UNPROTECTED

Backups may be made using standard copying procedures.

COMPATIBLE

Any version of Apple II® ProDOS™ *and* DOS 3.3 40 and 80 columns

· GPLE ·

GLOBAL PROGRAM LINE EDITOR
by NEIL KONZEN

GPLE lets you edit Applesoft program lines *fast* without awkward "escape editing" and slow cursor tracing.

INSERT & DELETE
GPLE works like a word processor for Applesoft. Program lines may be edited by jumping the cursor to the change point and inserting or deleting text. No need to trace to the end of the line before hitting Return.

GLOBAL SEARCH & REPLACE
Quickly search for any word or variable in your programs. For example, find all lines containing a GOSUB, or edit or delete all REM statements, or all occurrences of a particular variable. *Replace any variable*, word or character with any other. For example, change all X variables to ABC's, or change all occurrences of "Hello" to "Good Bye".

80-COLUMN COMPATIBILITY
All GPLE features support the Apple IIc and Apple IIe 80-column cards, as well as most others.

DEFINABLE FUNCTION KEYS
GPLE lets you define ESC plus any keystroke to perform any task. For example, *ESC-1* can Catalog Drive 1, *ESC-L* can execute a "HOME: LIST", *ESC-N* can type your name... *anything* you want. These functions may be defined, edited and saved quickly without running a separate program.

GPLE DOS MOVER
Gain an *extra 10K* of program space by moving DOS 3.3 to auxiliary memory.

PLUS APPLE TIP BOOK #7
Learn more about using *your* Apple! Beagle Bros' Apple Tip Book #7 (inside this manual) contains valuable Apple information and educational experiments, including some all new GPLE tips and tricks. Hours of educational, entertaining reading.

FREE PEEKS & POKES CHART
Apple's "Peeks, Pokes, Pointers and Calls" on one 11×17 poster. An indispensable Apple programming tool.

GPLE Command Summary

control-A toggles upper/lower case (page 23).
control-C cancels a listing (page 22).
control-E . . . enters Edit Mode (page 7).
control-O . . . selects GPLE Options (page 26).
control-P pages a listing (page 22).
control-S pauses a listing (page 22).
control-V toggles 40/80 columns (page 24)
control-W edits any text being typed (page 7).
ESC followed by a character calls an Escape function (page 15).
Delete (Apple IIe) acts as a backspace key (page 14).

GPLE Edit Commands

ENTERING EDIT MODE (page 7)
control-E line-number edits a program line.
control-E . (period) edits the last line edited.
control-W edits any text being typed.

MOVING THE CURSOR (page 8)
Left or Right Arrow moves cursor and cancels Insert Mode.
control-B jumps cursor to beginning of text being edited.
control-F character finds the character.
control-N jumps cursor to end of text being edited.

CHANGE COMMANDS (page 9)
control-C converts from upper to lower case and vice versa.
control-D deletes the character at the cursor.
control-I enters Insert Mode.
control-O allows one control-character to be inserted.
control-P packs text, removing spaces between quotes.
control-R restores the line being edited.
control-Z character deletes text from cursor to the character.

EXITING EDIT MODE (page 8)
Return (control-M) exits Edit Mode and accepts edited line.
control-Q exits Edit Mode, deleting text from cursor to end.
control-X exits Edit Mode, without accepting any changes.

GLOBAL EDITING
See instructions and examples on pages 11-13.

GPLE Escape Functions

See pages 15-21 for details.
The standard GPLE Escape Table is on pages 16-17.

IMPORTANT! BEFORE YOU USE GPLE:

DO NOT BOOT THE GPLE DISK if you want DOS 3.3 GPLE. Boot it only if you want *ProDOS* GPLE. More on page 2.

APPLE IIc USERS must use the CONFIG GPLE program (page 26) and select the "APPLE IIC.VID" video driver when asked for DRIVER FILE NAME.

GPLE and Apple Tip Book #7 (back section of this book)
ISBN 0-917085-05-1
Published by **BEAGLE BROS MICRO SOFTWARE, INC.**
3990 Old Town Avenue, San Diego, California 92110

Table of Contents

GPLE: An Overview

GPLE is a highly-sophisticated Apple II, II+ and IIe machine-language utility designed to simplify Applesoft program editing, reduce program development effort and save you time.

BASIC PROGRAM LINE EDITING

GPLE supports Integer BASIC editing under DOS 3.3 only.

GPLE lets you quickly edit Applesoft program lines without awkward insert techniques and slow cursor-tracing. In addition, GPLE's Global capabilities let you quickly search through and edit an entire program or section of a program at once.

ESCAPE FUNCTIONS

GPLE also allows you to program your ESC (Escape) key to perform almost any task— ESC plus the next key pressed can automatically type and execute any command or series of commands. For example, ESC-L (two keystrokes— **ESC** followed by **L**) can be programmed to type "TEXT: HOME: LIST" or "POKE-16304,0" or absolutely anything you want. Use GPLE's standard Escape Table, change it, or build your own.

GPLE IS TRANSPARENT.

This means that GPLE is always available when you need it, but not in the way when you don't, even though it stays in your Apple's memory. GPLE is unaffected by even the most "destructive" commands, such as NEW, INT and FP.

GPLE IS BACKUP-ABLE.

Unlike most commercial software, GPLE is not copy protected (just copy*right*ed). Your original disk may be backed up, and programs may be transferred from disk to disk using normal procedures—see your DOS 3.3 or ProDOS Manual.

Beagle Bros does not copy-protect any of its software, making it easier to use and much friendlier to work with. Please support our unlocked policy by not giving our programs away. You support us, and we'll support you.

ProDOS and DOS 3.3

IMPORTANT

This disk is formatted for *both* ProDOS and DOS 3.3. To access ProDOS GPLE, boot the GPLE disk or any normal ProDOS disk. **To access DOS 3.3 GPLE, DON'T boot the GPLE disk.** Instead, boot any normal DOS 3.3 disk (like the *System Master* Disk that came with your Apple).

The GPLE Catalogs

The GPLE disk has two catalogs. You will be able to see and access only the files that correspond to your current DOS. (see **ProDOS and DOS 3.3**, page 2).

PRODOS CATALOG ONLY:

STARTUP is the greeting program that runs when you boot the GPLE disk. STARTUP Bruns GPLE (page 5).

GPLE is the ProDOS version of GPLE (page 5).

PRODOS AND DOS 3.3 CATALOGS:

CONFIG.GPLE (ProDOS) or **CONFIG GPLE** (DOS 3.3) lets you choose GPLE default characteristics and 80-column parameters (page 26).

ESCAPE.SAVE (ProDOS) or **ESCAPE SAVE** (DOS 3.3) lets you store a customized Escape table on disk (page 21).

ESCAPE.PRINTER (ProDOS) or **ESCAPE PRINTER** (DOS 3.3) prints a list of the current Escape table (page 21).

REMOVE.GPLE (ProDOS) or **REMOVE GPLE** (DOS 3.3) removes GPLE from memory (page 28).

ULTRATERM.VID and other "**.VID**" files (if any) make GPLE work with certain 80-column cards (page 24).

NOTES covers program updates and other files and material not mentioned in this manual. RUN NOTES to learn more.

DOS 3.3 CATALOG ONLY:

HELLO loads the appropriate version of GPLE (page 4).

On older versions of DOS 3.3 GPLE, these files were called **PLE**.48, **PLE**.LC, **PLE**.DM and **PLE** DOS MOVER.

GPLE.48 is the 48K version of GPLE (page 4).

GPLE.LC is the version of GPLE that locates itself, usually with a second language, in the Language Card of an Apple II or II+, or in the upper 16K of a standard IIe (page 4).

GPLE.DM is the version of GPLE that resides on the Language Card or auxiliary memory with DOS (page 4).

GPLE DOS MOVER puts DOS on the Language Card, and runs HELLO, which loads GPLE.DM (page 4).

FIX FID, FIX MUFFIN and **FIX RENUMBER** make *System Master* programs compatible with GPLE.DM (page 28).

Loading GPLE: DOS 3.3

One way to load DOS 3.3 GPLE is to boot a normal DOS 3.3 disk (for example, your *3.3 System Master*), then insert the GPLE disk and **RUN HELLO**. HELLO determines which DOS 3.3 version of GPLE (see below) is appropriate for your Apple and loads it for you. HELLO and any of the files on the GPLE disk may be transferred to your disks.

Another way to load DOS 3.3 GPLE is to boot a normal 3.3 disk, and BRUN the GPLE version that you want, either with a direct keyboard command (for example, **BRUN GPLE.48**) or with the last command in a program (usually your greeting program). For example:

If you don't understand all of this, just RUN HELLO and skip to page 5.

```
PRINT CHR$(4);"BRUN GPLE.48"   or...
PRINT CHR$(4);"BRUN GPLE.LC"   or...
PRINT CHR$(4);"BRUN GPLE.DM"
```

For "BRUN GPLE.DM" to work, DOS must *already* have been put on the Language Card *by GPLE DOS MOVER.* To put it it there, **BRUN GPLE DOS MOVER** (see page 27).

THE THREE DOS 3.3 VERSIONS OF GPLE

(DOES NOT APPLY TO PRODOS)

In case you're new around here, "DOS" is the Disk Operating System, the machine-language program which runs your disk drives and so on.

Putting DOS and/or GPLE on the Language Card is often a wise move, because it frees up programming space, leaving room for larger programs.

ProDOS will not let you move DOS.

There are three versions of DOS 3.3 GPLE. All perform identically; they differ only in where they locate themselves in memory and where they expect to find DOS. See the inside back cover for GPLE memory maps.

GPLE.48 ("48K") This is the GPLE version to use on 48K Apples with no extra memory or Language Card. GPLE.48 resides in normal memory between DOS and the DOS buffers, costing you 4K (4096 bytes) of memory.

GPLE.LC ("Language Card") This version locates itself in the Language Card (or "RAM Card") of an Apple II or II+, or in the upper 16K of a normal 64K Apple IIe. Using GPLE.LC instead of GPLE.48 gives you 4K of programmable memory. GPLE.LC often shares the Language Card with Applesoft, Integer Basic, or Beagle Basic.

GPLE.DM ("DOS Mover") This version resides on the Language Card with DOS. DOS must be put there *first* with GPLE DOS MOVER.

Loading GPLE: ProDOS

To load ProDOS GPLE, simply boot the GPLE disk. This Runs STARTUP, which Bruns GPLE. OR you may boot any normal ProDOS disk and **BRUN GPLE**. This can be a direct keyboard command, or the last command in an Applesoft program. For example:

```
PRINT CHR$(4);"BRUN GPLE"
```

Using GPLE

ProDOS GPLE and all three versions of DOS 3.3 GPLE perform identically. Once GPLE is loaded into memory, you can practically forget about it until you need it—Load and Save programs, etc., as you normally would.

To use GPLE, you must type one of the "immediate" (not in a program) commands shown below. You will usually need to type additional information to complete the command.

GPLE intercepts every keystroke before sending it through the normal channels in your Apple. Most of the commands that you type will behave normally and not be affected by GPLE; you will find few surprises. There are some differences, however. For example, typing control-E usually does nothing special. But with GPLE installed, control-E prints the word "EDIT" on the screen, and puts you in GPLE's Edit Mode.

GPLE IMMEDIATE-MODE COMMANDS

ESC plus the next key pressed is used for GPLE's Escape Functions (page 15).

Control-A toggles between upper and lower case for older Apples with upper-case-only keyboards (see page 23).

Control-E enters Edit Mode (page 7).

Control-O selects GPLE Options (page 26).

Control-V toggles between 40 and 80 columns (page 24).

Control-W puts the text being typed in Edit Mode (page 7).

Delete (Apple IIe only) acts as an alternate, easier to reach, backspace key (Left Arrow equivalent).

