ANOTHER USE FOR APPENDER

&STORE can also be used to temporarily hide an Applesoft program while you run another program. Type “NEW” to get rid of the test program, then “&RECALL” to get your original back.

DUMP.40 and DUMP.80

Install only once: Type “-DUMP.40” for 40 columns or “-DUMP.80” for 80 columns. New command: &POP (stands for “Print On Paper”) Function: Prints the text screen on your printer.

Comments: &POP is fast because it doesn’t waste time printing blank spaces at the end of a line.

If you have an Apple IIc or Apple Ile with an 80-column card, you can use the new DUMP command instead (see page 48).

```
    PROGRAM EXAMPLE:
    10 REM DUMP.40 MUST BE IN MEMORY
    15 PRINT CHR$(21): REM 40-COL SWITCH
    20 FOR Y=1 TO 24: FOR X=1 TO 40: IF
X<40 OR Y<24 THEN VTAB Y: HTAB X:
```

PRINT CHR$(X+32+Y);: NEXT: NEXT

```
    30 VTAB 1: REM ANY PROGRAM
    40 &POP: REM DUMP SCREEN
    50 REM CONTINUE WITH PROGRAM
```

80-COLUMN PRINTER PROBLEMS?

DUMP.80 and the DUMP command (page 48) send a carriage return to the printer after every 80 characters. If your printer does this automatically and you are getting some kind of double-spacing, here’s the solution:

1. Type “BLOAD DUMP.80” (or BLOAD DUMP).
2. Type “POKE 16385,0”.
3. Type “BSAVE DUMP.80” (or BSAVE DUMP).

Note: DUMP.80 Requires 80 columns!

INPUT-40 and INPUT.80

Install only once: Type “-INPUT.40” for 40 column input or “INPUT.80” for 80 columns. New command: &INPUT Features:

- All visible characters are acceptable as input, including commas and colons.
- Control-characters are ignored.
- Maximum string length is programmer-definable. The standard length is 20 (if someone hasn’t changed it already).
- Pressing Esc returns a null string (as in A$=””) regardless of cursor position.
- Pressing Return accepts the input as it appears, regardless of cursor position.
- Pressing Control-Y clears from the cursor to the end of the line.
- Pressing Delete deletes the character to the left of the cursor and closes the gap created. The left and right arrow keys are used normally. (If you don’t have a Delete key, see next page.)

Limitations:

- String variables only (For numeric input, use the VAL function as shown in the examples).
- Screen input only (&INPUT cannot be used for getting data from a disk file).
- Insert is not available. Check the NOTES program; maybe we’ve added it by now.

Example:

10 REM INPUT.40 MUST BE IN MEMORY

15 &INPUT “TYPE YOUR NAME HERE (LAST, FIRST): “;A$ 20 PRINT “HELLO, “;A$;”!”

This example works whether or not the user typed a comma between his last and first names. Normal INPUT would have cut the name short at a comma.

INPUT.SETUP

This program lets you change two key features of the INPUT.40 and

INPUT.80 programs:

1. The maximum string length allowed. For example, you might want to limit someone to typing only a 5-character word.
2. The key to be used as the Delete key. For example, if your Apple doesn't have a Delete key, you might want to use control-D instead. Type "-INPUT.SETUP" and follow the instructions on the screen. When you are finished, a new version of INPUT.40 or INPUT.80 will be saved on disk.

POKING THE MAXIMUM STRING LENGTH

Another way is to change the maximum string length is to do it with a Poke. For example, to change the maximum allowable &INPUT string length to 5:

1. Install -INPUT.80 or -INPUT.40.
2. Immediately let LOC=PEEK(1015)*256+7
3. Let MAX=5 (for a maximum length of 5)
4. POKE LOC, MAX PROGRAM EXAMPLE:

10 REM INPUT.80 MUST BE IN MEMORY

```
20 REM AND LOC MUST HAVE BEEN SET
30 POKE LOC,10: REM LENGTH=10
40 &INPUT ‘TYPE A WORD: “;X$
60 POKE LOC,2: REM NEW MAX LENGTH
70 &INPUT ‘ENTER 2-DIGIT NO: “;TD$
80 TD = VAL(TD$): REM CONVERT STRING
TO NUMERIC VARIABLE
90 PRINT X$, TD * 2: GOTO 30
```

REMOVE and REM.OVE.128
Install only once: Type "-REM.OVE".
If you have 128K, type ‘REM.OVE.128”
New command: &REM

Function: Removes all REMarks from the Applesoft program in memory. Rem's are helpful when programming, but when a program is executing, they take up room and slow things down.

REM.OVE

This program simply deletes all Rem's from the program in memory. Before you make any changes to your program, you will probably want to load your Rem'd version back in from disk. To use it:

1. Type "-REM.OVE"
2. Load an Applesoft program with Rem's.
3. Type "&REM" to remove the Rem's.
4. Run the program without Rem's.
5. Re-load the Rem'd version from disk.
6. Make changes to it.
7. Save the program back to disk.
8. To test it again, go back to step 3.

REM.OVE.128

This program also deletes Rem's, but it first saves a copy of the

Rem'd version of your program, named ‘SOURCE” on the ProDOS RAM disk:

Slot 3, Drive 2.

