
100 FOR T=1 TO 6 POKE 820,T: SYS
   52800:NEXT
```

Use command 7 (Table 1) to read joystick 2, then Peek location 820 to see what the joystick value is. Table 2 lists the five possible values.

-
- 0 Joystick not moved
 - 1 Joystick up
 - 2 Joystick down
 - 3 Joystick right
 - 4 Joystick left

Table 2. Joystick values returned by command 7. (See Table 1.)

You can also Peek location 821 to see the status of the fire-button. If the value in 821 is 0, then the button is not down; if the value is 1, then the button is down. Try running the following program with the fire-button held down:

```

100 POKE 820, 7: REM READ
    JOYSTICK
110 SYS 52800
120 JOY = PEEK(820):REM JOY VAL
130 FIRE = PEEK(821):REM FIRE
    STATUS
140 IF JOY = 1 THEN GOTO 1000:
    REM BRANCH IF JOYSTICK
    GOING UP
150 :
1000 IF FIRE = 0 THEN 100: REM IF
    BUTTON NOT DOWN THEN
    RETURN

```

To use the scroll feature, choose command 8 or 9, depending on the direction you wish to scroll. Each time this feature is accessed, the entire

graphics screen scrolls one column left or right. To move from one screen to another, the scroll feature must be accessed by the SYS command 40 times: to scroll two screens, say from screen 1 to screen 3, it must be accessed 80 times. Try the following short routine, which scrolls left from screen 6 to screen 3:

```

100 POKE 820,9:REM WE ARE
    SCROLLING LEFT
110 FOR J=1 TO 120: SYS
    52800:NEXT :REM SCROLL 120
    COLUMNS = 3 SCREENS

```

Command 10 in Table 1 clears the graphics screen, while leaving untouched the lower portion of the screen. This area, which was the information screen in the Editor mode, is for your own use while in Program mode. It's a good place to keep scoring information, help menus or anything else your program requires.

HINTS AND TIPS

As I mentioned, Construction Set will print a character only in the last color assigned to it. If, after constructing several screens, you find that a certain character color is not working out, *don't* select a different color and redo all the screens by hand. Just choose a different color for the current offending character. Then, the next time a screen is recalled, that character will appear in the last color you chose for it.

Although Construction Set supports Multicolor mode in the editor, within your programs you must switch this in yourself. This can be done by using POKE 53270,PEEK(53270)OR16. Background colors 1 and 2 must also be set, using decimal locations 53282 and 53283, respectively.

If you plan to create several disk files with different backgrounds on each file, then those files must be kept on a separate disk. The reason is that the program always overwrites the old screen-file information with new file information. The names of the files created are .MAPS, .DATA and .COLS.

Whether or not you use a re-defined character set, the pro-

gram transfers the Commodore graphics set to location 12288-14300 decimal. If your Basic program is very large, you may have to protect this area.

For machine language programmers, Construction Set occupies memory from \$C000 to \$CECB, so DOS support can't be used while it is in memory. Also, Basic ROM must not be switched out in favor of a RAM configuration. The data for screen and color is stored underneath the ROM, and Basic is switched in and out as the information there is needed. Construction Set also uses many ROM routines.

Well, good luck with the program; and have fun! ☐

Disk Backup

By C. J. Mohler

RUN It Right

C-64; one or two disk drives

Backup is an easy-to-use program that copies unprotected disks with either one or two disk drives. It's particularly good for copying data disks, because it doesn't override the verification the disk operating system performs to assure correct reads and writes. Most fast-copy programs get some of their extra speed by rewriting the disk operating system to skip this safeguard. As an additional protection for your disks, Backup requires different ID numbers for the disk and its copy, so you can't ruin the copy by mixing them up.

As a machine language program, Backup is slightly faster than a Basic program, and its much larger buffer holds more than a fourth of a disk on each pass. Thus, you can copy a disk in no more than four passes. A chime sounds when it's time to change disks, so you can do

other things while the program is running.

If you have two drives, Backup copies an entire disk without stopping. In fact, I often use it instead of my fast-copy program, because I don't have to sit and change disks while it runs, and I don't have to unplug my printer.

Backup uses Commodore disk commands exclusively. I tested it on 1541 drives, but it should work with any drive that recognizes these commands. Attach one drive to the serial port as device 8. A second drive may have a device number from 9 to 11.

USING THE PROGRAM

If you've been working on the computer, turn it off and on again to clear the memory. Then keep the memory empty, because Backup uses it all and will wipe out anything that's already there. Also, make sure there's no cartridge in the cartridge port, and, as always, don't plug in or unplug a cartridge while the computer is turned on.

Now load Backup by typing in LOAD "BACKUP",8 and run it. To stop the program before it's finished, press the run/stop key or, if the screen is blank, the run/stop-restore combination. If you stop the program before copying is complete, your original disk will be fine, but the copy will be unusable.

Put the disk you want to copy in drive 8 and select the device number of the drive you'll use for the new disk. If you have only one drive, the number will be 8. Press the space bar until the large cursor rests on your choice, then press the return key. If you reach the end of the list without making a selection, press the space bar to send the cursor back to the first choice.

Next you have to choose between active and total backup. Most of the time you'll want active, which copies only those blocks that the disk BAM indicates are being used. The total-backup option copies every block, whether or not it's in use. You'd need this mode for copying a data disk from a program that doesn't mark its file blocks, or if you planned to use the copy to try restoring a scratched file. The program instructions will tell you, as a reminder, that you can't put anything but the file on the disk.

Type in a name, up to 16 characters long, for the new disk. If

you try to enter more than 16 characters, the computer will stop accepting them and you'll have to use the delete key to make room for changes. The special input routine screens out the cursor keys and most of the graphics characters. It also refuses to accept the quote mark, which wreaks havoc in a filename. Press the return key when you're satisfied with the name.

Then, on the same line as the disk name, type in a two-character disk ID that's different from the ID of the disk you're copying. The same characters that are unusable in the disk name are unusable here. When you're happy with the ID, press the return key.

Next, place a blank disk in the drive you picked when you started running the program. Make sure it's a blank one, because if it isn't, all the data on it will be erased. When the disk is ready, press the return key to start formatting. When the job is done, chimes will sound.

Then insert the disk you want to copy in drive 8 and press the return key. The name and ID of the disk will appear on the screen. If this is, indeed, the disk you want to copy, press any key but N. If you got your disks mixed up, press N to try another one. This time you don't have to press the return key after entering your choice.


The program now knows the ID

number of both disks, and it won't let you mix them up. If the ID of the disk you insert doesn't match the one the program is looking for, it'll prompt you again.

Now the screen clears and copying begins. The screen color will change every eight blocks to let you know the program is working properly. If you're using two drives, the screen display won't return until the copying is done.

If you're using one drive, the display will return and chimes sound when it's time to change

disks. Change them and press the return key to continue.

When the entire disk has been copied, you'll be instructed to remove it from the drive. However, first check the screen messages to be sure the program didn't stop because of a disk error. If it did, your copy will be unusable and you'll have to start over. The words BACKUP FINISHED indicate that the copy is good. Finally, run Backup again to copy another disk, or press any key to leave the program. 

Datafile 3.6

By Mike Konshak

RUN It Right

C-64; disk drive; printer

Datafile 3.6 is the newest version of the Datafile database-management system for the C-64—a memory-based system that uses sequential files. With Datafile, you can create your own database, choosing the number and length of fields, as well as their titles. After you've created a record file and entered your data, the program will search, sort, delete, modify and print the records.

Datafile's accessory program DFCALC will mathematically manipulate your file as it produces reports, and its other accessory program, DFPRINT, will create mailing labels and reports according to your customized formats.

As you work with Datafile, keep in mind that it will accept only non-shifted characters when you're inputting data, and that commas, colons, semi-colons and quotation marks are not allowed.

If you're adding a large number of records to a file at one sitting, be sure to save the file often, just in case the power goes out unexpectedly.

DATAFILE INSTRUCTIONS

It's a good idea to experiment with a small database file at first, to get to know the program's capabilities. You can load Datafile into memory from MENU 64, or, from Basic, type LOAD "DATAFILE",8 <return>. When the drive stops, type RUN and press the return key. The screen will display the main menu:

```
CREATE NEW FILE
QUIT PROGRAM
ADD RECORD TO CURRENT FILE
MODIFY RECORD IN CURRENT FILE
DELETE RECORD IN CURRENT FILE
VIEW OR EDIT FILE
SORT RECORDS BY FIELD
PRINT RECORDS USING DFPRINT/
  DFCALC
READ (LOAD) OLD FILE FROM DISK
WRITE (SAVE) CURRENT FILE
  TO DISK
@ DISK DRIVE COMMANDS
$ 4 DIRECTORY
```


Choose the menu options by pressing the key for the first letter of the option. If your program ever crashes or locks up because of a disk drive or printer error, just type GO 68 <return> to get back to the main menu without losing any record data.

CREATING A NEW RECORD FILE

Now you need to create a datafile, so enter C for the create option. When you create a record file, you're defining the structure to which all its records must conform, so evaluate carefully the needs of your application. It's rather difficult to change your mind later, although it's possible by using Datafile utility programs. (See *RUN*, November 1985, for the DFRESTRUCTURE utility.) The rules for creating record file structures are as follows:

1. It's advisable to have no more than 15 fields.
2. As indicated above, field titles cannot contain quotation marks, commas, colons or semicolons.
3. Field length, including the field title, cannot exceed 80 characters.

Now let's create a sample record file for keeping track of club members. The file MEMBERS will have the following structure:

| FIELD | TITLE | LENGTH |
|-------|-------------|--------|
| 1 | LAST NAME | 15 |
| 2 | FIRST NAME | 15 |
| 3 | STREET | 30 |
| 4 | CITY ST | 22 |
| 5 | ZIP | 7 |
| 6 | PHONE | 12 |
| 7 | DATE JOINED | 8 |

Enter this information and then watch for the display that tells you how many records the structure can hold. If the structure is a good size and otherwise satisfactory, press A to accept it. We'll assume this sample structure is all right. You now have a current file in memory.

MODIFYING RECORDS

Now, type about ten records into your file for some data to experiment with. When you press M, for modify, on the main menu, Datafile will ask which records you want to change. If you want to change just one and you know its number, type the number and press return. If you don't know the number or you want to change a number of records, press A to view all the records in the file, one at a time.

The current data in each field in each record will be displayed in turn. If you want to leave the field as is, press return. If you want to erase its data, press the > key. (The > will stay within that record until the file has been saved and reloaded.) You can

also copy a field's data by pressing the = key. At the end of each record, press N to advance to the next record or E to exit.

Unless you know the record number or have changes to make on every record, it's more convenient to use the alternative method of modifying records available through the view-or-edit option on the main menu.

DELETING RECORDS

When you press D for deleting records, you once again have to designate a record or press A to go through the whole file. If you can't remember the record number, go to the view-or-edit option on the main menu and delete the record from there.

Before Datafile will delete a record, it displays the record's entire contents on the screen. If you're sure you want to delete it, press shift/D.

As the deletion occurs, the count of records in the file decreases and all the records after the deleted one are renumbered accordingly. To put all your records back into order, you have to use the sort option in the main menu. Remember to save your revised file to disk.

VIEW OR EDIT FILE

This option offers the most flexibility for viewing, scanning and editing the current file. As each

record is displayed, you'll be given the following eight choices:

| | | | |
|--------|--------|-------|------|
| NEXT | LAST | JUMP | FIND |
| MODIFY | DELETE | PRINT | EXIT |

Next makes the screen step to the next record, Last steps it backward to the previous record and Jump takes it directly to a particular record number, instead of stepping one by one. Print sends the record currently on the screen to your printer.

Find lets you locate records having common data within a certain field. Then you can modify or delete each record. When you're using Find, the screen displays a list of the field names in your current datafile and asks you to enter the number of the one you wish to search. The field name is then displayed and you must enter the common item. Type in the string of text you're looking for and press return.

For example, if you choose a first-name field, you might enter the string JIM. The computer would search out all the records that begin with JIM in the first-name field. Not only would JIM come up, but JIMMY, because it begins with JIM.

SORTING

When you pick the sort option at the main menu, the screen displays the names and numbers of the fields in the file that's in

memory. You can sort the file by up to five fields, all in ascending order, and the sort will take less than 10 seconds.

Datafile stores all data as strings, not as actual numbers. For this reason, the value of each field, when compared for sorting, is determined by the position of each character. Therefore, be sure to be consistent with the format when you're entering field data.

WRITING FILES TO DISK

To save (write) a file to disk, choose W at the main menu. Datafile will ask for the name of the file, and save the file after you respond. The name may be up to 12 characters long.

When you save a record file, Datafile automatically adds the four-character prefix DFJ<space> to the filename. For instance, the name of your sample file will become DFJ MEMBERS. This prefix will show up when you list the directory of the files, but you usually won't need to use the prefix yourself. The accessory programs DFCALC and DFPRINT each add its own prefix to the filenames as well. This arrangement makes it possible for each of the three programs to use the filename you've chosen and still recognize its own version of the file.

Any time a record file is written onto a disk where a file with the same name resides, Datafile

makes the earlier version a backup and assigns it the suffix .BAK. Therefore, when your sample file is saved the second time, the first version will be retained on the disk with the name DFJ MEMBERS.BAK.

You can load the earlier version from the read-old-file option on the main menu. To do so, enter only your filename with the suffix—MEMBERS.BAK here. Don't include the prefix that shows in the directory.

Datafile keeps only one generation of backups, so the third time you save MEMBERS, the first version will disappear. If, for some reason, you want to keep more than one generation of backups, you must give the older ones a different filename.

READING FILES FROM DISK

You'll usually pick the read option from the main menu at the start of a Datafile session. It loads a file you've saved previously. After you've entered R, the program displays all the available files and asks which one you want to load. Type in its name and press return. The file will load and Datafile will return to the main menu. The program will also return to the main menu if you press the return key without typing a filename.

Remember, don't type in the

four-character prefix when you enter a filename to be read; just type the name to the right of the bracket and space.

PRINTING RECORDS

The accessory programs DFPRINT and DFCALC print Datafile 3.6 files you've already saved on disk. They are completely separate and self-contained programs that can be accessed through Datafile or loaded on their own. You access through Datafile by pressing P at Datafile's main menu.

DFPRINT produces reports and mailing labels according to your designs. It reads data directly off the disk, which makes the printing process fast and bypasses any garbage collection delays.

DFCALC prints reports, doing any necessary calculations in the process. It, too, reads the data directly off the disk. See the DFPRINT and DFCALC sections later in this documentation for instructions on how to use the programs.

DISK COMMANDS

Datafile has five disk commands. You access the disk-command menu, which contains these options plus another that returns you to the main menu, by pressing the @ key at the

main menu. The disk-command options are as follows:

FORMAT

This feature allows you to format a blank disk to use for saving files. Insert the disk into the drive, then enter a disk name (up to 16 characters long), a comma, and a 2-character disk I.D. (any combination of numbers and letters)—for example DATAFILE FILES,D2. Follow this sequence with a return. The drive will whirl for about 3½ minutes while it's formatting the disk, then return you to the main menu.

Make sure the disk you place in the drive for formatting is really the one you want to use, because this process will erase the entire disk!

DISK DIRECTORY

To list the directory of the disk currently in the drive, press the 4 key. After you've finished viewing the directory, press any key to return to the disk menu.

SCRATCH A FILE

To scratch any sequential file on the disk, enter the filename, including the DF] <space> prefix, at the prompt and press return. For instance, to scratch your sample file, you'd type DF] MEMBERS. Be sure to type in the name *exactly* as it appears in the

directory, so you don't scratch the wrong file by mistake.

RENAME A FILE

To rename a sequential file, enter the old name exactly as shown in the directory, then the new name when the prompt appears. Be sure to include the special prefixed characters; otherwise, Datafile won't recognize the newly named file and you won't be able to load it from the main menu.

VALIDATE A DISK

This option removes any corrupted files (splat files, with an * beside them in the directory) from your disk.

PRINTER-INTERFACE CONFIGURATION

Because there are so many printers that will attach to the C-64 computer, Datafile can be configured to your special system. Upon accessing DFPRINT or DFCALC, you'll be given a choice of four print configurations. The option display appears as follows:

```
PRINTER-INTERFACE CONFIGURATION
```

```
CURRENT OPTION IS 4
```

```
PRESS 1 CARDCO A, C = 1525  
      2 PRINTERS W/ GRAPHIC  
      INTERFACES
```

```
3 C = 1526, C = MPS801/  
      802/803
```

```
4 PRINT OUTPUT TO  
  SCREEN OR SEND PRINTER  
  COMMANDS  
EXIT TO CONTINUE
```

PRESS THE APPROPRIATE KEY

Options 1-4 are for choosing the configuration. The current option will change each time you press one of these numbers. After selecting one (such as 4, for printing to the screen), you can choose to send commands to your printer or exit this menu into the next part of the program.

DFPRINT has an extra menu not found in DFCALC, so you must turn to the appropriate sections of this documentation for continued instructions.

SENDING PRINTER CODES

Pressing S from the printer-interface configuration menu will deliver you to a routine for sending ASCII codes to set up your printer for various print modes and styles. Most Commodore printers don't offer much flexibility with compressed print and various type fonts.

You'll have to check your printer manual for the codes you can send. Look for statements that print numbers within CHR\$() commands, such as PRINT#4, CHR\$(15), which makes a Commodore printer print all double-

wide or enhanced characters, or PRINT#4,CHR\$(27)CHR\$(66)CHR\$(2), which invokes a compressed print mode (12 characters per inch) on Star printers. Commands that look like PRINT#4,ESC"W"1 must be converted to their ASCII equivalents, such as PRINT#4,CHR\$(27)CHR\$(87)CHR\$(1).

When you're sending printer command codes, enter only the numbers within the parentheses, such as 27, 66 and 2. Enter the numbers one at a time, following each number with a return. You may send one to four numbers, then terminate the sequence with an asterisk (*). Your printer will probably do a linefeed at this point, indicating it has received something.

You can test your change by pressing T at the end of the routine, and you can also choose to send another code by pressing A. E will return you to where you left off in the particular program you're in.

DFPRINT

You can load DFPRINT into memory from MENU 64; or from Datafile's main menu by pressing P, then P again in the ensuing print option menu; or, from Basic, you can type LOAD "DFPRINT",8 <return> and then RUN <return>. The configuration menu I've already de-

scribed will appear. You must choose among options 1-4 before continuing with the program.

DFPRINT will then present these options:

DATAFILE FAST PRINT PROGRAM

OPEN RECORD FILE ON DISK
\$ DISK DIRECTORY 4
QUIT PROGRAM

TRANSFER TO:
D DATAFILE
C DFCALC

PRESS THE APPROPRIATE KEY

DFPRINT assumes that you'll be printing a record file that was created, edited and written (saved) to disk with Datafile and that the record file is located on the disk in the disk drive. It prints the records in the order in which they were last sorted and saved by Datafile. DFPRINT can only read record files, and so cannot make any changes to the records.

Pressing \$ or 4 lets you see the disk directory, and Q quits the program altogether. Pressing O invokes a prompt for you to enter the name of the Datafile to be read. Insert the disk containing your records, type in the name of your record file, MEMBERS, and press the return key. The program will read the structure of the file to get information it will use later. After the light on

the drive goes out, you'll see the following menu:

RECORD PRINTOUT MENU

PRINT OPTION: NO FORMATS
PRESENT
UNFORMATTED LIST
REPORTS FORMAT Formats: RP] *
MAILING LABELS : ML] *

CONFIGURE:
LABELS: 5 ROWS, 34 CHRS,
1ACROSS
PRINTER: OPTION 4, DEVICE# 3
EXIT TO RESTART

PRESS THE APPROPRIATE KEY

This screen lets you choose the type of printout that best suits your needs and reconfigure the printer or change label sizes and type. You can also display the disk directory from this menu.

PRINTING UNFORMATTED LISTS

This is by far the easiest printout option, because you don't have to create a format. Records and fields will be printed in much the same way that you might fill out 3-by-5 cards. An unformatted report resembles the following:

```
[ RECORD # 1 ]-----  
NAME .....MIKE  
AGE .....39  
BIRTHDAY .....05/28
```

```
[ RECORD # 2 ]-----  
NAME .....BECKY  
AGE .....27  
BIRTHDAY .....06/27
```

```
[ RECORD # 3 ]-----  
NAME .....SARAH  
AGE .....11  
BIRTHDAY .....09/10
```

The name of each field is printed along with the field data and record number. The program will automatically skip over the perforations by calculating how many complete records will fit onto one page. The dashed line is printed across the entire page as a guide for cutting the paper in case you wish to attach the record data to a card. This type of printout uses a considerable amount of paper, so it's best to define a report format for printing large files.

If your field lengths exceed 60 characters, you'll have to put your printer into a compressed mode. This prevents the record data from wrapping around to the next line, which would throw off the pagination count.

SELECTING RECORDS

After you've chosen the unformatted-list option or entered a print format into the computer from the disk, the disk-drive light will turn on as the drive positions itself at the start of your record file. Then, using the following screen, select which records to print out:

PRINT OPTIONS MENU

THERE ARE 22 RECORDS IN
MEMBERS

CREATE NEW FILE QUIT PROGRAM
ADD RECORD TO CURRENT FILE
MODIFY RECORD IN CURRENT FILE
DELETE RECORD IN CURRENT FILE
VIEW OR EDIT FILE
SORT RECORDS BY FIELD
PRINT RECORDS USING DFPRINT/DFCALC
READ (LOAD) OLD FILE FROM DISK
WRITE (SAVE) CURRENT FILE TO DISK
@ DISK DRIVE COMMANDS \$! DIRECTORY

PRESS THE APPROPRIATE KEY
THERE ARE 0 RECORDS IN MEMORY

ALL RECORDS IN FILE
FIND RECORDS WITH COMMON
FIELDS
EXIT BACK TO START

PRESS THE APPROPRIATE KEY

Pressing A prints the entire file using the current format. You'll first be asked the starting record number. Record 1 is the default starting position.

Pressing F invokes a routine

that searches for records according to your entered data. After choosing the field to be searched, enter the data to which the records will be compared. If the beginning of the record field matches your data, the record will be printed out. (See the section on viewing records in the Datafile portion of this documentation.)

Pressing the return key during

printing stops the process after a record is completed. You can continue printing or exit the routine, as prompted.

PRINT-FORMAT OPTIONS MENU (DFPRINT ONLY)

Whenever you choose M for mailing labels or R for reports at the record-printout menu, you'll be confronted with the following screen:

PRINT OPTIONS MENU