Program Line Editing

If you haven't used Integer Basic, you probably never will. GPLE is compatible with it anyway.

GPLE lets you edit Applesoft and Integer Basic program lines with ease. Text may be inserted and deleted from lines, much as you would do on a word processor. The cursor may be moved instantly to any part of a line. There is no more need to "trace over" program lines or do time-consuming "cursor editing".

FIRST, AN EXAMPLE:

If you are unfamiliar with GPLE's line editing functions, load GPLE (see on page 4) and type this program line:

10 PRINT "THIS IS A TEST."

Now let's use GPLE to make a change:

1. Type **control-E** (hold down the CONTROL or CTRL key while you type an E). The word "EDIT" will appear on the screen with a blinking cursor.

2. Type the number of the line you want to edit (**10** in this case) and press **Return**. Program Line 10 will appear on the screen with a blinking cursor after the 10.

3. Use the Right Arrow key to move the cursor on top of the "T" in "THIS".

4. Type **control-I** (that's "**I**" as in "**I**nsert"). You are now in "Insert Mode" even though nothing visible has occured.

5. Type "**WATCH** ". Notice how the line is changed.

6. Use the Right-Arrow key to move the cursor to the "I" in "IS".

7. Type **control-D** five times to delete five characters.

8. You are finished, so hit **Return**. No need to trace to the end of the line.

LIST line 10 for inspection. It should read:

10 PRINT "WATCH THIS TEST."

You are now a semi-expert at using GPLE! To become a full-fledged whiz, load a program and play around with all of the GPLE editing commands that follow.

Entering Edit Mode

control-E (EDIT)

To edit a program line, type **control-E**, the number of the line you want to edit, and press Return. Edit the line. Normally, you press Return to exit the Edit Mode (see page 8). It doesn't matter where the cursor is when you press Return.

A few Rules: (1) Control-E must be the first character typed on a line. (2) Control-E is disabled during INPUT and while in the Monitor. (3) If you try to edit a non-existent line, nothing will happen. (4) If a program line is longer than the maximum allowable length (128 characters for Integer or 239 for Applesoft), the line will be "packed", with all spaces not between quote marks removed (see control-P, page 10).

Control-E is also used to edit Escape functions as described on page 18.

control-E . (RE-EDIT)

Control-E followed by a **period** will put the last line edited into Edit Mode; no need to type the line number.

control-W (IMMEDIATE EDIT)

Control-W allows you to enter the Edit Mode while you are entering text. (Notice that "W" is adjacent to the "E" key.)

Here is an example. Suppose you are typing—

LODE FINANCIAL REPORT NUMBER 123

(you haven't hit Return yet) and you realize you have misspelled "LOAD". Instead of retyping the entire line, type **control-W** to enter Edit Mode, and correct the error with normal editing procedures (described later) and press **Return** (no need to trace to the end unless you have to add to the line). Unlike control-E, control-W may be used to edit text being typed with or *without* a line number. Also notice that you may type a control-W *before* you start typing a line.

In Integer Basic (DOS 3.3 only), control-W retrieves the last line typed. In Applesoft, you may retrieve the last line *edited* by typing control-E followed by a period (see above).

Moving the Cursor

Left and Right Arrows

Typing a Left or Right Arrow is the same as typing a control-H or control-U.

After entering the Edit Mode, you may move the cursor in the usual manner with the Left and Right Arrow keys. Pressing either key will end an insert (see control-I, next page).

The Apple IIe's Up-and Down-Arrows (control-K and control-J) do *not* move the cursor in Edit Mode.

THESE GPLE COMMANDS TAKE EFFECT ONLY AFTER ENTERING EDIT MODE with control-E or control-W.

control-B (BEGINNING)

Control-B moves the cursor to the beginning of the line being edited. Control-B is often used prior to control-F (below).

control-N (END)

Control-N (rhymes with "END") instantly moves the cursor to the end of a line being edited.

control-F (FIND)

Use **control-F** followed by **a character** to find (move the cursor to) the next occurrence of that character. Type the character again (*without* another control-F), and you will fine the *next* occurrence of the character. Any other key terminates Find. **Control-F : : :** etc. will jump the cursor from colon to colon, and therefore, from statement to statement.

Exiting Edit Mode

Return (ACCEPT)

Typing a Return is the same as typing a control-M.

Press Return to exit Edit Mode and accept the edited line exactly as it appears on the screen. It doesn't matter where the cursor is when you press Return.

control-Q (QUIT)

Control-Q works like Return (above), but it deletes all text from the cursor to the end of the line. All text before the cursor is accepted.

control-X (CANCEL)

Control-X exits Edit Mode without any effect on the line being edited, whether or not changes have been made.

GPLE Change-Commands

THESE GPLE COMMANDS TAKE EFFECT ONLY AFTER ENTERING EDIT MODE with control-E or control-W.

control-I (INSERT)

GPLE lets you painlessly insert characters into program lines or any text being typed. Enter the Edit Mode with control-E line-number (or control-W), and move the cursor to the position where you want to insert characters. Type **control-I** (doesn't show) followed by the characters you want to insert. The balance of the line will move to the right. You will continue inserting characters until a **Left Arrow, Right Arrow, Return,** or any **control-character** is typed.

control-O (OVERRIDE)

Type a control-O before every CONTROL-character you want to insert. The control-O won't show and the inserted control-character will appear in inverse. As a test, insert a control-D in the statement:

10 PRINT "CATALOG"

Type control-E 10 (Return) to edit the line. Move the cursor to the "C". Type **control-O control-D** (hold the control key down while you type "**OD**"). You will see the control-D indicated by an inverse "D":

10 PRINT "[D]CATALOG"

Make other changes in the line if you want (remember, Right- or Left-Arrow gets you out of Insert Mode), and press **Return** to exit Edit Mode.

Control-O acts exactly like control-I (Insert) *except* the first character inserted *may* be a control-character. Some control-characters do not require a control-O before insertion, but to play it safe, always type a control-O first.

Besides the "normal" control-characters, you may want to play with inserting backspaces (control-H), returns (control-M) into Print statements, Rems or GPLE Escape functions. See the examples in the Tip Book section of this book.

THESE GPLE COMMANDS TAKE EFFECT ONLY AFTER ENTERING EDIT MODE with control-E or control-W.

control-D (DELETE)

Type **control-D** to delete the character at the cursor. The balance of the line will move to the left. You may, of course, use your Apple's repeat function to delete as many characters as you like.

control-Z (ZAP)

Control-Z deletes or "Zaps" all characters from the cursor to a specified character. Try control-Z on this program line:

10 GOSUB 123: PRINT: PRINT: PRINT "ALPHA PLOT"

Type **control-E 10** to enter Edit Mode. The cursor is on the "G" in "GOSUB". Now type **control-Z P** and the characters from the cursor to the next "P" will disappear. Type **P** and it will happen again. And again. Play around.

Control-Z : : : is a quick way to erase program statements, one for each colon typed. Try it.

control-R (RESTART)

Control-R can be a life saver! It restores the line being edited to its original condition, leaving you in Edit Mode.

Note: Control-R does not work after entering Edit Mode with control-W.

control-P (PACK)

Control-P instantly removes all spaces not between quote marks, thus "packing" a program line. Use control-P if you are editing and hear the bell warning you that a program line is getting too long.

To protect a REM statement from being packed, enter a quote mark as the first character after the word "REM". Quote marks around DATA items will keep them from losing their spaces when packed.

control-C (CONVERT)

Control-C changes the character at the cursor from upper- to lower-case or vice versa. To change many consecutive characters, use your Apple's repeat function.

Global Editing

GPLE features a comprehensive global editor which performs Search, Edit and Replace functions. "Global" functions apply to a program (or a *section* of a program) as a *whole.*

Here are some examples of what you might want to do with GPLE's global features. The possibilities are many—

SEARCH: You could tell your Apple to find and show you all occurrences of the variable X, or the string "DOG" or the command "GOSUB".

EDIT: GPLE lets you find and edit every line that contains a specified string or variable.

REPLACE: You could tell your Apple to replace all occurrences of the variable X with the variable XREF, or all GOSUBs with GOTOs, or all "Horses" with "Cows".

GENERAL FORMAT

THE EXAMPLES ON PAGE 13 ARE WORTH AT LEAST 1,000 WORDS.

To use GPLE's Global features, always start with **control-E** followed by information that tells GPLE exactly what you want edited and how. In normal line editing, the only information necessary after control-E is the line number you want to edit (for example, control-E 10). The general format for all GPLE editing, global and non-global is:

control-E L1, L2, "STR1", "STR2"/O (Return)

Each item is optional, although some don't make sense without one or more of the others. If you hear a beep, you did something wrong; try again.

L1 is the starting line number in a range. If you are editing only one line, L1 is *the* line number to be edited.

L2 is the end line number in a range.

STR1 is a string of characters (16 characters maximum) that you are searching for. This string must be enclosed in quotes. GPLE even lets you enclose a quoted statement in quotes (for example **"**"HORSE"**"**).

STR2 is the replacement string (16 characters maximum) that will replace STR1.

/O is an option flag, either a "/F" or a "/R" (page 12).

SEARCH & REPLACE PROCEDURE

THE EXAMPLES ON THE FOLLOWING PAGE ARE WORTH AT LEAST 1,500 WORDS.

After typing a global command (examples next page), GPLE will search and display each appropriate line IN EDIT MODE, allowing you to edit it. If you had requested a Replace, the replacement will be shown.

If you like what you see, press **Return**. If you want to restore the line to its previous condition, type **control-R**. If you want to halt the entire search, type **control-X**.

SEARCH PECULIARITIES

A search string will not find a match that is part of a longer string (exception: see /R below). This command:

EDIT "A"

will search for the string "A" or variable A in an entire program. It will *not*, however, find the "A" in "HT**A**B or in "PRINT "**A**LL"", but it *will* find the "A" in "**A** TEST" and in "SHAKE **A** LEG", and in "**A** = B + C". To be found, a string must be enclosed by two *non-alphabetic* and/or *non-numeric* characters such as a space or quote mark.

/**R** (RAW SEARCH)

Note: /R and /F may not be used in the same command.

Ending a search command with /**R** will cause a match even if the search string is part of a larger string. If GPLE alters lines you don't want changed, type control-R to restore that line.

/**F** (FAST SEARCH)

Ending a search command with /**F** will execute a global Search or Replace without letting you edit each line. Instead, program lines will simply be printed on the screen as they are located. Replacements (if any) will be made without asking your approval. **Control-C** will let you bail out if you start noticing things you don't like.

? (WILD CARD)

A question mark can be used as a "wild card" character that will match *any* character. **EDIT "A?"** means edit all occurrences of two-character variables starting with "A". **EDIT "P??K"** means edit all occurrences of four-letter words beginning with "P" and ending with "K". **Watch out!** GPLE's wildcard feature is a bit unpredictable.

Global Editing Examples

GLOBAL EDIT

Control-E is represented in these examples by the word "**EDIT**".

EDIT 10,50 Edit all lines between 10 and 50.
EDIT ,50 Edit all lines from program start through Line 50.
EDIT ., Edit all lines from the last line edited through the end.

GLOBAL SEARCH

EDIT "PRINT" Edit all lines containing the string or command "PRINT" (won't find the "PRINT" in "PRINTS").
EDIT 10,30,"FIG" Edit all lines, from 10 to 30, containing the string "FIG" (won't find the "FIG" in "FIGURE").
EDIT 10,,"FIG"/R Edit all lines, from 10 to the end, containing the string "FIG" (will find the "FIG" in "FIGURE").
EDIT "REM"/F Display all lines with REM statements.
EDIT "D"/R Edit all lines with a control-D. The inverse "D" is entered by typing control-O control-D (see page 9).
EDIT "X?" Edit all lines with two-character variables starting with X (X%, XA, X1, etc.).