From there, you can more-quickly retrieve your Rem’d version. To use it:

1. Type “-REM.OVE.128”
2. Load an Applesoft program with Rem’s.
3. Type “&REM” to save the program on the RAM disk and remove the Rem’s from main memory.
4. Run the program without Rem’s.
5. Type ‘LOAD/RAM/SOURCE”.
6. Make changes to the program.
7. To test it again, go back to step 3.
8. Save the program on disk.

Warning: Never use REMOVE from within a program; weird things might happen!

As Rem’s are removed, you will see an “*” for each full program line eliminated, and a “:” for each Rem dropped from the end of a line. When done, the total bytes saved will be printed on the screen.

GOTO WHERE?

If you’re in the habit of issuing GOTO’s or GOSUB’s to a Rem -only line, as in the sample below, insert a colon before the Rem before using REMOVE.

```
1000 GOTO 2000
2000 :REM NOTICE THE COLON.
2010 PRINT ‘HELLO”
```

After Rem’s are removed, the line wi ll be left with only a colon (“:”) following the line number:

1000 GOTO 2000

2000 :

2010 PRINT “HELLO”

Without the colon, the entire line would have been removed and you would get an ?Undef’d Statement Error when you ran the program. Actually, it would be more efficient of you to just GOTO the next line (you would save 6 whole bytes).

AND NOW A WORD FROM OUR SPONSOR

‘For the maximum in program compacting, including erased Rem’s, shortened variable names and automatic combining of program lines, use Beagle’s D CODE disk. See our catalog for details.”

SUPER.POKE

Install only once: Type “-SUPER.POKE”.
New command: &POKE

Function: Allows an easier way to poke a range of memory or a string of numbers from Applesoft. A demo program resides semi-hidden on the Big U disk.

Type “-COPYRIGHT. 1985” to see it.

FILL A RANGE OF MEMORY

Syntax: &POKE start: end, value

‘start”, ‘end” and ‘value” can be literal numbers or variables.

10 REM EXAMPLE #1: FILL A LINE

```
15 FOR X=0 TO 255
20 &POKE 1024:1063,X: NEXT
10 HGR: REM EXAMPLE #2: FILL HI-RES
```

```
          20 VTAB22: INPUT “NUMBER (0-255):”;X
          30 &POKE 8192:16384,X: GOTO 20
          POKE A STRING OF NUMBERS
          Syntax: &POKE start, value1, value2, etc.
```

“value 1” and “value2” can be literal numbers or variables. This routine is useful for replacing lengthy For -Next loops which read Data statements and Poke in values.

Example: &POKE 1024,193,194,195 prints “ABC” in on the screen. 1024 is the upper left of the screen. 193, 194 and 195 is ASCII for “ABC”.

And, adapted from Tip Book #7:

10 HGR2: HCOLOR=3: &POKE768,0,0,96,

```
          173,48,192,136,208,4,198,1,240,8,
          202,208,246,166,0,76,3,3,96
20 HPLOT 0,0: FOR P=1 TO 254 STEP 3:

          &POKE O,P,4: CALL 771: HPLOT TO
          INT (RND(1)*P),.7*P: NEXT
```

the prodos commands and the rest.

Big U’s COMMAND Programs

Each Big U Command .program gives you at least one new ProDOS command. All commands are loaded or “installed” by typing a hyphen (“-“) followed by the file name. The new commands may then be used just like regular ProDOS commands—from immediate mode (the keyboard) or from deferred mode (within a program) preceded by PRINT CHR$(4).

WARNING: LOAD ONLY ONCE!

Never load (or “install”) the same command more than once. Not only will you waste memory, but ProDOS might hang up in an endless loop which can only be broken by rebooting. Type “-BASIC.SYSTEM” to start over and get rid of extra commands.

COMMAND SYNTAX

Items in [square brackets] are optional. Items in UPPER CASE must be typed as shown. Items in lower case are parameters supplied by you.

ANYCAT

Install only once: Type “-ANYCAT”
New command: ANYCAT
Function: Catalogs DOS 3.3 disks under ProDOS control.
Syntax: ANYCAT [,S slot] [,D drive]

Comments: DOS 3.3 deleted files will be displayed as file type “D”. An inverse character will appear to the far right of each deleted file name. When the catalog pauses, press Esc to exit, or any other key to see more files.
If a ProDOS disk is present, the normal “CAT” command will take over.

If a prefix is set, it will be used instead of the slot and drive. Example: Type “ANYCAT,S6,D2” to catalog the DOS 3.3 disk in slot 6, drive 2. If a ProDOS disk is there, a “CAT” command will be executed.

COPY and COPY.1

Install only once: Type “-COPY” or “-COPY.1” Use COPY.] if you only have one drive. New command: COPY
Function: Copies files from disk to disk.
Syntax: COPY pathnamel, pathname2

Comments: “pathnamel” is the file to be copied, and “pathname2” is the file to be created. “pathname2” cannot be an existing file.