```
CURRENT RECORD FILE: MEMBERS
CURRENT FORMAT FILE:
LOAD OLD FORMAT
CREATE/CHANGE FORMAT
SAVE CURRENT FORMAT
PRINT RECORDS
DIRECTORY $
EXIT
```

PRESS THE APPROPRIATE KEY

If you're accessing this routine for the first time and no print format is present, then two of the options, print and save, will not be displayed. If you have a format on the disk, then you can load it by first pressing L and then entering the format filename as prompted.

If you've never created a print format for the type of printout you want, or if you desire to change one that's already loaded, you must press C to enter the respective format-definition routines. Cre-

ating formats will be discussed in subsequent sections.

Once a print format is present in memory, pressing P will advance you to the print-options menu, which selects records for printing. This menu was discussed earlier.

MAILING-LABEL FORMATS

Before records from your file can be printed out on labels, you must define a label format specifically for your file. Formats tell the computer which of your record fields to print in each row of the label. Besides the actual data that's to be printed, you must define the size of your label.

The size of a label is specified by the number of rows and the number of characters in each row. Datafile defaults to a standard label size, which measures $1\frac{5}{16}$ of an inch wide and 3 inches long. This popular label size can be printed with 5 rows and 34 characters.

There are various sizes and types of labels available. By types, I mean the number of labels across a page, sometimes referred to as one up, two up, and so forth. Datafile will print up to 30 rows and 136 characters per row on labels up to four across a page (four up). You should be able to design a format around any size or type of

RECORD PRINTOUT MENU

PRINT OPTIONS:NO FORMATS PRESENT

UNFORMATTED LIST

REPORT FORMAT

MAILING LABELS

CONFIGURE:

LABELS: 5 ROWS, 34 CHRS, 1 ACROSS

PRINTER: OPTION SCREEN

DISK DIRECTORY

EXIT TO RESTART

PRESS THE APPROPRIATE KEY

label. I prefer one-up labels, but some printers don't have adjustable tractors, in which case you must print the labels two (or more) up.

After defining the number of rows, you'll be asked which fields you want printed in each row. Up to three record fields may be printed in each row and in any order. You'll be required to enter at least one record-field number for the first field in each

row that will print data. If you want to skip a row (print a blank line), you must enter a zero in all three fields for that row.

When asked for fields 1-3 in each row, enter the record-field number behind the prompt. A zero will be preprinted for you the first time. If you only want one record field in a row, enter the desired record field number for field 1 and zeros for fields 2 and 3.

For example, let's use the address record file called MEMBERS for creating a label format. MEMBERS is a good candidate, since it needs to combine multiple fields in several rows. MEMBERS has the following structure:

| Field | Title | Length |
|-------|-------------|--------|
| 1 | LAST NAME | 15 |
| 2 | FIRST NAME | 15 |
| 3 | STREET ADD | 30 |
| 4 | CITY ST | 22 |
| 5 | ZIP | 7 |
| 6 | PHONE | 12 |
| 7 | DATE JOINED | 8 |

Knowing the record file structure, we'll create a label with five rows. The date the member joined will be in row 1, so he or she will know when it's time to pay dues. We'll skip row 2, the first name and last name will be in row 3, the street address in row 4, and the city, state and zip code in row 5. The rows would look like the following:

| Row | Field1 | Field2 | Field3 | Will print |
|-----|--------|--------|--------|----------------------|
| 1 | 7 | 0 | 0 | DATE JOINED |
| 2 | 0 | 0 | 0 | |
| 3 | 2 | 1 | 0 | FIRST & LAST NAME |
| 4 | 3 | 0 | 0 | STREET ADDRESS |
| 5 | 4 | 5 | 0 | CITY ST & ZIP |

As you can see, it's not important which position the record fields are in, because you can rearrange them when you create

the label format. Again, let me remind you that the default label size is 5 rows, 34 characters long and one up. If you're using a different size or type of label, you must alter the label configuration by pressing L at the print-options menu.

Label formats are handy if you wish to review all your records on the screen. Use printer option 4 and one of the multiple-up options, and change the number of characters to get as many records as possible across the screen.

When you finish defining your label format, you'll have an opportunity to review and change the format. If you press N, you'll be asked if you want to save the label format. You should do so immediately, lest you forget it later. Press Y and you'll be prompted with the current record filename. Either press the return key to accept the filename as is, or make the appropriate changes. Use no more than 12 characters for your label-format filename. A label format with the same name will be scratched and replaced with the new format file. Label format files appear on the disk in the form ML] MEMBERS.

CREATING REPORT FORMATS

Just like labels, a report format must be designed around your

record file if you wish to print the data in an organized and well-presented columnar report. Reports differ from labels in that the record-field data is printed in columns instead of rows. This report format routine is recommended for reports containing mostly text, although the last column may be used for totaling numbers. Use the DFCALC program for reports that require a lot of calculations. In every report, you have the capability for, and must define, the following parameters:

Width in characters. Up to 136 characters can be printed across the page, depending on the capabilities of your printer. You might have to put your printer into a compressed print mode to print more than 80 characters across the page.

Number of title lines. You can have up to four titles printed at the top of the page. The titles will appear only on the first page.

The title for each title line. Titles can be as long as the width of the report. They may not contain commas, colons or semicolons. Examples of titles are: company name, date, report information and sources.

Number of columns. You can have up to 16 columns, at least one of which must be printed.

Header for each column. Headers identify the contents of the column and may be any

name you choose. They are printed above every column at the top of every page. They cannot be wider than their respective columns, nor can they contain commas, colons or semicolons.

Position of each column. You must specify the starting location of each column—a number from 1 to the width of the report. You'll have to determine beforehand how wide you expect each column to be, based on the length of the record-field data that will be printed in the column. In doing your calculations, keep in mind that the columns will be separated by two spaces. You may have to go back and adjust the width of the report and the position of the columns after you see your first printout. A sheet of graph paper comes in handy here.

Contents of each column. Up to three record fields can be printed in each column. The fields within a column will be separated by one space, and the entire contents of the column will be printed left-justified. At least one record field must be chosen for each column and entered in the first field position. As in formatting labels, enter zeros for any unused fields. The record-field titles will be printed on the screen for easy reference. You can print only one record in each row of a report.

Total the last column? The only calculation the report format will perform is summing the contents of the last column. The record-field data defined for the last column must be numeric and must be entered into the first field position. If nonnumeric data (like the dollar sign) occurs at the beginning of the record field, the program will produce a result of zero for that record. The total of the column will be printed at the bottom of the report. The values in the column will be printed right-justified with two decimal places, such as 125.50. To select the totaling option, choose 1 when prompted at the end of the formatting process. Entering 0 turns off totaling. If you need more complex calculations, such as multiplication of record fields and columns within a row, use the DFCALC program.

When you're finished defining your report, you're asked if you want to go back and review or change the format parameters. If you respond with N, you're asked if you want to save the format. If you've made any changes at all to an existing format, or have created a new format, press Y at this time. Use the preprinted filename or type in a new name. Use a maximum of 12 characters in the name of the report-format file. If a file with

the same name already resides on the disk, that file will be scratched, to be replaced by the new one.

Using the file MEMBERS as an example, you might create a report format using the following parameters:

```
REPORT FORMAT FILE: MEMBERS
  (appears on the disk as RP]
  MEMBERS)
NUMBER OF CHARACTERS (wide): 80
NUMBER OF TITLE LINES: 2
TITLE 1: CLUB MEMBER ADDRESS
  LIST
TITLE 2: MAY 31 1985
NUMBER OF COLUMNS: 5
CLM 1: POSITION = 1 HEADER =
  FIRST/LAST NAME FIELDS = 2 1 0
CLM 2: POSITION = 19 HEADER =
  STREET ADDRESS
  FIELDS = 3 0 0
CLM 3: POSITION = 41 HEADER =
  CITY AND STATE FIELDS = 4 5 0
CLM 4: POSITION = 60 HEADER =
  PHONE NUMBER FIELDS = 6 0 0
CLM 5: POSITION = 74 HEADER =
  JOINED FIELDS = 7 0 0
TOTAL LAST CLM: 0
```

Enter the above values after the respective prompts. You'll notice that after you've entered a value, it will be preprinted for you when you go back to review the format parameters.

DFCALC

DFCALC is a companion program to Datafile. You can load it from MENU 64; or by pressing

CALCULATED REPORTS PROGRAM
PRINTER OR INTERFACE CONFIGURATION

CURRENT OPTION IS: 1

PRESS **1** CARDCO A, 1525
2 PRINTERS W/GRAPHIC INTERFACES
3 1526, MPS801/802/803
4 PRINT TO SCREEN

OR **S**END PRINTER COMMANDS

EXIT TO CONTINUE

PRESS THE APPROPRIATE KEY

P, for print option, in the Datafile main menu, then C in the resulting option menu; or from Basic by typing LOAD"DFCALC",8 <return> and RUN <return>.

DFCALC enables you to use your Datafile record-file data to produce spreadsheet-like reports. Since the program provides for considerable flexibility in designing your calculated reports, many applications are possible.

Calculations are performed on only one record at a time within a row of the report. However,

the entire contents of a column can be totaled or averaged. Various methods of justification are provided, as well as many mathematical tokens to expand the capabilities of your printed document.

The following is a more comprehensive description of DFCALC's capabilities:

1) A header consisting of up to four title lines can be printed at the top of the first page of your report.

2) You can define up to 16

columns in your report. The number of columns possible will depend on the width of each column and the number of characters your printer can fit across the page.

- 3) Columns can contain:
 - a) The record number.
 - b) The contents of a field or data within a record.
 - c) The contents of another column within the current row.
 - d) An equation, the results of which will be printed within the column.
 - e) A running total of the data found in a previous column.
- 4) Equations will perform operations using the following operands:
 - a) Numerical content of a field within a record.
 - b) Numerical content of a previous column within the row.
 - c) Numerical constants or values, such as 100, 3.14159(pi), 469.65, 2.6769E - 3, 8.965E10, and so on.
 - 5) Equations can perform operations employing up to three of 47 operators, such as +, -, x, /, %, !, sin(), log() and sqrt(). Included among these are conversion operators that will convert inches to millimeters, degrees to radians, and so on.
 - 6) End-of-column (EOC) operations can be defined as follows:
 - a) No operation (NOOP),

STRUCTURE OF DATAFILE**INVENTORY**

| | | |
|---|---------------------------|-----|
| # | RECORDS POSSIBLE IN FILE: | 253 |
| # | RECORDS IN CURRENT FILE: | 5 |
| # | FIELDS IN EACH RECORD: | 6 |

| # | TITLE OF FIELDS | LENGTH |
|---|-----------------|--------|
| 1 | ITEM | 20 |
| 2 | DESC | 20 |
| 3 | SER# | 20 |
| 4 | COST | 9 |
| 5 | YEAR | 2 |
| 6 | ROOM | 10 |

Figure 1.

- which doesn't print any values at the end of a column.
 - b) Calculate the sum of all the data printed in the column.
 - c) Calculate the average of all the data printed in the column (the total divided by the number of records printed).
 - 7) Column justification can take the following forms:
 - a) Left (for alphanumeric data).
 - b) Right (for alphanumeric data).
 - c) Currency (numeric data, right-justified, two decimal places).
 - d) Decimal (numeric data, right-justified, zero to eight places).
 - e) Percent (multiplies contents of column by 100).

| REC# | ITEM | DESC | SER# | COST | YEAR | ROOM |
|------|-----------|---------------|-----------|--------|------|---------|
| 1 | COMPUTER | COMMODORE C64 | S00011142 | 595.00 | 83 | DEN |
| 2 | TV | SONY | HA778800 | 365.50 | 72 | FAMILY |
| 3 | TAPE DECK | SHARP | LG324M01 | 129.95 | 77 | FAMILY |
| 4 | MICROWAVE | GENERAL ELEC | 444TT6789 | 435.00 | 84 | KITCHEN |
| 5 | CAT | CRUISER | > | > | 81 | |

Figure 2.

DFCALC INSTRUCTIONS

DFCALC will produce accurately calculated reports from your Datafile record files, assuming you design the report format correctly. For the purpose of learning the program, we'll create a fictitious record file as an example.

DECIDING ON A REPORT

Let's assume you've previously created a record file called Inventory with Datafile. It will appear on the disk directory as DF] INVENTORY. The special characters preceding the right bracket identify the sequential file as having been written by Datafile. The Inventory file will have the structure shown in Figure 1. The five records in the sample file appear in Figure 2.

Except for REC#, this report could have been designed and printed with DFREPORT.

Now that you have a file and

know what the structure is, let's determine what to do with the data when you enter DFCALC. The records in Inventory will be used to keep a file of valuables for insurance purposes. Besides the actual records in Inventory, it might be useful to determine the replacement cost of each item. This would require adjusting the original cost by the age of the item and the yearly inflation rate, thus introducing two additional data fields, each containing a formula.

REPORT LAYOUT

Figure 3 suggests the layout of the report, which is 67 printout columns wide. Regarding the fields as "columns," they are numbered 1 through 8. The field descriptions, or column headers, are below the column numbers, and the bottom line, labeled "position," gives the printout column in which each report column begins. There are two spaces between columns. If your record

| COLUMN: | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|-----------|------|------|---------|---------|------|------|-----|---------|
| HEADERS: | REC# | ITEM | DESCRIP | SERIAL# | COST | YEAR | AGE | REPCOST |
| POSITION: | 1 | 5 | 16 | 29 | 43 | 52 | 56 | 61 |

Figure 3.

| COL | CONTENTS | EQUATION | EOC | JUSTIFY |
|-----|----------|--------------------------|-----|---------|
| 1 | # | | N | R |
| 2 | F1 | | N | L |
| 3 | F2 | | N | L |
| 4 | F3 | | N | L |
| 5 | F4 | | T | C |
| 6 | F5 | | N | L |
| 7 | E | 86-F5 | A | R |
| 8 | E | $((F4 * .05) * C7) + F4$ | T | C |

Figure 4.

field data is longer than the width of a column, the data will be truncated, starting at the right.

Once you've positioned the columns, you must determine what contents and operations will be part of each column. Figure 4 shows the structure of each column. I've explained the criteria above.

CONTENTS OF COLUMNS

In the example, INVENTORY, the record's number is printed in the first column, using #. Columns 2 through 6 will contain whatever is found in the respec-

tive fields for that record number, indicated by F1 through F5.

Columns 7 and 8 are defined to have formulas, or equations, denoted by E. Column 7's equation involves subtracting the value found in column 6 (F5, the year of purchase) from the constant 86 (the current year). This results in the age of the item. Column 6 (C6) could be used in place of F5 as notation in the equation, since that column contains the field data. The formula would then be $86 - C6$.

Column 8's equation uses the maximum number of operators

| | NUMBERS | TEXT |
|---------------------------|-------------|----------|
| L — LEFT justify | (123) | (ABC) |
| R — RIGHT justify | (123) | (,ABC) |
| C — CURRENCY, 2 places | (123.00) | (.00) |
| D4 — DECIMALS, 0-8 places | (123.0000) | (.0000) |
| P1 — PERCENT, RES*100 | (12300.0) | (.0) |

Figure 5.

allowed within one column, three, to derive the replacement value. The original cost (F4) is multiplied by .05 (the yearly inflation factor), and this result is multiplied in turn by the age of the item as calculated in C7. Finally, this result is added to the original cost (F4).

Column 8's formula could also have been written $((C5*.05) * C7) + C5$. Since there were no calculations performed in column 5 (C5), the column and the field have the same value.

TOTALS AND AVERAGES

Still referring to Figure 4, the End of Column (EOC) operation must be defined with one of the following three options:

N—*No operation (NOOP)*—do nothing with the column.

T—*Total*—find the sum of the contents of the column.

A—*Average*—find the average of the contents of the column (average = total ÷ number of records).

Since columns 1-4 and 6 are just displaying field data, we choose no end-of-column operations, using N. Column 5 contains the original cost, so we choose T to obtain the total of the original cost of all the items. Column 7 calculates the age of all the items, so it might be interesting to see the *average* age of all our goods by entering A for that column. We choose T for column 8 simply because we'd like to know how much to ask the insurance company for (or at least to realize how much new things cost nowadays).

JUSTIFICATION OF CONTENTS

You must choose how the field data, record number and calculated result will be printed within the column. Your choices are detailed in Figure 5.

Choosing either decimals or percent results in a prompt for

the number of decimal places you want to the right of the decimal point. Any number from 0 through 8 is valid. If you think a calculation might result in answers being printed in scientific notation, choose either left or right justification. If text data is formatted with a numerical format, the value of that column will be zero.

In our Inventory example, we chose left justification for text data, right for numeric and currency for the two columns of costs.

RUNNING DFCALC

Now that we have a game plan (the game gets easier the more we play), let's load DFCALC. As I mentioned earlier, you can load it from Datafile by pressing P at the main menu, and then C at the following menu, or, from Basic, by typing `LOAD"DFCALC",8 <return>`, and then `RUN <return>`.

The first display you'll see relates to configuring your printer, which I discussed earlier. The display looks like this:

CALCULATED REPORTS PROGRAM

USE OLD PREDEFINED FORMAT
OR NEW FORMAT
\$ DISK DIRECTORY
QUIT PROGRAM

TRANSFER TO:
DATAFILE
DFPRINT

PRESS THE APPROPRIATE KEY

Pressing Q terminates the program and \$ displays the directory of the disk currently in the drive. Pressing O brings up the operations menu, in case a calculated report format has already been designed. Pressing N accesses the define report routines for entering all the requirements for printing the sample file Inventory. Press D. When the screen displays the following prompt, enter INVENTORY as shown, then press the return key.

INSERT DISK WITH THE DATAFILE
TO BE USED FOR CALCULATED
REPORTS.

ENTER THE NAME OF THE
DATAFILE:
? INVENTORY

The program will look at the record file named Inventory and retrieve some important file-structure data that will be useful later on. The screen will then display the following sequences as you enter the data as shown:

DEFINE REPORT HEADER FORMAT

ENTER WIDTH OF REPORT (136
MAX)
? 70

ENTER NUMBER OF TITLE LINES IN
THE REPORT HEADER (4 LINES
MAX)
? 2

TITLE #1
? HOUSEHOLD INVENTORY

PRESS CONT REDO START JUMP
EXIT

After you press the return key following the second title, the mini-menu PRESS C R S J E will appear. You'll see this menu often throughout the defining sequence. It lets you change your mind as often as it appears. Here's a description of what each choice does.

C—*Continue* on to the next step. (You're satisfied with what you've done so far.) Actually, pressing any key, including return, will advance you to the next step. You'll find this useful when you're plowing back through the format.

R—*Redo* the last step. This lets you change the very last step or column. DFCALC won't allow you to go backwards more than one step or to the start of a column.

S—*Start over*. Pressing S will send you back to the screen shown above, where you started defining the format. Don't fret! You won't have to retype everything. All your entries will be pre-printed on the screen for you. If you don't want to make a change, just press return to accept the value shown. Keep pressing the return key to walk back through the sequence until you get to where you left off.

J—*Jump* to the end of the def-

inition sequence. Pressing J will send you to the operations menu, bypassing all further entry or updating of the format definition. J assumes you've already been completely through the sequence or have loaded a previously defined format.

E—*Exit* the routine to the beginning of the program. This totally bails you out from the definitions sequence, back to where you first started the program. You might use this key if you discover you're working with the wrong record file or just want to quit.

Now, press return or any other key to continue. You'll get this display:

```
DEFINE REPORT COLUMNAR FORMAT  
  
CALCULATE THE TOTAL LENGTH OF  
ALL FIELDS TO BE INCLUDED IN  
THE REPORT ALLOWING 2 SPACES  
BETWEEN EACH COLUMN.
```

```
ENTER THE NUMBER OF COLUMNS  
IN REPORT. (16 COLUMNS MAX).  
? 8
```

```
FIELD NAME (LENGTH) DATA  
FOR: INVENTORY  
1 ITEM 20          4 COST 9  
2 DESC 20         5 YEAR 2  
3 SER# 20        6 ROOM 10
```

Before you enter 8 for the number of columns, notice the bottom of the screen. The num-

ber, name and length of the fields in the record file Inventory are displayed, so you don't have to remember! This is why you had to enter the name of the record file before starting. From now on, at least until you complete your format definition, the datafile references will stay on the screen.

IMPORTANT NOTES!

1) Don't use your cursor controls during these processes. DFCALC and *all* the Datafile programs use input statements for entering data. Consequently, if you use your up/down cursor, the screen editor won't know where it is. To be safe, use DEL to backspace.

2) Don't use commas, semicolons, colons or quotation marks within any input prompt, or you'll probably lose some of your data or accidentally leave the program.

Pressing the return key brings up the mini-menu (CRSJE), then pressing return again will display a prompt for the position of column 1. Enter, as you are prompted, the individual column positions: 1, 5, 16, 29, 43, 52, 56 and 61. You'll notice the column number (in reversed print) incrementing each time you press return.

The next screen asks for the heading of column 1. Enter the headings as shown on the example, pressing return each time. The screen displays the column currently being worked on, as well as the width of that column. Any header you type in that's longer than the number of characters shown will be truncated. The column headers, you recall, are: 1)#; 2)ITEM; 3)DESCRIPTION; 4)SERIAL #; 5)ORGCOST; 6)YR; 7)AGE and 8)REPCOST.

At this point we'll begin to define the individual columns as to content, justification and end-of-column operations. Continuing with the next screen:

```
FORMAT INPUT ROUTINES FOR  
COLUMN 1
```

```
ENTER CONTENTS OF COLUMN 1
```

```
CHOOSE FIELD DATA  
EQUATION  
RUNNING TOTAL  
# RECORD NUMBER
```

```
? #
```

F will always be preprinted on the input until the fields have been defined. Remember, we want to use the record number for column 1, so press #. You'll then see this screen:

```
SET COLUMN 1 JUSTIFICATION
```


OPTIONS: LEFT
RIGHT
CURRENCY (\$.00)
PERCENT (%)
DECIMALS TO RIGHT

? R

L is the default prompt. Enter R for right justification. Now, we've just completed the definition of column 1. In case you were wondering, the End of Column screen was bypassed, because the column contained only the record number, not any important data. From now on, you'll see the additional menu, and more. The next three screens reveal:

FORMAT INPUT ROUTINES FOR
COLUMN 2

ENTER CONTENTS OF COLUMN 2
CHOOSE F E # Note: simplified menu
? F

WHICH FIELD # WILL BE IN
COLUMN 2
ENTER 1 THROUGH 6
? 1

SET END-OF-COLUMN 2 OPERATION
OPTIONS: TOTAL SUM OF COLUMN
AVERAGE CONTENTS OF
COLUMN
NO OPERATION
TO COLUMN

? N

SET COLUMN2 JUSTIFICATION
OPTIONS: L R C P D
? L

That completes column 2! Complete the same sequence for columns 3, 4, 5 and 6, according to our previous layout, changing the justification and EOC as necessary. Stop when you get to column 7.

Now we'll define the formulas, or equations, for columns 7 and 8. The following display should be on your screen:

FORMAT INPUT ROUTINES FOR
COLUMN 7

ENTER CONTENTS FOR COLUMN 7

CHOOSE F E R #
? E

After you enter E for equation, a new screen will appear with prompts for the equation:

ENTER EQUATION 1 FOR COLUMN 7

RES =

OPERAND X? .
OPERATOR ? .
OPERAND Y? .

Note that you're prompted for equation 1. You can define up to three operations in one column. Enter the operands and operator for the equation we defined earlier for this column: 85 - F5.

Press the return key after each entry shown on the following screen. The completed equation will be shown in the conventional

form, followed by the mini-menu. Press return once more to get past the prompt:

ENTER EQUATION 1 FOR COLUMN 7

RES =

OPERAND X? 85
OPERATOR ? -
OPERAND Y? F5

RES = 85 - F5

PRESS C R S J E

EXPAND EQUATION? Y OR N
? N

Since the calculation in column 7 is sufficient, we won't expand on the equation. Enter N as shown and press return. We finish defining this column by continuing through the End of Column and Justification screens. A reminder: column 7's EOC is A and justification is R.

We are now at the beginning of column 8, which will contain the equation $((F4 * .05) * C7) + F4$. Keep in mind that we must start at the innermost set of parentheses and work out. The entire sequence of screens for column 8 follows. (Yours should grow similarly as you keep entering data and pressing return.)

FORMAT INPUT ROUTINES FOR
COLUMN 8

ENTER CONTENTS OF COLUMN 8

CHOOSE F E R #
? E

ENTER EQUATION 1 FOR COLUMN 8

RES =

OPERAND X? F4
OPERATOR ? *
OPERAND Y? .05

RES = F4 * .05

PRESS C R S J E

EXPAND EQUATION? Y OR N
? Y

ENTER EQUATION 2 FOR COLUMN 8

RES = F4 * .05

OPERAND X? RES
OPERATOR ? *
OPERAND Y? C7

RES = F4 * .05 * C7

PRESS C R S J E

EXPAND EQUATION? Y OR N
? Y

ENTER EQUATION 3 FOR COLUMN 8

RES = (F4 * .05) * C7

OPERAND X? RES
OPERATOR ? +
OPERAND Y? F4

RES = ((F4 * .05) * C7) + F4

PRESS C R S J E

SET END-OF-COLUMN 8 OPERATION

OPTIONS: T A N
? T

SET COLUMN 8 JUSTIFICATION

OPTIONS: L R C P D
? C

PRESS C R S J E

When you've finished defining column 8, you've completed the entire calculated-report format.

OPERATIONS MENU

The last return you pressed presented this screen:

CALCULATED REPORTS
OPERATIONS MENU

CURRENT DATAFILE: INVENTORY

SAVE REPORT FORMAT
LOAD REPORT FORMAT
CHANGE REPORT FORMAT
PRINT RECORDS
RECONFIGURE PRINTER
\$ DISK DIRECTORY
EXIT THIS DATAFILE

PRESS THE APPROPRIATE KEY

Pressing \$ displays the directory of the current disk in the disk drive, E exits and closes the file you've been working with and

sends you back to the beginning, and C sends you to the start of the format-definition routines so you can change one or all of your entries.

You can jump back to this menu, once you've made your corrections, by pressing J at one of the many mini-menu prompts. R sends you to the printer or interface configuration menu, where you can send printer CHR\$ commands, and S saves the current calculated-report format to the disk.

Warning! Any calculated-report format file (shown on the disk as CRJ INV...) of the same name, will be scratched and replaced with the current format. If you've made changes to a format you want to keep, and also wish to retain the original, give the revised format a new name.

Make sure you save your current format before using E or L. However, you'll be given a second chance if you happen to forget.

L loads a format file from the disk into the computer's memory. This overwrites the format currently held in memory.

THE FINAL PRODUCT

Let's assume you pressed A, for all records, at the print-options menu. The computer will print the title and heading of the

report. Then it will start at the beginning of the sequential file and commence reading the records. After every record, the equations for each column will be calculated and the results printed in the current row on the report. Then the next record will

be read, and so forth. When all the records have been read, the end-of-column information will be processed and printed. Finally, at the bottom of the report, the name of the record file will be printed, along with the number of records reported. ®

Operands are variables (data) being operated on by operators, such as *, /, +, -, sqr and int. Operands are denoted in the List of Legal Operators (see Table 2) as X and Y. The value created by a calculation is called the resultant. To help clarify and define these terms, I'll use the following simple equations as examples:

A=B*C Z=X+Y RES=.SINX RES=X*Y RES=RES/200

In these equations, anything to the *left* of the equals sign is the *resultant* of the equation. The variables A, Z and RES are, therefore, resultants. In all the equations that you'll define using DFCALC, RES will be your resultant.

The items to the *right* of the equals sign will always consist of two operands and one operator, the middle term being the operator. B, C, X, Y, RES, 200 and the period are all operands. *, +, SIN, / and / are all operators. The following is an expanded and more detailed description of what you can use as legal operands:

Record fields. An equation using record fields (F1, F4, F12, etc.) can take any of the following forms: RES=

F2*F3; F41F2; F413; 100*F5; C3+F3; .ABSF1; .SINF4; RES*F2; F4+F5

The following conditions apply:

- 1) The number of the field defined must exist in the record file being printed.
- 2) The same field may be used more than once in any one column or in a number of column equations in either operand position.
- 3) Fields may be used in an equation with any operator except the sum through (+ +) operator.
- 4) The field referenced must contain numeric data. Alpha (non-numeric) characters such as A through Z, \$, %, etc., in the data will produce a result of 0, an unreliable result, or an error condition when used in a calculation. If the equation cannot be resolved, "??error??" will be printed as a result or the program may crash.

Columns. An equation using columns may take the same kinds of forms as those involving fields, subject to the following condition: The column number referenced must be less than the number of the column containing the equation. If, for example, you're defining an equation for column C6, you can't use C6, C7, C15, etc., as operands.

Constants. Constants used in an equation may occupy either operand position and may have a positive or negative value, but must be within a range operable for the computer. Pi (3.14159265...) is the only predefined constant.

Other operands. The period is used as the first operand when there is only one operand being operated on. This is usually the case when a higher level operator (identified by a three-letter code) is being used. For example: .SIN 30; .INV F1; .DEG 45; .SQR 4; and so on.

RES. The resultant (RES) of a previous calculation within the current column must be used as an operand if more than one operator is used. Up to three operators or equations may be used to complete a calculation within a column. The first operator may use any two legal operands, but the second and third operators *must* have RES as one of the operands in either position.

Table 1. Rules for Legal Operands

| Op | Func | Description of Operation |
|-----|---------|--|
| ++ | CX + CY | SUM the columns X through Y. |
| + | X + Y | ADD the values X and Y. |