GLOBAL REPLACE

EDIT "PADDLE","MOUSE" Change the word "PADDLE" to "MOUSE" in the entire program.
EDIT "I","AMT" Change all I variables to AMT. Watch out, because this command also tells GPLE to change things like PRINT "I WIN!" to PRINT "AMT WIN!".

Notice that the search strings may come before the line numbers.

EDIT "X","Y",10,50/F Change all occurrences of "X" to "Y" in Lines 10-50. The "/F" means don't ask for approval.
EDIT "X","Y". Change X to Y in the last line edited.
EDIT "HOME","" Delete all HOME commands.

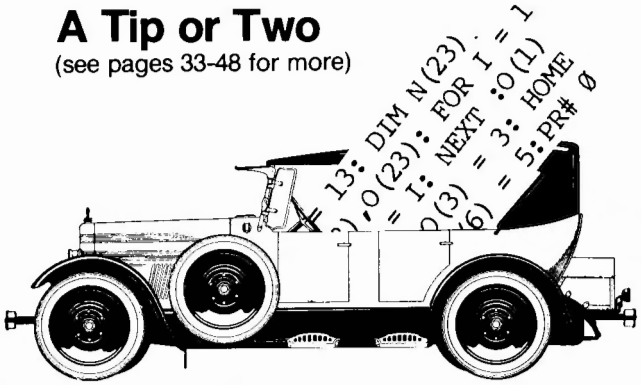
SEARCHES AND LONG PROGRAM LINES

A Search or Replace will sometimes fail on very long program lines, because GPLE "auto-packs" them and removes unneeded spaces. You must use /**R** Raw Mode to have a successful Search on certain long lines.

EXTRA-LONG lines cannot be edited with GPLE at all. This rarely is a problem, however.

A Tip or Two

(see pages 33-48 for more)



MOVING PROGRAM LINES

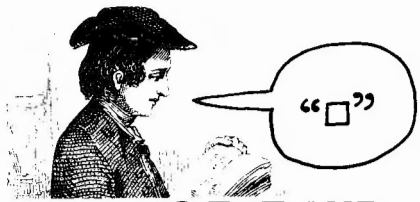
To move a line in a program, just EDIT it, type a new line number, hit Return and delete the old line. For example, to move Line 10 to Line 22, type **control-E 10**, change the 10 to a **22**, hit **Return**, and type **10** (Return).

GPLE.??

You can have your programs check to see if GPLE is in memory, and if so, exactly which version. This check is virtually semi-foolproof:

These Peeks apply only to DOS 3.3. For ProDOS, you can Peek location **1007**. A value of **1** means GPLE is present.

```
IF PEEK(1005)=1 THEN PRINT "GPLE.48"
IF PEEK(1005)=2 THEN PRINT "GPLE.LC"
IF PEEK(1005)=3 THEN PRINT "GPLE.DM"
IF PEEK(1005)=234 THEN PRINT "NO GPLE"
```



HERE'S THE RUB

GPLE makes the Apple IIe's **Delete** key act as a backspace instead of printing the cursor character as it was designed to do. This makes backspacing a lot easier. If you miss the nice little square this key used to print, just tell your programs to PRINT CHR$(127) instead of PRINT "□". This unique "Rub" character, actually a lower case underscore(!), is the only one that looks decidedly different on a IIe (it prints as a speckled square) than it does on a II or II+ (a solid square). On most printers we know, it won't print as anything!

GPLE Escape Functions

Escape functions are user-definable two-key commands (ESC plus another key) that can be programmed to perform any typeable task. An identical "Escape table" containing many Escape functions is built-in to each version of GPLE. A bunch of programmers sat around and decided what should go into this "standard" table (next page). Undoubtedly you will disagree and want to make some changes. Instructions for doing so begin on page 18.

An Escape function is called by pressing the **ESC** key (and releasing it), then the key that has been assigned to a particular function. For example, with GPLE loaded, type:

ESC 1 (don't hit Return)

Your Apple will automatically type "**CATALOG,D1**" and a **Return**, thereby displaying a disk catalog of drive 1. The free space on disk will be printed at the end of the catalog, as if you had typed ESC S (try ESC S while you're at it). Similarly, **ESC 2** will catalog drive 2.

Escape functions operate something like disk EXEC files; the automatically-typed characters are executed *as if you had typed them yourself.* Unlike EXEC files, the access time of Escape functions is almost instantaneous.

EXAMINING THE ESCAPE TABLE

To see the Escape functions that are in memory, type:

control-E ? (Return)

Type control-P to "page" the list forward just like a normal listing. The Escape character appears in the left column followed by a hyphen.

Control-characters appear on the screen in inverse—Inverse-C is control-C, inverse-M is control-M (carriage return), etc. ESC is represented by its control-**[** equivalent. Don't confuse **control-[** (ESC) with **control-]**, which is used in nested and repetitive functions (see page 19).

An apostrophe (**'**) means that the instructions which follow will not print when executed (the *results* will print, however).

You can print your current Escape functions on your printer (see page 21).

Standard GPLE Escape Functions

The following Escape functions are included in your three versions of GPLE. See the Tip Book section of this book for possible changes.

ESC control-D displays your GPLE version number.
ESC control-L removes GPLE from memory.
ESC control-P moves Applesoft program start to 16384 ($4000), just above hi-res page 1. Type ESC control-P *before* loading your program.
ESC control-Q exits 80-column mode.

To use this append, you must renumber your programs properly. Beagle Bros' DOUBLE-TAKE merges programs without renumbering.

ESC control-N "hides" an Applesoft program so another may be appended to it. Load a program. Type ESC control-N to hide it. Load a second program. Type ESC control-^ to append the programs.
ESC control-^ completes the append described above. (^ is shift-6 on Apple IIe's and shift-N on older models.)

‡ **ESC control-T** "hides" an *Integer* program (see above).
‡ **ESC control-R** appends *Integer* programs (see above).

Use these functions on pre-IIe Apples to type the "untypable" characters.

ESC ! types a left-bracket
ESC " types a backslash.
ESC # types an underscore.
ESC $ types a control-left-bracket.
ESC % types a control-backslash.
ESC & types a control-underscore.

ESC / types "PRINT" (no return).
ESC 0 clears the screen.

ESC 1 catalogs drive 1 (default slot).
ESC 2 catalogs drive 2 (default slot).
ESC 5 catalogs slot 5 (default drive).
ESC 6 catalogs slot 6 (default drive).
ESC 4 switches to 40-columns.
ESC 8 switches to 80-columns.

Enter & exit monitor keys are grouped for the IIe and non-IIe.

ESC : enters the monitor.
ESC - exits the monitor.
ESC * enters the monitor.
ESC = also enters the monitor.
ESC Delete (Apple IIe only) exits the monitor.

ESC A, B, C, D and **ESC I, J, K, M** move the cursor exactly as they do without GPLE (see your Apple manuals).

ESC E and **F** clear text exactly as they do without GPLE (see your Apple manuals).

ESC H displays control characters as inverse. Type "**&**" or CALL 1016 to turn this feature off.

ESC L lists the program currently in memory.

ESC N prints the negative or positive equivalent of any number. Let A=a number, then type ESC N.

ESC O executes a "PR#0" (printer off).

ESC P executes a "PR#1" (printer on).

ESC Q prints the decimal value of any two consecutive bytes in memory. Type "A=n", where n is the address of the first byte in the pair, then ESC Q.

ESC R executes a RUN command.

‡ **ESC S** prints the free sectors on the last disk catalogged.

ESC T executes a TEXT command.

ESC U sets up an additional GPLE recovery vector at $0A by copying the contents of $3F8. Helpful if you are going to run a program that uses both the control-Y and ampersand (&) vectors. To recover GPLE from the monitor, type "AG". From Applesoft, CALL 10 or use "X=USR(0)".

Very technical. A good function to delete if you're not too technical yourself.

ESC V moves the cursor to VTAB 1.

‡ **ESC W** prints the starting address and length of the last BLOAD in hex (use ESC W immediately after a BLOAD).

ESC X prints the number of free bytes in the Escape table.

ESC < moves the cursor 40 spaces to the left (repetitive).

ESC , moves the cursor 8 spaces to the left (repetitive).

ESC > moves the cursor 40 spaces to the right (repetitive).

ESC . moves the cursor 8 spaces to the right (repetitive).

NOTE: You might find Escape functions in your standard table that are not listed here. Look around; these are probably called by *other* functions. For example, **ESC control-C** ("Catalog") is called by ESC 1, ESC 2, ESC 5 and ESC 6.

ANOTHER NOTE: Some Escape functions, like ESC E and ESC F, "call themselves", in order to preserve their original characteristics. Consider this before changing one of these functions.

‡DOS 3.3 only (not applicable to ProDOS)

Defining Escape Functions

Any immediate mode command or string of characters may be defined as an Escape function. Each function may be as simple or as complex as you like.

To define or edit any function, type **control-E** followed by **ESC** and the **key to be edited**. This will place you in EDIT MODE, just as if you were editing a program line. All of the GPLE commands discussed in previous pages will be in effect.

A TEST

Let's program **ESC Z** to clear the screen & print "DOGFOOD":

1. Type **control-E ESC Z** (think, *"Edit Escape Z"*). A "Z" will appear with a blinking cursor. If no other characters appear, then the ESC Z function is unused.

2. Type **HOME : PRINT"DOGFOOD"M**. The M is a control-M (carriage return). Insert it into the line by typing **control-O control-M**. Without the carriage return, ESC Z would print "PRINT"DOGFOOD" " on the screen but not execute it.

3. Hit **Return** to finish.

4. Test the results by typing **ESC Z**. If it doesn't clear the screen and print "DOGFOOD", go back to step 1.

ANOTHER TEST

Let's re-program **ESC H** to print "HICCUP" with a beep. ESC H is a "standard" GPLE function programmed to show control-characters in inverse. We will replace this with a new function.

1. Type **control-E ESC H**. An "H" will appear, followed by programming code which we will replace.

2. Type this right over the existing command:

'PRINT"HICCUPG'M . The beginning apostrophe prevents the command from printing, but still lets the *results* of the command print. The inverse-G is a control-G beep.

3. Type **control-Q** to chop off the old command to the right of the cursor. That's all there is to it.

4. Test the results by typing **ESC H**. The original function we deleted may be restored by re-booting.

NESTING ESCAPE FUNCTIONS

ESC X (unless you changed it) will display the number of free characters in the Escape table.

GPLE allows up to 1152 characters in its Escape table. If you enter an Escape command that causes the total to exceed this limit, it will only be partially accepted (a beep will be heard). One way to gain space is to delete rarely used functions. Or...

To save space and avoid repetitive words, you may "nest" commands, having one command call another, a second call a third, and so on. GPLE lets you do this up to eight levels deep. The third function, after execution will return to the second and resume execution where it left off, the second will return to the first, and so on.

A Nested Example: In the standard Escape table on page 16, **ESC 1**, which catalogs drive 1 and prints free space on disk, is stored as two nested functions and only three other characters— **]C ,D1M] S**

]C calls **ESC control-C**, which types "CATALOG".
,D1M prints "**,D1**" and does a **Return**.
] S calls **ESC S**, which prints the free space on the disk.

Since there are four keys in the Escape table (1, 2, 5 and 6) that catalog a disk, much space is saved by calling repeated words in this manner.

Don't confuse control-] with control-[.

The nesting character ("] ") is control-]. Insert it like you would any other control-character. (On non-IIe Apples, "]" is shift-M.)

REPETITIVE FUNCTIONS

Escape functions may be defined as "repetitive" by inserting a control-] ("] ") as the last character. After a repetitive function has been executed, you can execute another Escape function (or the *same* function) without needing to hit ESC first. This is helpful in screen editing and rapid cursor movement.