COPY. 1’s COPY requires 2 full pathnames (both starting with “/”).
COPY’s COPY, however, will substitute the current prefix if you don’t

supply it (see last example on this page).
Directory (DIR) files cannot be copied.
You cannot COPY between disks with the same name. You may, however, quickly rename disks by typing a command like “RENAME /DISKA,/DISKB”.
Examples:
Type “COPY /BIG.U/COPY, /MYDISK/COPY” to copy COPY onto your disk.
Type “COPY FILE, FILE.A” to make a backup copy of a file on the same disk.
In programs, you may use strings as pathnames:

```
10 PRINT CHR$(4);”COPY”;A$:”,”;B$
```

With COPY (not COPY.1), you may take advantage of the current prefix.

For example, there are two ways to copy TEST from /DISKA to /DISKB:

1. Type “COPY /DISKA/TEST, /DISKB/TEST”.
2. Or type “PREFIX, D2” (DISKB’s location),

Then type “COPY /DISKA/TEST, TEST”.

DUMP

Compatibility: See Comments below.
Install only once: Type “-DUMP”
New command: DUMP
Function: Type “DUMP” to print the entire text screen to a slot 1 printer. DUMP determines whether 40 or 80 columns are being viewed, and prints accordingly. Press Esc to stop early.

Comments: DUMP is compatible with Apple IIc’s and most 80-column IIe’s. Use the DUMP.40 program (page 39) if you don’t have an 80-column display, or use DUMP.80 if your 80-column card doesn’t work with the DUMP command. See page 39 if you get blank lines in your 80-column dumps. DUMP won’t let you crop or print graphics. Use Beagle Bros’ Triple -Dump disk instead.

Example: Type “CAT” in 40 or 80 columns. Then type “DUMP” to send the screen to the printer.

EST (ProDos 1.1.1 only)
Install only once: Type “-EST”.
New commands: ERASE, SPACE and TIME
Type “ERASE” to effectively erase the Applesoft program in memory (RENEW should restore it), reset all Applesoft pointers and the ProDOS bit map, turn off 80 columns and execute NORMAL, TEXT and SPEED=255. ProDOS commands added after EST was installed will be lost when you type “ERASE”. ERASE is similar to 3.3’s FP command. Type “SPACE” to show the disk volume name and free space for the disk in the last drive accessed. Do not use this command from inside a program.

Type “TIME” to display the time and date stored in memory. This command cannot change the values; to do that, use the DATE.SET program (page 63). If your Apple has a clock, you can use TIME in a program like this:

```
10 VTAB 1: HTAB 24: PRINT CHR$(4); “TIME”: GOTO 10
```

HEX

Install only once: Type “-HEX”.

New commands and functions:

XC: Hex/dec/bin converter (page 50)
XD: Memory disassembler (page 50)
XP: Register and flag displayer (page 51)
XS: Hex/Ascii memory scanner (page 51)
Comments: Any leading “X” is considered to be a command when entered from the keyboard, so type a colon before defining variables that start with X (not necessary from within programs). For example, with HEX installed, if you type “X=5”, you’ll get a ProDOS ‘Syntax Error” message. Typing “:X=5” will works normally.

HEX.H (ProDos 1.1.1 only)

HEX.H is the same as HEX (above), but it “hides” inside BASIC.SYSTEM from $BB4C.BC79 so that no programming memory is used. HEX.H may be added to BASIC.SYSTEM so that it is installed when you boot, at no cost in memory:

1. Boot a backup copy of the Big U disk.
2. Type “-HEX.H” to install HEX.H.
3. Type “CALL -151” to enter the monitor.
4. Type “BLOAD BASIC.SYSTEM,TSYS,A$2000”.
5. Type “454C<BB4C.BC79M” to move HEX.H into BASIC.SYSTEM.
6. Type “4707: 50 BB” to point the external command to HEX.H.
6. Type “BSAVE BASIC.SYSTEM,TSYS,A$2000”.
7. Reboot and try some HEX commands.

Negative numbers will not be shown in the result when XC (next page) is used with HEX.H, although negative numbers may still be entered.

XC (HEX command 1 of 4)

Function: Convert a number to hexadecimal, binary, decimal and negative decimal. Syntax: XC [,E number]

Comments: “number” is any decimal number from 0 to 65535, any hex ‘$” number from $0 to $FFFF, or any negative decimal number from -1 to -32768. Typing “XC” without a number will repeat the last number converted. HEX.H’s XC will not convert to negative decimal. Example: Type “XC,E -151” to convert -151 to hex, binary, decimal and negative decimal. The same result would be achieved by typing “XC,E$FF69” or “XC,E65385”.

XQ (HEX command 2 of 4)

Function: Disassemble memory. This is an enhanced version of the monitor List command. Syntax: XD [,E address] [,L lines] Comments: “address” is the decimal or hex location where you want to start disassembly. ‘lines” determines how many lines (1 -255) will be disassembled. Because a line can contain 1 to 3 three bytes, the actual amount of memory disassembled will vary. If you specify a large number of lines, you can pause the listing by pressing Control-S. Any key continues. Typing “XD” will repeat the last disassembly. Example: Type “XD,E$FBD9,L11” to see the speaker beep code. Continue viewing 11 -line segments of memory by typing “XD,E address” only. Free Tip: To print a machine language listing, preview it on the screen using the XD command. When you see what you like, type “PR#1” and then “XD” to repeat the listing on your printer.