| - | X - Y | SUBTRACT the value Y from X. |
| * | X * Y | MULTIPLY the values X and Y. |
| / | X / Y | DIVIDE the value X by Y. |
| ↑ | X↑Y | Raise the value X to the POWER indicated by the EXPONENT Y. Y may be positive, negative or fractional. |
| SGN | .SGNX | Changes the SIGN of X from + to - or - to +. |
| SQR | .SQRX | Returns the SQUARE ROOT of X. |
| INV | .IN VX | Returns the RECIPROCAL of X (1/X). |
| ABS | .ABSX | Returns the ABSOLUTE VALUE of X. |
| INT | .INTX | Makes a fractional X into an INTEGER. |
| LOG | .LOGX | Returns the NATURAL (base e) LOG of X. To convert to log base 10, divide by .log10. |
| EXP | .EXPX | Returns the value of the mathematical constant e (2.71828...) raised to the power of X. |
| SIN | .SINX | Returns the SINE of X, where X is measured in RADIANS. |
| COS | .COSX | Returns the COSINE of X. |
| TAN | .TANX | Returns the TANGENT of X. |
| SEC | .SECX | Returns the SECANT (1/COS) of X. |
| CSC | .CSCX | Returns the COSECANT (1/SIN) of X. |
| COT | .COTX | Returns the COTANGENT (1/TAN) of X. |
| ATN | .ATNX | Returns the ARCTANGENT (inverse tangent), or angle, in radians, whose tangent is X. |
| ACO | .ACOX | Returns the ARCCOSINE (inverse cosine), or angle, in radians, whose cosine is X. |
| ASN | .ASN X | Returns the ARCSINE (inverse sine), or angle, in radians, whose sine is X. |
| ASC | .ASCX | Returns the ARCSECANT (inverse secant) of X. |
| ACS | .ACSX | Returns the ARCCOSECANT (inverse cosecant) of X. |
| ACT | .ACTX | Returns the ARCCOTANGENT (inverse cotangent) of X. |

Table 2. List of Legal Operators

| | | |
|-----|-------|---|
| RAD | .RADX | Converts the angle X from degree measure to RADIAN measure. |
| DEG | .DEGX | Converts the angle X from radian to DEGREE measure. |
| MMS | .MMSX | Converts the distance X from inches to MILLIMETERS. |
| INS | .INSX | Converts the distance X from millimeters to INCHES. |
| KMS | .KMSX | Converts the distance X from miles to KILOMETERS. |
| MLS | .MLSX | Converts the distance X from kilometers to MILES. |
| DFR | .DFRX | Converts the temperature X from degrees centigrade (Celsius) to degrees FAHRENHEIT. |
| DCG | .DCGX | Converts the temperature X from degrees Fahrenheit to degrees CENTIGRADE (Celsius). |
| GMS | .GMSX | Converts the weight X from ounces to GRAMS. |
| OZS | .OZSX | Converts the weight X from grams to OUNCES. |
| KGM | .KGMX | Converts the mass X from pounds to KILOGRAMS. |
| LBM | .LBMX | Converts the mass X from kilograms to POUNDS. |
| NWF | .NWFx | Converts the force X, in foot-pounds, to NEWTONS. |
| LBF | .LBFx | Converts the force, in newtons, to FOOT-POUNDS. |
| MPS | .MPSX | Converts the velocity X from feet per second to METERS PER SECOND. |
| FPS | .FPSX | Converts the velocity X from meters per second to FEET PER SECOND. |
| NSM | .NSMX | Converts the pressure X from pounds per square inch to NEWTONS PER SQUARE METER. |
| PSI | .PSIX | Converts the pressure X from newtons per square meter to POUNDS PER SQUARE INCH. |
| = | X=Y | If X equals Y then True (res=1), else False (res=0). |
| < | X<Y | If X is less than Y then True. |
| > | X>Y | If X is greater than Y then True. |
| <> | X<>Y | If X is not equal to Y then True. |
| or | XorY | If either X or Y is True (greater than 0) then True. |
| and | XandY | If both X and Y are True (greater than 0) then True. |

When DFCALC is performing calculations, it will try to catch as many mathematical errors as possible. Some errors caused by calculations may force the computer to crash, while others may just produce inaccurate results. Syntactical errors (which cause the premature termination of the program) have, to the best of my knowledge, been accounted for.

The following produce an error condition, resulting in a value of 0 for the particular row/column cell, or an `??error??` flag.

- 1) Dividing by zero: $X/0$.
- 2) Taking the square root (or any even root) of a negative number: $\text{sqr}(-X)$.
- 3) Taking the CSC or COT of zero: $\text{cot}(0)$.
- 4) Taking the natural log of a number less than or equal to zero.
- 5) Raising the mathematical constant e to a power greater than 87: $\text{exp}(87)$.
- 6) Using an invalid or illegal operator or operand.
- 7) Using an operator in an equation where an operand should be (and vice versa).
- 8) Using a numerical constant outside the range of the computer.
 - a) Integers (whole numbers without decimal places) must be within the range of -32768 to $+32767$.
 - b) Floating-point numbers, in scientific notation, must be within the range of $+2.93873588e-39$ to $+1.70141183e+38$.

Note: A calculation that tries to produce a result outside the legal range of the computer will result in an `?OVERFLOW ERROR` and will definitely crash the program. There's no simple way to test for this, so stay away from galactic calculations. Since DFCALC uses a buffer for each column to keep track of column totals and values for calculations within a row, it's possible that your field data may inadvertently cause an `?OVERFLOW ERROR`.

To help prevent what is intended to be printed as text from being interpreted as a number, you should precede suspect record-field items with an alpha character, such as # or P. This will always return a value of 0, since numbers after alpha characters are ignored.

Table 3. Causes of Error

Please send me back issues of ReRUN

- Gamepak Cassette version(s) at \$11.47*
 Summer Edition Disk version(s) at \$21.47
 Fall Edition
 Productivity Pak (Disk only)
 Winter Edition
 Jan./Feb. 1986 (Disk only)**
 March/April 1986 (Disk only)**
 May/June 1986 (Disk only)**
 July/August 1986 (Disk only)**

** Prices include postage and handling. Foreign Air Mail please add U.S. \$1.50 per item and \$25 per subscription. Prepayment only.*

*** Includes 128 mode programs.*

Payment Enclosed MC VISA AE

Card # Exp. Date

Name

Address

City State Zip

Signature

ReRUN • Elm Street • Peterborough, NH • 03458

BEAT THE RUSH!

Please send me:

- 1 year (6 issues) for \$89.97
 September/October ReRUN disk for \$21.47.*

**Available in October.*

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

Price includes postage & handling. Foreign Air Mail please add U.S. \$1.50 per item and \$25 per subscription. Prepayment only.

Payment Enclosed MC VISA AE

Card # Exp. Date

Name

Address

City State Zip

Signature

ReRUN • Elm Street • Peterborough, NH • 03458

Please send me back issues of ReRUN

- Gamepak Cassette version(s) at \$11.47*
 Summer Edition Disk version(s) at \$21.47
 Fall Edition
 Productivity Pak (Disk only)
 Winter Edition
 Jan./Feb. 1986 (Disk only)**
 March/April 1986 (Disk only)**
 May/June 1986 (Disk only)**
 July/August 1986 (Disk only)**

** Prices include postage and handling. Foreign Air Mail please add U.S. \$1.50 per item and \$25 per subscription. Prepayment only.*

*** Includes 128 mode programs.*

Payment Enclosed MC VISA AE

Card # _____ Exp. Date _____

Name _____

Address _____

City _____ State _____ Zip _____

Signature _____

ReRUN • Elm Street • Peterborough, NH • 03458

BEAT THE RUSH!

Please send me:

- 1 year (6 issues) for \$89.97
 September/October ReRUN disk for \$21.47.*

**Available in October.*

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

Price includes postage & handling. Foreign Air Mail please add U.S. \$1.50 per item and \$25 per subscription. Prepayment only.

Payment Enclosed MC VISA AE

Card # _____ Exp. Date _____

Name _____

Address _____

City _____ State _____ Zip _____

Signature _____

ReRUN • Elm Street • Peterborough, NH • 03458

Productivity for the Whole Family!

*Word Processing - Database Management
Telecommunications - Spreadsheet
Manipulation - Education - Graphics
Disk Utilities - Desktop Accessory*

For the G-64:

- ▶ Menu 64
- ▶ Runscript 2.40*
- ▶ 1650 Autodial*
- ▶ 1670 Autodial*
- ▶ Runterm Plus
- ▶ Logo for Kids
- ▶ Construction Set*
- ▶ Disk Backup*
- ▶ Datafile*
- ▶ DFPrint*
- ▶ DFCalc*

For the G-128:

- ▶ Menu 128
- ▶ Runscript 2.40*
- ▶ Reminder 128*
- ▶ Gradebook 128

***New program releases created for this ReRUN disk!**

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 • Elm Street • Peterborough, NH 03458

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

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

The entire contents are copyrighted 1988 by CW Communications/Peterborough. Unauthorized duplication is a violation of applicable laws.

©Copyright 1988 CW Communications/Peterborough