For example, to cursor-trace an entire line from the screen, place the cursor on the first character and type **ESC.....** (no ESC between periods). To resume normal typing after a repetitive Escape function, you must terminate with an unprogrammed key. The Space Bar is a good choice, since it is probably unused.

RENAMING ESC FUNCTIONS

Suppose you wanted ESC G to catalog your disk instead of ESC 1. Edit ESC 1 with the command:

control-E ESC 1

Back the cursor up one space to the **1** at the left margin, type a **G** and hit **Return**. That's it. Now, both **ESC G** *and* **ESC 1** will catalog drive 1. Let's delete ESC 1...

DELETING ESC FUNCTIONS

To delete **ESC 1** (as an example), type:

control-E ESC 1

Leave the cursor where it is and type **control-Q**. ESC 1 will now perform no function at all.

'NON-PRINTING FUNCTIONS

Inserting an apostrophe stops the printing of an Escape function from that point on. The function's results will still print. This ESC Z example:

ZPRINT"ABC";:'PRINT"DEF"M

will produce this when ESC Z is typed:

]PRINT"ABC";:
"ABCDEF"

You cannot turn the apostrophe's effect back "off" after you have turned it "on".

ESC CURSOR-MOVES

With GPLE, there are three ways to do single ESC cursor moves:

1. **ESC A, B, C** and **D** (non-repetitive)
2. **ESC I, J, K** and **M** (repetitive)
3. **ESC Arrows** (repetitive)

There is no reason not to delete two of these methods if you need space in your Escape table. Method 1 is an obsolete left-over from the old Apple days. Methods 2 and 3 are identical— If you use an Apple IIe, take your pick. If you don't, you have no up and down arrow keys, so delete Method 3.

Old GPLE users will notice that **ESC .** and **ESC ,** have replaced ESC Left-Arrow and ESC Right-Arrow as the multi-step moves. Re-define them if you want, to suit your style.

ESC TO THE MONITOR

In case you hadn't noticed, an Escape function may enter the monitor (CALL-151), perform a machine-language function, and then exit back to Basic (3D0G). Look at the machine code samples in the standard Escape table (ESC H, ESC S and ESC W) for more tricks.

Saving an Escape Table

After you have created an Escape table that you like, you need to save it on disk so it will take effect the next time you boot. There is a program on the GPLE disk called "ESCAPE SAVE" that will write your Escape table into one of the three versions of GPLE. To run it, type:

In ProDOS, this file is called **ESCAPE.SAVE** (with a period).

RUN ESCAPE.SAVE (ProDOS)
BRUN ESCAPE SAVE (DOS 3.3)

In response to "GPLE FILE NAME: ", you must answer "GPLE" (if ProDOS), or "GPLE.48", "GPLE.LC" , etc. (if DOS 3.3), depending on which version you intend to use. You may specify slot and drive with your response (for example, "GPLE.48,S6,D2"). Your Escape table will be written into that file and the old Escape table will be over-written. Now, when that version of GPLE is loaded, your Escape table will be in effect.

You may want to have several different versions of GPLE on different disks, each with a different Escape table.

Saving an Escape table will RE-WRITE GPLE, and, in effect, ERASE your old Escape table. **MAKE BACK-UPS** of versions of GPLE containing Escape tables that you want to keep.

ESCAPE SAVE will not work properly with ProntoDOS.

Printing an Escape Table

To send the Escape table currently *in memory* to your printer, turn the printer on and activate it with a **PR#1** command (PR#slot), then:

In ProDOS, this file is called **ESCAPE.PRINTER** (with a period).

BRUN ESCAPE PRINTER

The resulting printout will display control-characters as lower case instead of inverse.

ESCAPE PRINTER will not work properly with ProntoDOS.

GPLE List Control

ESC or **control-S** (PAUSE)

To pause a listing, press the **ESC** key, or the usual **control-S** command. To resume listing, hit any key. If ESC doesn't pause your listings, GPLE has probably become "disconnected" with a command like PR#0 (see page 29). To re-connect GPLE, type control-E (return). Beagle Bros' DOUBLE-TAKE utility, which lets listings scroll in both directions, will also respond to ESC as a pause key.

control-C (CANCEL)

To terminate a listing entirely and return to Basic, type a **control-C**. Without GPLE you get a "BREAK" (beep) message. With GPLE, you don't.

control-P (PAGE)

To advance a listing one screen-full or "page", type a **control-P**. Any other key will continue listing normally. If control-P isn't working, re-connect GPLE with control-E (return).

GPLE Lower Case Entry

The Apple IIe allows you to enter and display lower case text using the shift key; no problem. You will probably want to disconnect the **control-A** shift feature described below with CONFIG GPLE or the **control-O L** option (page 26).

If you use an Apple IIe, ignore this, and disconnect control-A with the CONFIG GPLE program or the control-O L Option (page 26).

control-A (UPPER/LOWER CASE TOGGLE)

Apple II's and II+'s have no built-in lower case capabilities. GPLE will solve half of the problem by allowing lower case text entry. Yet, unless your non-IIe Apple is equipped with hardware (such as a Paymar Adaptor or an 80-column card) or software (such as Flex Type) for *displaying* lower case, lower case will display as unintelligible garbage on your monitor screen. Printer output *will* be readable, however.

To enter lower case, type **control-A** followed by your lower case text. Another **control-A** will cause the next character only to be printed in upper case. Yet *another* **control-A** (that's two in a row) will lock you in upper case. This command is available in Edit Mode as well as in immediate mode.

There is a one-wire shift key modification that can be made to your Apple II or II+ (see the article by Dan Paymar in the May 1982 *Call -A.P.P.L.E.* magazine). This modification makes your shift key perform as it should; it works with your Apple in lower case mode (after typing one control-A).

COMMANDS TYPED IN LOWER CASE

GPLE lets lower case DOS, Basic and monitor commands work as if they were typed in upper case. A lower case "print" or "catalog" command, for example, will not cause a ?Syntax Error or Syntax Error as it would without GPLE. All lower case text in program lines not between quotes will be converted to upper case when listed.

Quote marks are necessary around lower case strings in DATA statements and around REM statements containing lower case. Without the quote marks, lower case will be converted to upper case.

The Edit Mode **control-C** command is handy for converting lower case at the cursor to upper case and vice versa.

GPLE and 80-Column Apples

To display 80-columns instead of the "normal" 40, your Apple must have an "80-column card" installed. Unfortunately, the manufacturers of these cards have used no standard format for their hardware, and GPLE must be re-configured for each particular brand. See CONFIG GPLE (page 26), which allows you to select one of the supported 80-column cards (or none). At press time, these cards are supported by GPLE:

Apple IIe 80-column card
ALS Smarterm
M & R Sup'R Term
Videx Videoterm
Vista Vision 80
Wesper Wizard-80

Because of hardware limitations peculiar to some of these cards, not all of GPLE's features will be available. For example:

Vision-80: This card's firmware assumes that standard 48K DOS is present, thus it cannot be used with GPLE.DM. Also, CONFIG GPLE and ESCAPE SAVE must be accessed from 40-column mode only.

Regardless of hardware, when a brand new copy of GPLE is first loaded, the display is automatically set to 40-column mode, and it is assumed you have an Apple IIe 80-column card installed. These defaults may be changed with CONFIG GPLE (page 26).

IMPORTANT: When GPLE is in effect, you should NOT attempt to use any of the control characters or commands specified in your 80-column card manual, as these are disabled by GPLE.

If no 80-column card is present, 80-column commands will be ignored.

ALSO IMPORTANT: Do not use PR#3 (PR#slot) to enter 80-column mode; you will murder GPLE (to un-murder it, hit Reset). Instead, use any of these commands:

Switch to 80-columns: **control-V, control-O 8** or **ESC 8**.
Switch to 40-columns: **control-V, control-O 4, ESC 4** or **ESC control-Q**.

GPLE's Type-Ahead Buffer

Although it has its place, under most circumstances, we recommend you don't use the type-ahead buffer; it can be a trouble-maker.

When typing with an Applesoft prompt, GPLE will store every character that you type, even if they are typed during program execution or during a printout. This means that you can type commands without waiting for previous commands to finish executing. For example, during a listing, you could type a "CATALOG" command. When the listing is complete, your disk will catalog.

Unfortunately, the type-ahead buffer will "eat" keystrokes typed while peeking the keyboard strobe. Programs such as the following one will not work:

```
10 PRINT "PRESS THE SPACE BAR."
20 KEY=PEEK(-16384)
30 IF KEY=160 THEN GOTO 50
40 GOTO 20
50 POKE -16368,0
60 PRINT "NICE JOB."
```

To turn off the type-ahead buffer, type **control-O N**. To turn it back on, type **control-O T**. OR BRUN CONFIG GPLE, and answer "N" to "Enable Typeahead?". The type-ahead buffer will be off if you boot a brand new GPLE disk.

TYPE-AHEAD FACTS

If you type more characters than GPLE's buffer can hold, the bell will sound and subsequent characters will not be saved.

In Integer Basic, the type-ahead feature works only during character output; for example, while listing a program.

Occasionally, the type-ahead buffer will contain keystrokes you don't want. To "flush" the buffer and remove all stored characters, type **control-C** or **control-X**.

GPLE immediate mode command keys, control-P, control-S and ESC will not be stored by the buffer.

Some characters may be lost if typed while the disk drive is spinning. It is best to not type while your disk drive's "In Use" light is lit.

GPLE Options Command

The **control-O** Options command lets you turn certain GPLE features off or on any time you want. The CONFIG GPLE program (below) lets you determine which of these features will be in effect when you *first load* GPLE.

Here are the options selectable from immediate mode:

control-O 4 enters 40-column mode.
control-O 8 enters 80-column mode.

control-O T enables the type-ahead buffer.
control-O N disables the type-ahead buffer.

control-O U enables the control-A shift command.
control-O L disables the control-A shift command.

Config GPLE

In ProDOS, this file is called **CONFIG.GPLE** (with a period).

CONFIG GPLE will not work with ProntoDOS in memory.

CONFIG GPLE will write these changes onto disk, into the GPLE version you have selected. This save will not affect GPLE in memory.

The "CONFIG GPLE" program determines which 80-column card will work with GPLE and which GPLE features will be in effect when you boot. (Also see "Options" above.)

RUN CONFIG.GPLE (ProDOS), or...
BRUN CONFIG GPLE (DOS 3.3)
Answer the questions on the screen:

GPLE FILE NAME: Answer "GPLE" (if ProDOS), or "GPLE.48", "GPLE.LC", etc. (if DOS 3.3.), depending on which version of GPLE you want to change.

80-COLUMN CARD: Type the number of the card you are using. If it isn't listed, select "Other" and you will be asked for the "DRIVER FILE NAME". Type in the name of the appropriate "Video Driver" file (ending with ".VID"). If there is none on the disk for your 80-column card, contact Beagle Bros. Maybe we can help; maybe we can't.

DEFAULT 40 COLUMNS? Type "Y" if you want GPLE to come up in 40-column mode. Type "N" for 80.

ENABLE TYPE-AHEAD? Type "N" to have the type-ahead buffer disabled when you boot.

ENABLE CTRL-A SHIFT? Type "N" if you have an Apple IIe. Type "Y" for a II or II+ with a non-functioning shift key.

GPLE DOS Mover

by Cornelis Bongers

(**DOS 3.3 ONLY**—ProDOS cannot be moved.)

GPLE DOS MOVER is a powerful utility that will move DOS into the uppermost 16K of an Apple IIe or onto the Language Card or comparable RAM card of an Apple II or II+, and make the space normally occupied by DOS (over 10,000 bytes) available for your programs. To move DOS, simply type:

BRUN GPLE DOS MOVER

GPLE DOS MOVER will move DOS and then run any program on the disk named "HELLO". GPLE's HELLO program will automatically BRUN GPLE.DM and put GPLE on the Language Card with DOS (see page 4).