XP (HEX command 3 of 4)

Function: Display the 6502 registers and flags. Syntax: XP . Example:

Type “XP”. The screen will show:

```
A=04 X=8C Y=00 P=2B S=98
          NV-BDIZC
          P=00101011
```

Comments: AA and Y are registers; P is the processor status register and S is the stack pointer. The processor is repeated in binary. Its flags from left (bit 7) to right (bit 0) are: N (negative), V (overflow), “-“(unused), B (break), D (decimal), I (interrupt), Z (zero), and C (carry). Bits are considered set if 1, or clear if zero. For example, a BCS (Branch on Carry Set) instruction would branch if a “1” appears under the “C”.

The register values are taken from the storage area at $45.$49. To see what flags are set when P=9, just type “POKE 72,9” followed by “XP”; or type “CALL -151” followed by “48:9” and “XP”.

XS (HEX command 4 of 4)

Function: Scan memory in hex and ASCII. Syntax: XS [,E address] [,L lines]

Comments: “address” is the starting address of memory, and ‘lines” is the number of lines displayed (independent of the bytes per line). XS displays 8 bytes per line on a 40-column screen, and 16 bytes per line in 80 columns.

“XS” with no parameters repeats the last scan.

Example: Type “XS,E$B878,L20” to see the ProDOS command table. Typing “XS,E address” will then continue to show 20 -line segments of memory.

MON (ProDos 1.1.1 only)
Install only once: Tpe “-MON”
New commands: MON and NOMON
Function: Turn ProDOS monitoring on and off
(similar to 3.3’s MON and NOMON commands).

The MON command will echo the following ProDOS commands to the screen:

CREATE, OPEN, READ, WRITE, CLOSE, FLUSH and DELETE. For text files, data being input after a READ or printed after a WRITE will also be shown on the screen.

CREATE, DELETE and OPEN will show the pathnames involved. The others will give the pathnames for the first two open files, and open files from 3 to 8 will be referenced by number.

Syntax: MON [,R rate] Syntax: NOMON "rate" is either 0 (zero) or 1. Use zero for continuous operation and 1 for single -step (press a key to continue). If NOMON has been used, typing "MON" will return to monitoring with th e same "rate" setting used previously.

Examples:

Type "MON". Now type an illegal command like "CATS" to see ProDOS's "Close All", "Flush All" and "Syntax Error" messages. Type "MON,R1" followed by "CAT" to single -step through a catalog listing. Type "MON,RO" to return to continuous MONitored operation. . Type "NOMON" to turn off monitoring.

IMPORTANT: Turn MON off, reboot or type "BYE" before using a SYS program like Apple's FILER.

Why? Because MON intercepts MLI calls at $BFOO in the ProDOS global page. The original addresses are reset by NOMON, and by MON when "BYE" is used.

ONLINE

Install only once: Type "-ONLINE"
New command: ONLINE
Function: Lists the current ProDOS volume names for each of your drives. If slot and drive are specified, only that drive is checked. Syntax: ONLINE [,S slot] [,D drive]

Examples:

Type "ONLINE" to display volume names for all drives. Or type "ONLINE,S6,D2" to display the volume name for slot 6, drive 2 only.

SEE

Install only once: Type "-SEE"
New command: SEE
Function: Lets you view any program on a disk without loading it. SEE will print text (TXT) files as well as Applesoft program (BAS) files. Other file types may be dumped too, but not as effectively. Syntax: SEE pathname [,S slot] [,D drive]

Comments: "pathname" is any filename. Press any key to pause a listing, followed by Esc to exit, or any other key to continue. Examples: Type "SEE/BIG.U/STARTUP" to display Big U's Applesoft (BAS) startup program.

You can use SEE in a program (Esc to quit):

10 PRINT CHR$(4)"SEE PRODOS,S6,D1"

PRODOS is a binary (SYS) file. Text (TXT) files look better, but there are none on the Big U disk.

SHOW

Install only once: Type "-SHOW"
New command: SHOW
Function: Load a normal hi-res image from disk and/or reveal the hi-res screen without clearing it.

For double hi-res, use SHOW.WIPE (next page).
Syntax: SHOW [pathname] [,A page] [,F format] [,S slot] [,D drive]
"pathname" is the name of a picture to see.
"page" is a number: 1=page 1, 2=page 2
"format" is a number: 0=full graphics, 1=mixed graphics and text. If no format value is given, 0 is assumed.
Examples:

Run the file called "RANDY.BRANDT" to see a demo of how SHOW can be used.

Type "SHOW,A1" to reveal all of hi -res page 1 without erasing it (works like HGR, but doesn't erase and doesn't include four text lines).

Type "SHOW" to reveal the last screen accessed (full graphics, no text).

Type "SHOW PIC,A2" to load a hi -res picture named "PIC" to page 2 and reveal it.

SHOW.WIPE (128k required)

Install only once: Type “-SHOW.WIPE”
New commands: SHOW and WIPE

SHOW (SHOW.WIPE command 1 of 2)

Function: Loads a single or double hi-res image from disk and/or reveals the normal or double hi-res screen without clearing. SHOW works only with Beagle Graphics’ two -file picture format. Syntax: Same as on previous page, except a third “page” parameter is allowed: 1=hi-res page 1, 2=hi-res page 2, 3=double hi-res. For double hi-res picture names, SHOW expects the second file to end in “.AUX”.