DOS MOVER LIMITATIONS

1. Only up to five files can be "open" at a time (MAXFILES 5). Refer to your DOS Manual to read about open files.

2. INIT does not work normally. You may INIT a disk in the usual manner, but it will not have an image of DOS on it, as is usually the case (the INITialized disk will not, therefore, boot). This may be overcome by running MASTER CREATE (on your System Master disk).

3. FID, MUFFIN and RENUMBER must be modified to work correctly, using the appropriate "FIX" program(s) on the GPLE disk (page 28).

4. Moving DOS prevents you from loading a second language into RAM.

5. ProDOS cannot be moved.

More about moving DOS can be found in the July-August and November-December 1981 *Call-A.P.P.L.E.* magazines.

Beagle Bros also publishes a DOS Move utility called "DOS-UP", on Tom Weishaar's **ProntoDOS** disk. Compatible with GPLE, ProntoDOS triples the speed of loading and saving files. Contact Tom Weishaar, 913-649-0567, if you want to license ProntoDOS or DOS-UP for use in your programs.

Removing GPLE

In ProDOS, this file is called **REMOVE.GPLE** (with a period).

A GPLE memory conflict will usually only occur with GPLE.48 and very large programs that attempt to locate themselves in GPLE's location above the DOS Buffers (see inside back cover). In this rare instance, you might get a "Program Too Large" message when attempting to load a program, or an ?Out of Memory Error during program execution.

In this unlikely event, yu can **BRUN REMOVE GPLE** to disable GPLE and return Himem and the DOS Buffers to their normal locations (see inside back cover).

AN EASIER WAY TO REMOVE GPLE is to type **ESC control-L** (assuming this standard command is still in your Escape table).

REMOVE GPLE cannot be BRUN from within a program, since doing so will clear memory and wipe out that program. Removing GPLE.LC and GPLE.DM will not affect a program in memory.

Fixers for GPLE.DM

The GPLE disk includes accessory routines that will make certain utilities on your System Master disk work with GPLE DOS MOVER. You should copy FID, MUFFIN and/or RENUMBER onto your GPLE disk. To use the three fixers, type these three commands:

EXEC FIX FID (return)

EXEC FIX MUFFIN (return)

EXEC FIX RENUMBER (return)

Each FIX file will load the appropriate program, fix it, then save it under the same name onto the same disk. The new saved versions should work with or without GPLE.DM.

Technical Info

THE INITIALIZATION PROCESS

DOS 3.3 only. Not applicable to ProDOS.

GPLE.48 will set Himem 4096 bytes lower than normal. It also concludes initialization by executing an INT or FP command, thus any program in memory will be lost. GPLE.LC and GPLE.DM will not affect any program in memory when they are BRUN or removed, nor will they occupy user RAM.

All versions of GPLE are unaffected by MAXFILES, INT or FP commands.

RE-ENTERING BASIC

99% of you can ignore this paragraph.

If your Apple has an Autostart ROM (most do), Reset will always return you to (or keep you in) Basic with GPLE up and running. On an Apple II or II+ without an Autostart ROM (a rare machine), you should always exit the monitor with **control-Y**. If, instead, you press Reset and type **3D0G**, GPLE will be disabled until Basic is re-entered with a CALL 1016 (or a monitor control-Y).

RECONNECTING GPLE AFTER PR# AND IN#

A PR#0 command will temporarily "disconnect" GPLE's Listing control features. Any EDIT command (even **control-E Return**) will re-connect GPLE. IN#0 will disconnect GPLE completely. If control-E fails to re-connect GPLE, try **&** or **CALL 1013** or **Reset**. One of these commands will work.

VECTOR REPAIR

CALL 1013 and/or **&** will be disabled if you run a program that uses control-Y, CALL 1013 or **&**, for other purposes. It is a good idea to write down the data found at **3F5.3FA** so you can re-establish these vectors by simply re-entering six bytes:

DATA FOUND AT 3F5: .....

Troubleshooting GPLE

Control-E won't edit: Type **&** or **CALL 1013**, or hit **Reset**.

No keyboard response: If a program asks you a question and won't accept anything as an answer, it may be caused by GPLE's type-ahead buffer (see page 25). Cancel the buffer's effect by typing **control-O N**.

GPLE.48 won't load: You may have INITialized a disk with GPLE.48 in memory. Boot a non-GPLE disk such as the System Master and then re-INITialize. Never INIT a disk with GPLE.48 in memory.

?Out of Memory Error: If you get this error (or "Program Too Large") with GPLE.48 in memory, but not without it, your program is too big. You should use one of the other two versions of GPLE (or none at all) with this particular program.

I/O Error during Bload: You might be Bloading on top of GPLE. Bload elsewhere or remove GPLE from memory.

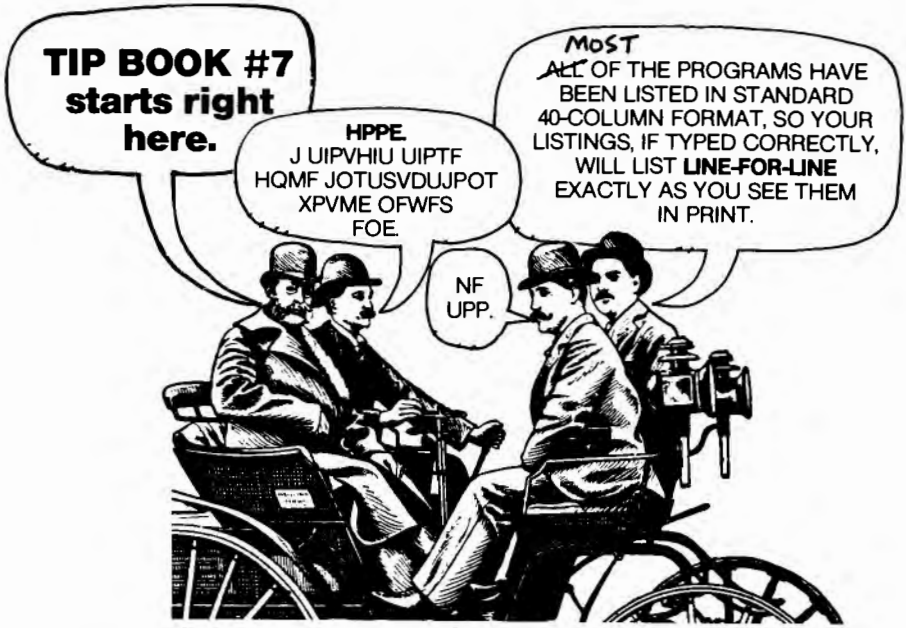
System hangs with GPLE.LC: Perhaps you hit Reset instead of control-C to terminate a listing. To prevent the problem, always put something (Integer, Applesoft or Beagle Basic) on the Language Card **with** GPLE.LC.

System hangs on an 80-column command: BRUN CONFIG GPLE to configure GPLE for your particular 80-column hardware.

Your program won't GET control-A: Disable control-A with a **control-O L** command.

Listings won't pause or page: This is usually caused by a PR#0 command. EDIT anything (or nothing; **control-E Return**) for a quick fix.

CONFIG GPLE or ESCAPE SAVE problems: Perhaps you have ProntoDOS in memory. These two programs won't work with ProntoDOS. GPLE itself will.



SECRET DE-CODER

To translate what the two guys in the middle are saying, type in Line 20 as shown below and add a Line 10 that sets A$ equal to what it looks like each man said.

Remember HAL, the wise computer in the movie *2001, A Space Odyssey*? Rumor has it that he got his name using a program something—but not quite—like this.

```
20  HOME : FOR L = 1 TO  LEN (A$)
     :A =  ASC ( MID$ (A$,L,1))
25 A = A - (A > 64) - 19 * (A = 3
     2): PRINT  CHR$ (A);: NEXT
```

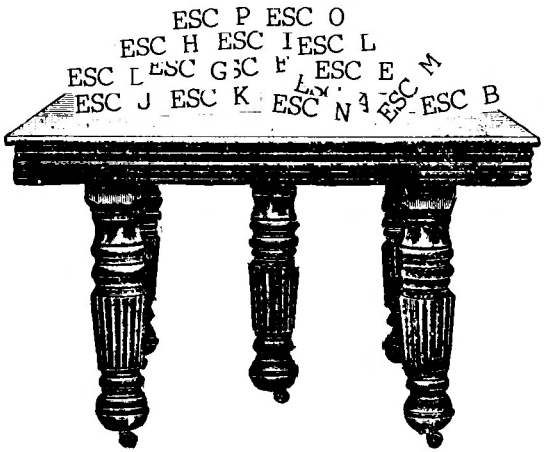
"QUOTES "WITHIN" QUOTES"

You probably thought you couldn't put quote marks inside of a quote statement. Well, run this program, then list Line 20. The carets will have been replaced by quotes. Trouble is you can't re-run the program or re-encounter Line 20 without hanging. **Page 47 has a better way of printing quotes.**

```
10  HOME : ONERR  GOTO 60000
20 A$ = "THIS IS NOT AS ^USEFUL^
     AS IT LOOKS.": GOSUB QUOTE
30  PRINT A$: LIST 20: END
60000 ER =  PEEK (220) +  PEEK (2
     21) * 256: FOR EE = ER - 2 TO
     1 STEP  - 1:P =  PEEK (EE): POKE
     EE,P - 60 * (P = 94): IF P <
      > 34 THEN  NEXT
60001  POKE ER + 1,178: RESUME
```

The GOSUB QUOTE is illegal, causing an ONERR jump to Line 60000.

Line 60001 changes the GOSUB to a REM, so the program can RESUME.



ESCAPE TABLE TIPS

The standard Escape table on your GPLE disk (see page 16) is totally flexible. You should change it to fit your Apple and your programming style. Here are some suggestions; take each one with a grain of silicon—

■ First of all, eliminate any functions you don't need. The space you save can be put to better use.

■ Delete **ESC control-T** and **ESC control-R**. These Integer append routines are useful only if you program in Integer Basic. Most of us don't.

■ Delete **ESC U**. Unless you understand it.

■ Delete **ESC A,B,C,D** or **ESC I,J,K,M** or **ESC Arrows**. You undoubtedly won't use all three cursor-move methods.

■ Delete **ESC S** if you don't need a Free Space function.

Multiple statements in Integer Basic must be separated by carriage returns instead of colons.

■ Expand **ESC L** to **SPEED=255: TEXT: HOME: LIST** (return). That way you can LIST directly from graphics without needing a TEXT command. But wait— The TEXT command will switch Flex Type to text when you don't want it to, and it could cause trouble with some 80-column cards.

■ Likewise, **ESC 0** (zero) can do a **TEXT: HOME: NORMAL** instead of just a CALL-936 (the equivalent of HOME).

■ If you make the two changes above, you probably won't need the **ESC T** TEXT command.

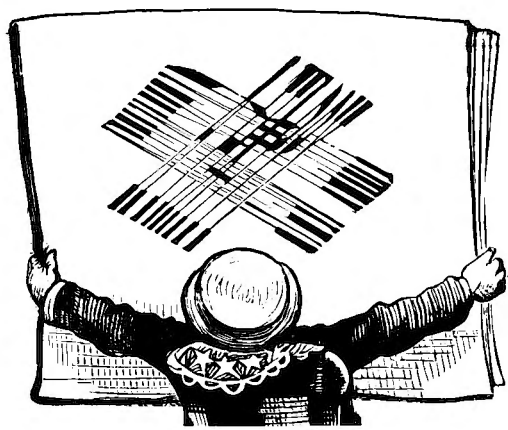
■ You may want to make **ESC Arrows** jump the cursor instead of ESC . and ESC , . Older versions of GPLE used the Arrows and you may be used to that system.