Examples: Type “SHOW DOUBLE,A3” to load the files “DOUBLE” and “DOUBLE.AUX” and display them as one double hi -res picture. Or simply type “SHOW,A3” to reveal the double hi -res screen.

WIPE (SHOW.WIPE command 2 of 2)
Function: Reformats (erases) the ProDOS RAM disk.
Syntax: WIPE [,A3]
Comments: “WIPE” reformats /RAM. “WIPE,A3” does the same, but also saves hi -res page 1 in a /RAM file called “DHGR.AUX”, thus protecting double hi-res graphics memory from being overwritten. It also protects future /RAM files from being overwritten by double hi-res graphics.
Examples:

Type “WIPE” to erase all /RAM files. Type “CAT/RAM” to see the blank directory.

Load a normal hi-res picture to page 1. Now type “SHOW,A3” followed by “WIPE,A3”.

XLISTER

Install only once: Type “-XLISTER”
New command: XLIST
Function: Displays Applesoft programs in improved list format. Each statement is listed on a new line. For-Next loops are indented, and statements following IF’S are marked with a “*”. See the sample printout on the next page.

Syntax: XLIST [*] [line number or range]

Specify single lines or ranges just like in Applesoft’s LIST command. Because XLIST is a ProDOS command, it can be entered in lower case (and even be used from the monitor!).

Type “XLIST*” to list to a printer in slot 1. With printer listings, you get page breaks and page numbers. If your program starts with a REM statement, that statement will appear as a header at the top of each page. If you’re using ProDOS 1.1.1, XLISTER will automatically print the date and time in each header.

Examples: Type “XLIST -100” to display li nes 0-100 on the screen. Or type “XLIST* 10,20” to send lines 10 through 20 to your printer. In the example program below-You can type “RUN” to RUN it. You can type “RUN2’ to SAVE it. You can type “RUN1” to XLIST it to printer. Each page of the printer listing will have line 0’s REM printed on the top. To print only part of the program, add the line range to line 1.

0 GOTO 3: REM SAMPLE

```
1 PRINT CHR$(4);"XLIST*" : END
2 PRINT CHR$(4);"SAVE SAMPLE": END
3 REM PROGRAM STARTS HERE
```

Sample Program XLISTing:

10 REM CUSTOM.LIST.DEMO

```
15 HGR
:HOME
:PRINT CHR$(21)
:VTAB 22
:NORMAL
20 FOR X=0 TO 279
POKE 228,99+X/99
HPLOT 0,0 TO X,159
IF X<160 THEN GOSUB 90
* : HPLOT 0 ,0 TO  279,X
*    :GOTO 30
25 GOSUB 1
30 NEXT
     :HTAB 1
     :PRINT “DONE”
     :CALL -868
     :END
```

90 VTAB 22

```
:HTAB 1
:PRINT “DRAWING: “;X;
          :RETURN
100 :HTAB 1
          :PRINT “ALMOST DONE: “;
```

X;

```
:RETURN
```

FOR PRINTER EXPERTS ONLY

XLISTER contains eight printer enhancement bytes which may be used to turn on printer options such as compressed, bold, etc. Here’s how to use the m:

1. Type “BLOAD XLISTER”.
2. Get out your printer manual and look up the hex codes for the desired printer options.
3. Type “CALL -151” to enter the monitor.
4. Type “4599:00 00 00 00 00 00 00 00”, replacing the 00’s with the proper codes.
5. Type “CALL 16384” to install the modified XLISTER, or type “BSAVE XLISTER” to make the changes permanent.

Big U’s UTILITY Programs

BEEPERWORKS Requires Appleworks

This program is for AppleWorks users only (versions 1.1 or 1.2). Even though we don’t get paid to endor se AppleWorks, we’ll let it be known that we use it for all of our word-processing needs. Its error beep, however, can get just a little annoying, especially when you make as many errors as we do. Thanks to BEEPERWORKS, your ears can at least get a little variety.

USING BEEPERWORKS

1. Type “-BEEPERWORKS” and a familiar screen will appear. Insert a working copy of your AppleWorks Startup disk (not the Program disk), and follow the instructions on the screen. If you can use AppleWorks, you can use this program.
2. Select option 2 from the main menu. The Beep Change Menu will be displayed, with values for the beep’s pitch, duration and pause. Three sets of these values are shown; the current values, the boot-up values (stored on your disk), and AppleWorks’ original values.
3. Do some experimenting, and pick the beep you like. We like the pitch at 20, the duration at 180, and the pause at 50. Pick your own beep.
4. When you’re done, use the main menu save option to make your new beep a “permanent” part of AppleWo rks. Hats off to AppleWorks for being unprotected!

BIGLINER

BIGLINER is used to give your Applesoft programs “illegal” program lines numbered 65535. This prevents the average human from deleting or otherwise messing with those program lines, unless of course, he or she has a copy of BIGLINER! Line 65535 is commonly used to store Copyright notices in Rem statements.

For some reason, Applesoft does not allow line numbers higher than 63999. Therefore, line 65535 cannot be deleted. No Goto’s, Then’s or Gosub’ s can access it either, but the program will work normally otherwise.