■ If you list to your printer a lot, set up an Escape command that turns the printer on (**PR#1**), lists your program (**LIST**), and turns the printer off (**PR#0**). You will need carriage returns (control-M's) between each command; colons would cause a Syntax Error.

■ *Any* time you are typing repetitive copy— your name and address, a particular program line, or even a complete subroutine— set up an Escape function to type it for you.

■ Set up a command like **ESC Z** to be **LOAD** [].....[M] That way you can position the cursor next to a file name in a catalog and type ESC Z to type "LOAD" and trace over the file name in no time! You will need a space after "LOAD" to erase the file's sector number.

■ **ESC !, ESC "** and **ESC #** print the un-typable characters on older Apples, but aren't necessary if you use an Apple IIe. Some II+ users prefer **ESC /** for the backslash, **ESC hyphen** for the underscore and **ESC]** (shift-M on the II+) for the left-bracket. While we're on the subject, eliminate **ESC $, ESC %** and **ESC &**. Who needs them anyway?



LOWER CASE DATA

Lower case strings in Data statements are perfectly legal. With GPLE in charge, however, you can't *enter* or *edit* lower case strings without enclosing each one in quote marks. The reason why is outrageously boring.

?LINE MISMATCH

Running the program below will produce a "?Syntax Error in 10" error message.

```
10 DATA 1, 2, 3, 4, FIVE, 6, 7, 8
20 READ A: PRINT A: GOTO 20
```

Wrong! Line 10's syntax is perfectly legal. The problem is in Line 20, where we are trying to read a string ("FIVE") as a numeric variable (A). Next time Applesoft accuses you of an improper data statement, this might be your problem.

TOWERS OF HANOI

This is a puzzle involving three vertical pegs. On peg #1 there is a stack of "disks" (not floppy) of decreasing sizes. The object is to transfer these disks, one at a time, in their *same order,* to peg #2. Use peg #3 as temporary storage. The big stickler is that you can **never place a disk on top of one that is smaller.** Run the program that follows, and your Apple will solve the puzzle for you. Use the Return key to view the process step by step. You may want to alter the program so that you can make the moves yourself.

One of the neat tricks in this program occurs in Line 1000, which GOSUBs itself repeatedly until N(TS) equals zero.

```
20  DIM N(20),X(20),Y(20),Z(20)
30  TEXT : HOME : INVERSE : PRINT
     " TOWERS OF HANOI ": PRINT :
     NORMAL : INPUT "NUMBER OF D
    ISKS (1-10):";N$:N =  VAL (N
    $):N =  INT (N): IF N < 1 OR
    N > 10 THEN 30
35  HOME : PRINT "<ANY KEY> TO PA
    USE, <SPACE> TO CONTINUE": PRINT
    "<RETURN> TO STEP THROUGH"
```

(continued)

(HANOI, continued)

```
40 X = 1:Y = 2:Z = 3: GOSUB 1050:
    GOSUB 1000: END
1000 TS = TS + 1:N(TS) = N:X(TS) =
    X:Y(TS) = Y:Z(TS) = Z: IF N(
    TS) THEN N = N(TS) - 1:X = X
    (TS):Y = Z(TS):Z = Y(TS): GOSUB
    1000:M1 = X(TS):M2 = Y(TS): GOSUB
    1030:N = N(TS) - 1:X = Z(TS)
    :Y = Y(TS):Z = X(TS): GOSUB
    1000
1020  HTAB 1: VTAB 2: NORMAL :TS =
    TS - 1: RETURN
1030 T = M1:M = T(T,TP(T)): GOSUB
    5000:T = M2: GOSUB 4000: IF
     PEEK ( - 16384) < 128 THEN
     RETURN
1040  POKE  - 16368,0
1045 A =  PEEK ( - 16384): IF A <
     > 160 AND A <  > 141 THEN 1
    045
1046  IF A <  > 141 THEN  POKE  -
    16368,0
1047  RETURN
1050 TS = 0: INVERSE : FOR I = 1 TO
    3: FOR J = 4 TO 22: HTAB 13 *
    I - 12: VTAB J: PRINT " ": NEXT
    J,I: NORMAL
1060 T = 1: FOR I = 10 TO 1 STEP
     - 1:C(I) = T + 2:T = T + 1:
     NEXT :T = 1: FOR M = N TO 1
     STEP  - 1: GOSUB 4000: NEXT
    : RETURN
4000 TP(T) = TP(T) + 1:T(T,TP(T))
     = M: INVERSE : HTAB 13 * T -
    12: VTAB 24 - 2 * TP(T): PRINT
     SPC( M + 1): RETURN
5000  NORMAL : HTAB 13 * T - 12: VTAB
    24 - 2 * TP(T): PRINT  SPC(
    12): HTAB 13 * T - 12: VTAB
    24 - 2 * TP(T): INVERSE : PRINT
    " ": TP(T) = TP(T) - 1: RETURN
```

CONTROL-M COLUMNIZER

GPLE's Edit Mode lets you insert carriage returns (control-M's) into print statements. Notice this program line:

```
10 HOME: PRINT "THESE": PRINT "WORDS":
   PRINT "WILL": PRINT "PRINT": PRINT "IN A":
   PRINT "COLUMN."
```

Try this (in Edit Mode):

To insert control-characters, see **control-O** (OVERRIDE).

10 HOME: PRINT "THESE[M]WORDS[M]WILL[M]PRINT[M]
IN A[M]COLUMN."

The inverse-M's are control-M's. Both program lines above will produce the same result. The second method will print and *LIST* in a column; it also takes up just a bit less memory than normal.

REM TRICKS

By injecting some backspaces (control-H's) and carriage returns (control-M's) into REM statements, you can make some nice headings for your Applesoft program listings.

```
]LIST
10  REM

PROGRAM NAME
------------

20  PRINT "START OF PROGRAM"
```

HOW IT'S DONE:

10REM[M][M]PROGRAM NAME[M]----------
20 PRINT "START OF PROGRAM"

ANOTHER EXAMPLE:

```
]LIST
------------
PROGRAM NAME
------------

20  PRINT "START OF PROGRAM"
```

Notice how the line number is hidden by hyphens. This is done with control-H's (backspaces), which back the cursor up so the hyphens write over the line number!

HOW IT'S DONE:

You must adjust the number of control-H's according to the number of digits in the line number.

10 REM[H][H][H][H][H][H][H][H][H]----------[M]PROGRAM NAME[M]----------[M]
20 PRINT "START OF PROGRAM"

NOW YOU SEE IT...

In case you want to edit a line like the previous example, but you don't know what the line number is, set SPEED=1 and LIST. A-*HA!* Set SPEED=255 to regain normal speed (ESC L does this for you).

FILENAME TRICKS (DOS 3.3 ONLY)

Without a utility like *Byte Zap*, you can't have control-characters and inverse characters in the same file name.

To put flash and inverse characters in file names, see Tip Book #2 or our Utility City disk.

You can use the REM TRICKS from the previous page in catalog *file names* too. By employing DOS's RENAME function and a little ingenuity, file name characters can "back up" over their sector numbers, or have carriage returns or control-J's inserted into them.

```
*A Ø12 FILE #1
FILE #2 <------ This file name is actually "A[HHHHHHHHH]FILE #2".
*A Ø27 FILE #3  Type control-W and then
*A Ø62 FILE #4  "RENAME FILE #3, A[HHHHHHHHH]FILE #2"
```

CONTROL-J REMS

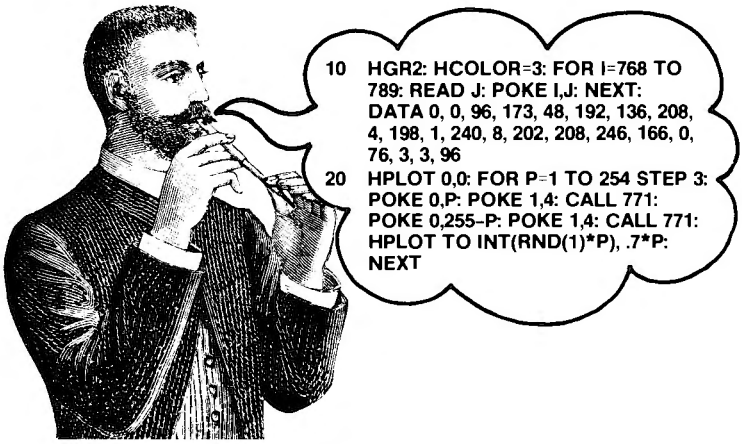
Control-J's can be put into REM statements without GPLE:

```
1Ø  REM <-------------------- Insert two control-J's here.
        SHAPE TABLE <-------- Insert one control-J here.

11  FOR S = 768 TO 773: READ A: POKE
     S,A: NEXT : DATA 1,Ø,4,Ø,5,Ø

2Ø  HGR : HOME : POKE 232,Ø: POKE
     233,3: ROT= 16:Z = 49200:T =
     2:L = 1:B = 6:M = 50:Q =  -
     16384:R =  - 16368
29  REM

        EXECUTION

3Ø  FOR X = C TO 279:S =  PEEK (Z
    ):S =  INT ( RND (1) * M): SCALE=
    S: XDRAW L AT X,8Ø - S / T: PRINT
     CHR$ (B + (S = C)): NEXT
```

APPLE METRONOME

Psst... Here's a routine that makes your $2000 Apple act like a *$29.50 metronome (wait, this metronome flashes; that's worth* an extra ten bucks for sure!). The variable TIME controls the metronome's speed; change it until you like what you see and hear.

```
10 TIME = 300: REM LOWER=FASTER
20  FOR J = 1 TO TIME: NEXT
30  VTAB 10: PRINT  SPC( 80)
40  FOR I = 1 TO 5:A =  PEEK (492
     00): NEXT I
50  HOME : INVERSE : GOTO 20
```



IIe OR NOT IIe?

From half a block away, you can't tell the difference between a new Apple IIe and an old Apple II or II+. Basically, the IIe contains the features that people were adding to their Apples in the form of add-on peripherals. It also has an improved standard (almost) typewriter-like keyboard. All ASCII characters can now be typed, including the backslash, brackets, underscore, etc. And the shift key *works* to display lower-case properly (Hooray!). The "e" comes with 64K of RAM (language card built in), and it accepts an inexpensive add-on 80-column card. Overall, the Apple IIe has far fewer parts and far better connectors.

GPLE, by the way, works well with the Apple II, II+ and IIe (and III in emulation mode), using the features of each to best advantage.

PEEKING AT THE LAST FILE

Not applicable to ProDOS.

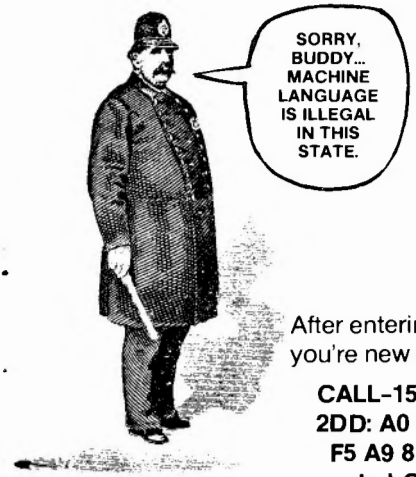
This little one-liner will print the name of the last file accessed by DOS, as well as its starting Address and Length (valid only if it was a Binary file). Problem is, you can't RUN this program, you can only GOTO it (type "GOTO 10") or use it within a program.

```
10  PRINT "LAST FILE: ";: FOR C =
    43637 TO 43667: PRINT  CHR$
    ( PEEK (C));: NEXT : PRINT "
    A"; PEEK (43634) + 256 *  PEEK
    (43635);",L"; PEEK (43616) +
    256 *  PEEK (43617)
```

If the last disk command did *not* contain a file name ("CATALOG" for example), then the first character of the file name will be erased.

The handiest way to use this program is as a GPLE Escape function. Then no GOTO is necessary! You could replace ESC W with an abbreviated form of line 10 above.

Oops, almost forgot. If you want it, here's a machine language version of the same routine, without the Address and Length:




```
02DD-   A0 00       LDY   #$00
02DF-   B9 75 AA    LDA   $AA75,Y
02E2-   20 ED FD    JSR   $FDED
02E5-   C8          INY
02E6-   C0 1E       CPY   #$1E
02E8-   90 F5       BCC   $02DF
02EA-   A9 8D       LDA   #$8D
02EC-   20 ED FD    JSR   $FDED
02EF-   60          RTS
```

After entering this code, type **CALL 733** to execute it. In case you're new at machine language, to enter this program, type:

CALL-151 (return)
2DD: A0 00 B9 75 AA 20 ED FD C8 C0 1E 90 F5 A9 8D 20 ED FD 60 (return)
control-C (return)
CALL 733 (return) to execute.

PICK YOUR VERSION

Not applicable to ProDOS.

Your GPLE Hello program could pause and ask you which version of GPLE you want loaded—.48, .LC, .DM or none.

MONITOR BLUES?

They say (and I tend to believe them) that a black monitor screen with white text is hard on your eyes, if not your brain. A better choice is one of the "green screen" monitors; some claim that amber is even better. If you do a lot of Visicalc-ing or word processing, an easy-eye monitor may be well-worth the investment. The only thing worse than a black-and-white monitor (for the eyes, that is) is a TV, particularly a color TV.

ULTRA-POP

Someone once told me (I think it was my Pop) that it is *poor programming practice* to exit a subroutine with a POP command. Instead you should always exit properly via a RETURN. And never jump out of a FOR-NEXT loop without finishing the loop first.

Well, no one around here programs by the book (if they can help it). Heck, I get so deep into dodeca-nested loops and sub-subroutines that I wouldn't know how many POPs to use if I had to! Here's a CALL (other than "He-lpp!") that will bail you out of all kinds of trouble:

CALL 54915

This big command "clears the stack" of all un-NEXTed FORs and all un-RETURNed GOSUBs. This program proves it:

```
100  REM CALL54915
120  GOSUB 500
500  GOSUB 1000
1000 N = N + 1: HTAB 1: PRINT N;
1010  GET A$
1020  GOTO 100
```

RUN it and press any key a bunch of times; you won't be able to get the counter above 12. Remove the word "REM" from Line 100 so it reads "100 CALL 54915" and RUN again! It may be poor programming practice, but it works!

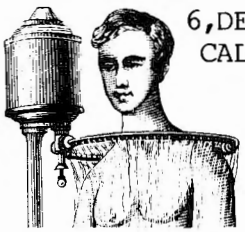
SCREEN SHIFTER

This program performs a fast, simple memory move from Applesoft, similar to the one on the Peeks & Pokes chart. RUN it and press the Left Arrow a few times until any character other than a space is in the upper left corner of the screen. Then press the RIGHT Arrow!

The Left Arrow moves the text screen memory DOWN one byte. The Right Arrow moves it UP. Due to the nature of this memory move, the upward move COPIES the value of the first byte (Vtab 1, Htab 1) all the way through the screen!

Try pressing the Right Arrow when a character is in the upper left corner of the screen.

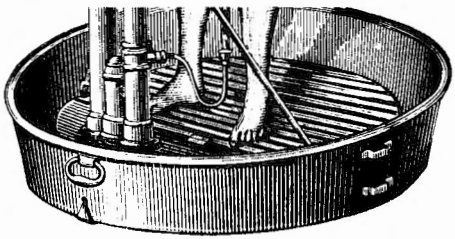
```
10  POKE 768,216: POKE 769,160: POKE
    770,0: POKE 771,76: POKE 772
    ,44: POKE 773,254: LIST
20  HTAB 1: PRINT "<- ->:";: GET
    A$:ST = 1024:EN = 2046
30 DEST = 1024 + (A$ =  CHR$ (21)
    ) - (A$ =  CHR$ (8)): GOSUB
    100: GOTO 20
100 I =  INT (ST / 256): POKE 60,
    ST - 256 * I: POKE 61,I
110 I =  INT (EN / 256): POKE 62,
    EN - 256 * I: POKE 63,I
    120 I =  INT (DEST / 256): POKE 6
    6,DEST - 256 * I: POKE 67,I:
    CALL 768: RETURN
```



SCREEN FILLER

This program works like the one above. It's task is to instantly fill the screen with any character you print in Line 10.

```
5  POKE 768,216: POKE 769,160: POKE
   770,0: POKE 771,76: POKE 772
   ,44: POKE 773,254
10  FOR X = 1 TO 12: VTAB 1: PRINT
    MID$ (":-+*@*+-:",X,1)
20  POKE 60,0: POKE 61,4: POKE 62
   ,254: POKE 63,7: POKE 66,1: POKE
   67,4: CALL 768: NEXT : RUN
```



PAGE-2 HELP PAGE

This program installs a "Help Page" on text page 2 and lets you view it whenever you want, by typing control-I (for "Info") or pressing the TAB key on your Apple IIe. It's up to you to expand this into a full-blown word processor. Call us when you're finished.

```
1  IF  PEEK (103) +  PEEK (104) *
     256 <  > 3073 THEN  POKE 103
     ,1: POKE 104,12: POKE 3072,0
     : PRINT  CHR$ (4)"RUN THIS P
     ROGRAM"
10 X$ = "TAB": IF  PEEK ( - 637) <
     255 THEN X$ = "CTRL-I"
15  TEXT : HOME
20  PRINT "HELP PAGE": PRINT : PRINT
     X$;" = HELP": PRINT "CONTROL
     -Q = QUIT": FOR I = 3 TO 9: PRINT
     "COMMAND #";I;" =": NEXT : TEXT
     : PRINT "<ANY KEY> TO CONTIN
     UE";: REM  PRINT HELP PAGE
50  POKE 768,216: POKE 769,160: POKE
     770,0: POKE 771,76: POKE 772
     ,44: POKE 773,254: POKE 60,0
     : POKE 61,4: POKE 62,255: POKE
     63,7: POKE 66,0: POKE 67,8: CALL
     768: REM MOVE PG.1 TO PG.2
80  HOME : PRINT "TYPE ON THIS PA
     GE ("X$" FOR INFO)": PRINT
100  GET A$: IF A$ =  CHR$ (9) THEN
      GOSUB 2000: GOTO 100
105  IF A$ =  CHR$ (17) THEN  END
     : REM QUIT IF CONTROL-Q
110  PRINT A$;: GOTO 100
2000  POKE  - 16299,0: GET A$: POKE
      - 16300,0: RETURN
```

WARNING: SAVE THIS PROGRAM BEFORE YOU RUN— Use your program name in Line 1. These commands move the program above text page 2.

VERTI-CALC

Actually, this program is MUCH too valuable to be given away like this, but here goes, anyway:

```
2  TEXT : HOME : ONERR  GOTO 15
10  FOR I = 1 TO 20: POKE 32,2 *
     I - 1: POKE 33,1: HOME : READ
     A$: PRINT A$: NEXT I: TEXT :
      VTAB 23: END
15  POKE 216,0: RESTORE : ONERR  GOTO
     15
16  RESUME
20  DATA HAVE YOU EVER NEEDED,TO
     PRINT YOUR TEXT IN,VERTICAL
     COLUMNS,INSTEAD OF,HORIZONTA
     L ONES?,NO?,ME NEITHER!
```

READ THIS IMMEDIATELY

You may have noticed that you can't normally do an immediate mode READ command. ("Immediate" means without a line number.) That's only because DOS thinks you mean "READ" as in "Read a text file", instead of "READ" as in "Read Data". To fool DOS, type "**:READ**" (notice the colon) instead of "READ". DOS commands are only recognized at the far left of the screen. The colon could be anything legal, like a PRINT command— anything to separate READ from the left margin. For a quick test, type a program line:

```
10 DATA 1,2,3,4,5,6,7,8,9
```

Now type ":READ A: PRINT A" a few times. It works!

AND FURTHERMOIRE...

In Tip Book #5 we learned how to use black and white lines to get color on the screen. Not to be outdone, Tip Book #7 shows you how to make curves out of straight lines:

```
20  HGR : HOME : FOR C = 1 TO 6: VTAB
     22: HTAB 18: PRINT "COLOR =
     ";C: HCOLOR= C: FOR X = 1 TO
     93: HPLOT 2 * X + 50,0 TO 3 *
     X,191: NEXT : NEXT
```

JUST ANOTHER DICE PROGRAM

This program demonstrates two things—

1. Most dice, even computer dice, are normal (REM: Some dice *players* are not).

2. You can make a nice bar graph on the text screen (REM: Who needs graphics?).

This program, in the long RUN, should prove that lucky **7** comes up more than any other number. Line 20 rolls the dice and Line 30 prints the graph. First one to the right side of the screen wins!

```
10  DIM N(40): TEXT : HOME : NORMAL
    : FOR V = 2 TO 12: VTAB V *
    2 - 3: HTAB 1 + (V < 10): PRINT
    V: NEXT
20 D1 =  INT ( RND (1) * 6) + 1:D
    2 =  INT ( RND (1) * 6) + 1:
    D = D1 + D2:N(D) = N(D) + 1
30  VTAB D * 2 - 3: HTAB 3: INVERSE
    : PRINT  SPC( N(D)): IF  PEEK
    (36) THEN 20
40  NORMAL : FOR V = 2 TO 12: VTAB
    V * 2 - 3: HTAB 3: PRINT ":"
    ;N(V); SPC( N(V) < 10): NEXT
    : PRINT  CHR$ (7)
```

SEE YA LATER TRUNCATER

If you want to delete some program lines numbered greater than 63999, add this line (can be any number you want) and GOTO it.

```
10  REM THIS LINE WILL STAY.
20  REM SO WILL THIS ONE.
5000 X =  PEEK (121) +  PEEK (122
    ) * 256: POKE X + 1,0: POKE
    X + 2,0:X = X + 3: POKE 175,
    X -  INT (X / 256) * 256: POKE
    176, INT (X / 256): END : REM
    ADD THIS LINE & GOTO IT
6000  REM THIS LINE DISAPPEARS.
65535  REM SO WILL THIS ONE.
```

NO-WORD-BRE AK TIP

Printing text on the 40-column screen without breaking words at the right margin isn't always easy, even with GPLE.