USING BIGLINER

Load a backup copy of an Applesoft program and try the following steps:

1. Type “-BIGLINER”. You will see a menu offering you four choices, followed by a listing of the highest legally-numbered line (if there is one).
2. Press ‘L” to list the entire Applesoft program. Pause with Control -S, or quit with Control-C.
3. Press ‘R” to raise the listed line number to number 65535. The highest legal line listed will be replaced.
4. Press ‘L” to list the program. The last line listed will now be 65535.
5. Press ‘C” to change the first line 65535 to 63000. If line 63000 is present in the program, the next available line number will be used. BIGLINER only changes the line numbers themselves; it doesn’t check to see if the line is referenced by another statement.
6. To quit, press ‘Q”.

CAT.DATER (ProDos 1.1.1 only)

This is a small Applesoft program that makes the current ProDOS date and time appear following the directory name at the top of each catalog. If you don’t have a clock/calendar in your Apple, use the DATE.SET program (page 63) to set the date first.

USING CAT.DATER Type “-CAT.DATER”. Then type ‘CAT” or ‘CATALOG” to see the date in your catalog.

TECHNICAL NOTES: CAT.DATER is relocatable, so its address (variable AD in line 10) may be changed if you want to use page 3 of memory for other purposes. The existing setting allows it to be compatible with the CAT.FIXER (next page) and CAT.STEPPER (below).

CAT.STEPPER (ProDos 1.1.1 only)

This program modifies the catalog routine so that you can “step” through a directory one or more files at a time.

USING CAT.STEPPER

Type “-CAT.STEPPER” to install the routine. Now your catalogs will pause every 19 files, about one screenful. Press the space bar to step one file at a time, or any other key to continue. Control-C exits the catalog, as usual.

ADJUSTING CAT.STEPPER

You may adjust the number of file names between pauses by Poking a number into location 838. For example, typing ‘POKE 838, 5” would pause every 5 file names.

CAT.FIXER (ProDos 1.1.1 only)

This program lets you enhance the functions of the ‘CATALOG” and ‘CAT” commands. The ‘Multiple File” options are handy for printer printouts of catalogs in multiple columns.

USING CAT.FIXER Type “-CAT.FIXER” to get the program going. For

reference, we have added letters A through G to the screen below:

CAT.FIXER CHOICES

```
CAT COMMAND                        CATALOG COMMAND
------------------            ----------------------
NORMAL CAT           A              NORMAL CATALOG
SMART 40/80        B              AUTO 80-COLUMN
MULTIPLE FILES       C              AUTO 80-COLUMN
MULTIPLE FILES       D              NORMAL CATALOG
```

USE CURSOR KEYS AND RETURN FOR SELECT

OPTIONS FOR MULTIPLE FILES CAT

```
40-COLUMN SCREEN                    80-COLUMN SCREEN
----------------                --------------------
NORMAL CAT            E               TWO FILES/LINE
TW0 FILES/LINE        F               FOUR FILES/LINE
COMPLETE MESS           G                FIVE FILES/LINE
```

CAT/CATALOG OPTIONS (top half of screen):

Use the arrow keys to move the flashing pointer to the set of options (A, B, C or D) that you want. (If your Apple can’t display 80-columns, only options A and D are appropriate.):

A: CAT acts normal.
CATALOG acts normal.
B: CAT acts like CATALOG if 80-columns visible.
CATALOG auto-switches to 80-column screen.
C: CAT makes multiple-columns.
CATALOG auto-switches to 80-column screen.
D: CAT makes multiple-columns.
CATALOG acts normal.
Press Return to select option A, B, C or D. If you select option A or B, skip to step 3. Selecting options C and D will move the pointer down to the bottom half of the screen. Press Esc to exit the program.

2. MULTIPLE-FILE OPTIONS (bottom half of screen):
   The options on the lower half of the screen apply to the CAT command only. Use the arrow keys to move the pointer to the set of options (E, F or G) that you want:

   E: 40-columns: normal CAT.
   80-columns: 2 file names per line.
   F: 40-columns: 2 file names per line.
   80-columns: 4 file names per line.
   G: 40-columns: Unreadable
   80-columns: S file names per line.

Press Return to select option E, F or G; or press Esc to go back to options A, B, C and D.

3. DEMO: Press ‘D” to test your new CAT or CATALOG display. Or press Esc to skip the demo. Press Esc once more to quit the program.

NORMALIZING CAT AND CATALOG

CAT.FIXER alters ProDOS’s Catalog function in memory. To get things back to normal, type “ -CAT.FIXER” and select option A. Or reboot, or type “-BASIC.SYSTEM”.

DATE.SET (ProDos 1.1.1 only)

If you don’t have a clock/calendar for your Apple, use this program to set the date and time which will appear next to your saved file names (instead of “<NO DATE>”).

USING DATE.SET

1. Type “-DATE.SET” to load the program.
2. Type ‘CALL 20000” to run it.

When you CALL 20000, the current date and time in memory will be displayed. If no valid date is present, the date stored in DATE.SET itself will appear.

CHANGING THE VALUES Press the Left and Right Arrow keys to highlight the day, month, year, hours or minutes. Then press the Up and Down Arrows (or A and Z) to increase or decrease each one. (A “<NO DATE” option lies between December and January.) When you’re satisfied, press Return. Or press Esc to exit with the original values restored.

A DATE.SET BOOT-UP PROGRAM:

Here is a sample time-set program that you could save under the name “STARTUP” so it will run each time you boot a particular disk:

```
10 D$=CHR$ (4): HOME: DS=20000: REM DS = DATE.SET ADDRESS
20 PRINT D$”-DATE.SET”: CALL DS: REM GET DATE AND USER RESPONSE
30 IF PEEK(DS-1)<>0 THEN PRINT D$”BSAVE DATE.SET”
40 REM CONTINUE WITH PROGRAM
```

Line 30 in the program above saves DATE.SET with the new date if a change has occurred. With this program on your boot disk, the last boot date will always be given as the default value.

In case you’re interested, DATE.SET stores a zero in byte 19999 if the user exits with a date identical to the one stored in DATE.SET.

DISK.COPY.MORE

DISK.COPY.MORE is a program which modifies Extra K’s DISK.COPY program to work with 38 -track (instead of normal 35track) disks (see pages 29 and 31). If you don’t have Extra K, go buy it. We’ll wait here.

USING DISK.COPY.MORE

Type “-DISK.COPY.MORE” to run the program. When prompted, insert the Extra K disk and press Return. If you only have one 38 track drive, you must use a one-drive copy and do some disk swapping. To make things easier, put DISK.COPY.MORE and DISK.COPY on a 38-track disk along with a 304-block version of ProDOS (page 31).

CHANGING DISK.COPY.MORE FOR 40 TRACKS

If you have a special 40-track drive, here’s how to upgrade

DISK.COPY.MORE to work with it:

1. Type “BLOAD DISK.COPY.MORE”.
2. Type “CALL -151”.
3. Type “404A: 28” (Hex for 40).
4. Type “40A9: 34 30” (ASCII for 40).
5. Type “BSAVE DISK.COPY.MORE”.

If you want, you can look up your own hex and Ascii values for 39 or 8-track drives and plug them in.

ERROR.EDITOR

Type “-ERROR.EDITOR” to change any of ProDOS’s 19 error messages. The standard error messages will be displayed on the left half of the screen. The right side will show “-SAME-“ if the message has not been changed.

CHANGING ERROR MESSAGES

1. Press “E” to begin editing. The center message will be highlighted by an inverse bar.
2. Use the arrow keys to move to the message you wish to change. Then press Return.
3. Type your new message. You may use letters of the alphabet (except J, Q & Z); and the characters “/”, “(“, “)”, “.”, and “:”. Press Esc if you don’t want to change the message, or press Return when you’re done.
4. Select another message to edit if you want, or press Esc to leave edit mode.

The words at the bottom of the screen tells you:

LENGTH of your new message as you type it (20 characters maximum). AVAILABLE NIBBLES for your message. A nibble is half a byte. Some characters count as one nibble, and some count as two. A click will be heard whenever a two-nibble character is entered or deleted. Your message is limited to the available nibbles or 20 characters maximum. ALTERED is the number of messages changed so far. UPDATING BASIC.SYSTEM If you quit ERROR.EDITOR, your new error messages will be in effect until you reboot. (Test them with an illegal command or two.) If you want the new messages to be in effect every time you boot, you will need to update BASIC.SYSTEM on your boot-up disk:

1. Press “U” to update ProDOS’s BASIC.SYSTEM file. You will be asked “UPDATE DISK IN DRIVE 1? (Y/D)”.

2. Press “Y” to update BASIC.SYSTEM in the main directory of slot 6, drive 1. Or press “ D” to toggle between drives 1 and 2. Press any other key to skip updating. Note: If you have renamed BASIC.SYSTEM, change the variable S$ in line 5.

RESTORING ORIGINAL ERROR MESSAGES

Press ‘N” from the editor menu to normalize the messages in memory.
Then press ‘U” to normalize the messages on disk.
EXITING Leave the editor by pressing ‘Q” for Quit. Type ‘RUN” to restart the program with your altered error messages intact.

ABOUT TESTING ERROR MESSAGES

To test ProDOS’s ‘Syntax Error” messages, type somethin g illegal like ‘CAT3”. Typing something illegal like ‘COTALOG” will give you Applesoft’s “?Syntax Error” message instead. You can’t change this message, unless you buy a copy of our Beagle Basic disk.

RAM.SAVE 128K required

RAM.SAVE will copy all of your current RAM disk files onto a floppy disk in seconds, providing ‘permanent” storage for them. RAM.LOAD (next page) will load the files back into /RAM.

USING RAM.SAVE

1. Save a few files on /RAM in slot 3, drive 2.
2. Type “-RAM.SAVE”.
3. Insert an erasable formatted disk into slot 6, drive 1 (or the drive you are set up for—see page 69). Warning: All previous files on the flopav will be erased.
4. Press Return. All of the /RAM files will be copied onto the floppy disk. Until you reformat it, this disk will not hold more than 119 blocks (about half of normal).
5. Catalog the floppy. It is now called ‘/RAM”, and contains the same files as the original /RAM in slot 3, drive 2.

WHAT’S A RAM DISK ANYWAY?

A phantom disk named ‘/RAM” lives in imaginary slot 3, dr ive 2 of 128K Apples with ProDOS. It has the advantage of letting you load and save files into memory at lightning speed. The problem is, RAM disk files disappear when the power goes off.