The subroutine at Line 1000 below will do the dirty work for you, breaking words only at spaces or hyphens. If you don't like what you see, you can go back and insert a hyphen or two with GPLE and then re-RUN. This program also converts lower case to upper on non-IIe's.

```
100 E2 = 0: IF  PEEK ( - 637) <  >
    223 THEN E2 = 1
200 A$ = "User-friendly software
    never tells the user that he
     is a klutz for typing an in
    appropriate answer to a ques
    tion. Instead, a program sho
    uld be polite; a slight rap
    on the user's knuckles will
    suffice.": GOSUB 1000: END
1000 WD = 39: REM WIDTH MINUS 1
1002 HT = 1: REM LEFT MARGIN
1005  HTAB HT: IF  LEN (A$) < WD THEN
     FOR I = 1 TO  LEN (A$):J =
     ASC ( MID$ (A$,I,1)): PRINT
     CHR$ (J - 32 * (J > 95 AND
    E2 = 0));: NEXT : RETURN
1010  FOR LTR = WD TO 1 STEP  - 1
    :X$ =  MID$ (A$,LTR,1): IF X
    $ <  >  CHR$ (32) AND X$ <  >
    "-" THEN 1050
1015  IF E2 = 1 THEN  PRINT  LEFT$
    (A$,LTR - (X$ =  CHR$ (32)))
    : GOTO 1040
1020  FOR I = 1 TO LTR:J =  ASC (
     MID$ (A$,I,1)): PRINT  CHR$
    (J - 32 * (J > 95));: NEXT :
     PRINT
1040 A$ =  RIGHT$ (A$, LEN (A$) -
    LTR): GOTO 1000
1050  NEXT : PRINT : PRINT  CHR$
    (7);"WORD TOO LONG";: END
```

CARD SHUFFLER

This program shuffles cards. It doesn't play cards, it just shuffles them.

```
10  TEXT : HOME : NORMAL : PRINT
     "SHUFFLING..."
20  DIM CARD$(52),CHECK(52)
30 SUIT$ = "CDHS"
40 N$ = "A23456789TJQK"
50  FOR S = 0 TO 3: FOR C = 1 TO
     13:CARD$(C + S * 13) =  MID$
     (N$,C,1) +  MID$ (SUIT$,S +
     1,1): NEXT : NEXT
60  FOR X = 1 TO 52
70 XX = X - 1: VTAB 3 + XX -  INT
     (XX / 13) * 13: HTAB 1 + 10 *
      INT (XX / 13)
80  PRINT  SPC( X < 10);X;":";
90 N =  INT ( RND (1) * 52) + 1
100 S =  PEEK (49200)
110  IF CHECK(N) THEN 90
120 CHECK(N) = 1
130  INVERSE : PRINT CARD$(N);
140  NORMAL : PRINT  SPC( 5)
150  NEXT X: VTAB 1: PRINT "SHUFF
     LED";: CALL  - 868: VTAB 20:
      END
```

HERE'S A GOOD ONE FOR YOU—
ALL THINGS CONSIDERED EQUAL,
WHICH WILL WEAR OUT FIRST, AN APPLE
THAT IS TURNED ON BUT DOING
NOTHING, OR AN APPLE THAT IS COUNTING IN AN INFINITE LOOP?
(ANSWER ON PAGE 255.)

NIGHT DRIVER

This is a neat program that lets you follow a Lo-Res car down a long road to nowhere.

```
10  GR : HOME : NORMAL :A = 1:B =
     21:D = 22:L = 20:R = 39:S =
     40:U = 3:V = 15:A = 1:M = 10
     000:Z = 0:E = 36:F = 37
20  COLOR= 2: FOR Y = 28 TO 34: HLIN
     23,27 AT Y: NEXT : COLOR= V:
      HLIN D,28 AT 35: COLOR= 5: PLOT
     25,33: REM DRAW.CAR
30  FOR C = Z TO V STEP U: IF C =
     Z OR C = 6 THEN  COLOR= V *
     (C = Z): FOR I = 23 TO 26: HLIN
     L,I AT I + A: NEXT I: HLIN 2
     4,26 AT 30: HLIN 24,26 AT 29
     : HLIN B,D AT 28: PLOT D,29:
      REM HEADLIGHTS.ON/OFF
40  FOR X = L TO R STEP A + (C >
     U): COLOR= C: HLIN S - X,X AT
     R - X: VLIN S - X,X AT X: VLIN
     R - X,X AT S - X: REM BKGRD
50  COLOR= U:M = M + A: HTAB 24: PRINT
     M;: PLOT 27,41: REM MILEAGE
60  COLOR= X: VLIN E,F AT 23: VLIN
     E,F AT 27: REM TIRES
70  COLOR= V: VLIN D,R AT L: COLOR=
     Z: FOR J = B + X -  INT (X /
     U) * U TO R STEP U: PLOT L,J
     : REM WHITE.LINE
80  NEXT : NEXT : NEXT : GOTO 30
```

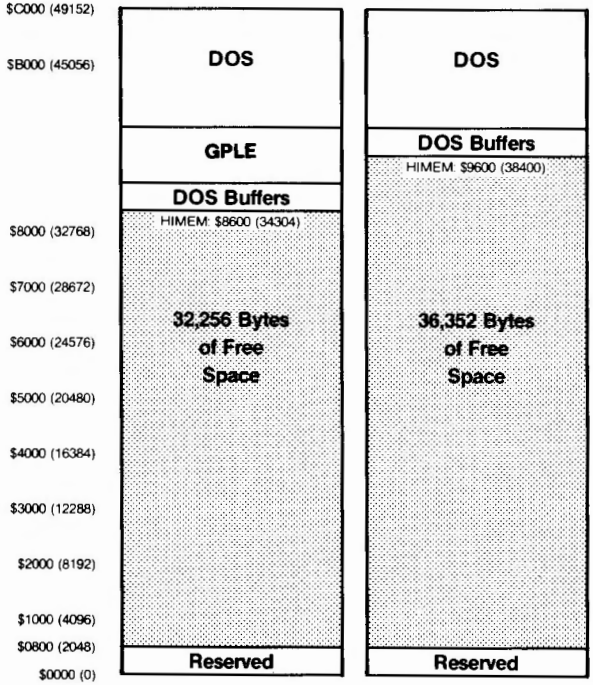
GOSUB QUOTE CONVERSION

The string will be printed on the screen, and then the carets will be converted to quote marks.

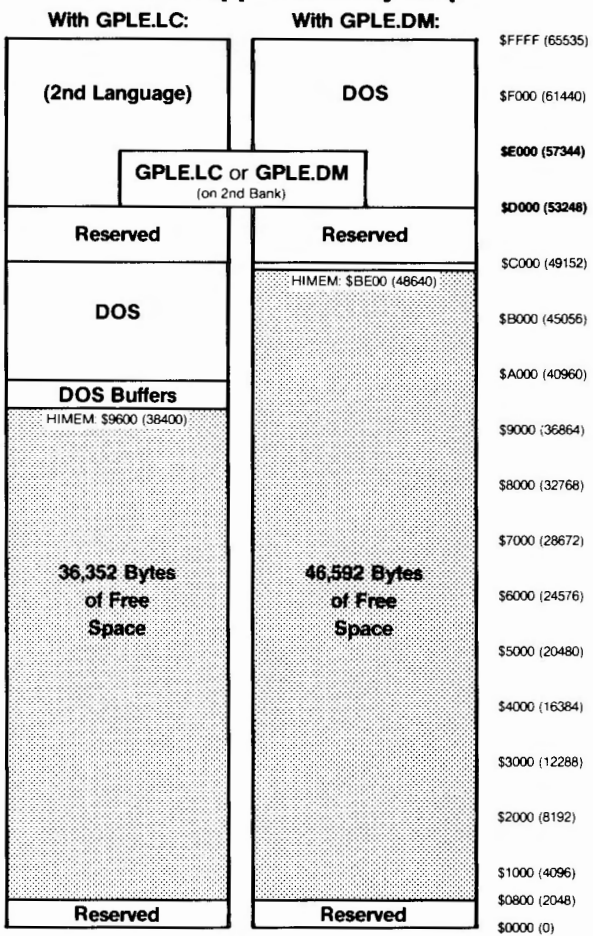
```
10  PRINT "PUT THIS IN ^QUOTES^ F
     OR ME.": GOSUB 1000: END
1000  VTAB  PEEK (37):L =  PEEK (
     40) +  PEEK (41) * 256: FOR
     I = L TO L + 39: IF  PEEK (I
     ) = 222 THEN  POKE I,162
1001  NEXT : PRINT : RETURN
```

GPLE Index

ProDOS 64K and DOS 3.3 48K Apple Memory Map



DOS 3.3 64K Apple Memory Map



Other Beagle Bros Apple Software

CHECK OUR APPLE MAGAZINE ADS TO SEE WHAT'S *NEW.*

■ *GRAPHICS* ■

❑ **ALPHA PLOT** (II+, IIe, IIc)† $39.50
Normal hi-res (6 colors, 280x192 pixels) drawing and typing on both hi-res pages. Compress pictures to 1/3 disk space.

❑ **APPLE MECHANIC** (II+, IIe, IIc)† 29.50
Create hi-res shapes for animation with Applesoft's DRAW & XDRAW commands. Put fancy hi-res type in your programs. List & learn demo programs teach you hi-res programming.

❑ **APPLE MECHANIC TYPEFACES**† 20.00
26 new editable fonts to be used with Apple Mechanic.

❑ **BEAGLE GRAPHICS** (IIc or 128K IIe)★ 59.95
Double hi-res drawing (16 colors, 560x192 pixels) and typing in many typestyles (all editable). Color fill, cut & paste, 200+ color mixes. 33 new commands for using double-res in your programs. Convert normal hi-res pictures and programs to double hi-res, compress pix to 1/3 disk space...

❑ **FLEX TYPE** (II+, IIe, IIc)† 29.50
Variable-width text (wide, normal, condensed) controllable with normal Applesoft commands. No 80-column card reqd.

❑ **FRAME-UP** (II+, IIe, IIc)† 29.50
Make Apple "slide shows". Keyboard controlled or unattended, using your existing hi-res, lo-res and text screens.

❑ **TRIPLE-DUMP** (II+, IIe, IIc)★ 39.95
Transfer any image, including double hi-res, to your dot matrix printer. Make Giant (8" high characters) Banners too.

■ *ALL-PURPOSE* ■

❑ **DISKQUIK** (IIc or 128K IIe)† $29.50
Acts like half a disk drive in slot 3. Silent and fast as a hard disk. Load/save files in memory with normal commands.

❑ **FATCAT** (II+, IIe, IIc)★ 34.95
Reads all of your DOS 3.3 and ProDOS file names into one or more Master Catalogs for sorting, searching and printing. Alphabetize file names on disks. Compare any two files.

❑ **PRONTO-DOS** (II+, IIe, IIc)† 29.50
Triples the speed of loading and saving. New TYPE command displays text file contents. Move DOS for extra 10K.

■ *PROGRAMMING* ■

❑ **BEAGLE BASIC** (IIe, 64K II+)† $34.95
Puts Applesoft in RAM so you can change it and add enhancements—new commands like if-then-ELSE, SWAP variables, GOTO/GOSUB-a-variable, TONE, HSCRN, etc.

❑ **DOS BOSS** (II+, IIe, IIc)† 24.00
Reword DOS 3.3 commands. Change "Catalog" to "Cat", "Syntax Error" to "Oops" or *anything.* Includes many meaty tips for altering DOS, including program "save-protection".

❑ **DOUBLE-TAKE** (II+, IIe, IIc)★ 34.95
2-way scroll for Listings & Catalogs. Better List-format, fast variable+line number display, better renumber/append, auto line-numbering, instant hex/dec converter and more.

❑ **GPLE** (II+, IIe, IIc)★ 49.95
Edit Applesoft without cursor-tracing. Features insert & delete and fast search & replace. Make all keys be "function keys" to type anything you like (ESC-1 catalogs disk, etc.). Move DOS 3.3 out of main memory to add 10K of space.

❑ **SILICON SALAD** (II+, IIe, IIc)† 24.95
Over 100 utilities and tricks— hi-res program splitter, DOS killer, disk scanner, hi-res text imprinter, 2-track catalog...

❑ **TIP DISK #1** (II+, IIe, IIc)† 20.00
100 tips on disk from Tip Books 1-4. Fascinating Apple programming techniques. Includes Apple Command Chart.

❑ **UTILITY CITY** (II+, IIe, IIc)† 29.50
21 utilities— List-formatter puts each statement on a new line, multi-column catalogs, invisible/trick file names, etc.

■ *GAMES* ■

❑ **BEAGLE BAG** (II+, IIe, IIc) $29.50
12 games on one disk. Voted to 1983's MOST POPULAR list in *Softalk* poll. The best Apple game bargain on the market.

† Supports DOS 3.3 only
★ Supports DOS 3.3 *and* ProDOS™

ISBN 0-917085-05-1

Published by BEAGLE BROS MICRO SOFTWARE, INC.
3990 Old Town Avenue, San Diego, California 92110