The RAM disk works just like any disk, except it is fast and only holds 119 blocks-worth of files. To test your RAM disk:

1. Load a program.
2. Type ‘SAVE /RAM/FILENAME”.
3. Type ‘CAT/RAM” or ‘CAT,S3,D2”.
4. Or type ‘PREFIX/RAM” followed by ‘CAT”.

Get the picture? If not, you probably don’t have 128K or you’re not working with ProDOS.

RAM.LOAD

RAM.LOAD loads files into /RAM from a floppy disk created by RAM.SAVE (previous page).

USING RAM.LOAD

1. Type “-RAM.LOAD”.
2. When prompted, insert your special floppy disk that was created with RAM.SAVE. Warning: use only special /RAM disks created by RAM.SAVE or semi-disaster will surely result!
3. Press Return to load all of the special floppy’s files into /RAM.

FLOPPY /RAM NOTE You may add to or delete files from the /RAM floppy as long as the floppy was created by RAM.SAVE.

RAM.SETUP

This program lets you configure the RAM.SAVE and RAM.LOAD programs for the following two characteristics:

I. The slot and drive that files will be copied to and from.

2. Whether or not you want the program to pause and prompt you to insert a disk. The pause option prompts you to insert the /RAM floppy and press Return before file copying begins. The no-pause option immediately begins reading and writing data. Use the nopause option only if RAM.LOAD will be used on the /RAM floppy. Then you can insert it, type "-RAM.LOAD" and everything else will be automatic.

USING RAM.SETUP

1. Type "-RAM.SETUP".
2. Follow the instructions on the screen.

RENAMING TIP

You may want to rename RAM.LOAD and RAM.SAVE to reflect the slot and drive selected. For example, RAM.SAVE.S6.D2 could be the version that works with drive 2. To do this, insert your own diskette before saving the configured program. After it's saved, type "RENAME RAM.SAVE, RAM.SAVE.S6.D2".

RENEW

RENEW is a short routine for restoring Applesoft programs which have been zapped by a "NEW" command. Type " -RENEW immediately after "NEW" has been typed. If it works, hooray! If not, retype your program.

RUN.COUNTER

This program will print the date your Applesoft program was last Run and the number of times it has ever been Run, each time it is Run. You need to append your program to ours. Here's how you do it:

1. Type "-APPENDER" (unless the APPENDER program is already in memory).
2. Type "LOAD RUN.COUNTER".
3. Type "&STORE" to hide the program.
4. Load the Applesoft program that you want to use with RUN.COUNTER.
5. Make sure your program has no program lines numbered lower than 9.
6. Type "&RECALL" to append the two programs.
7. Change the value of P$ in line 1 to the name of your program. For example: 1 P$="MY.FILE"
8. Re-save your program using the name in line 1.

Your program is now ready to use. Every time it is Run, the program name will be printed on the screen with the Run number and last used date. After you enter today's date, the Rem in line zero will be updated and the program will be re-saved under the name defined by P$. To prevent the Run number from printing, delete program line 4. To prevent the date from printing, delete line 5.

SAVE.40 and SAVE.80

These programs let you save 40-and 80-column text screens on disk. You may view the saved screens by typing a hyphen followed by the name of the saved screen. 80column files are compatible with those created by CRT.WRITER.

USING SAVE.40

I. Bload SAVE.40 2. CALL 20134 USING SAVE.80

I. Bload SAVE.80

2. CALL 20010

After the CALL, the screen will be copied to a buffer on hi-res page 2, and you will be asked for a file name (a legal ProDOS pathname, which may include slot and drive). The screen will then be saved. To prevent your typed message from being saved, you should always save screens from within your programs. For example:

```
10 PRINT CHR$(4)"BLOAD SAVE.80"
20 FOR I=1 TO 20: PRINT "STUFF ALL OVER THE 80 -COLUMN SCREEN.": NEXT
30 CALL 20010 : REM SAVE SCREEN
40 REM CONTINUE WITH PROGRAM
```

Now that you’ve s aved a screen or two, type ‘HOME” and then “ -filename” (substitute your saved screen’s name for ‘filename”). Loading an 80-column screen automatically switches to 80-column display. Try this miniexample:

1 PRINT CHR$(4)”-FILE.NAME” : REM USE THE APPROPRIATE FILENAME 2 HOME : GET X$: CALL 20366 : REM USE CALL 20416 FOR 40-COLUMNS 3 GET X$: GOTO 2 : REM PRESS CONTROL-RESET WHEN YOU GET BORED The CALL in line 2 keeps recopying the screen from its buffer.

PRODOS COMMAND SUMMARY

(Deleted. See other references for this information)

MEMORY USAGE

Most of the BIG U AMPER and COMMAND programs take away some programming memory by moving HIMEM down and installing themselves above the ProDOS buffers. The following list shows how much memory is used by each program:

One-pagers (256 bytes): APPENDER, DUMP.40, DUMP.80, EST, ONLINE and SUPER.POKE.

Two-pagers (512 bytes): ANYCAT, COPY, COPY.ONE, DUMP, HEX, INPUT.40, INPUT.80, REMOVE, REM.OVE.128, SEE, SHOW and SHOW.WIPE.

Four-pager (1024 bytes): MON
Five-pager (1280 bytes): XLISTER

PAGE 3 USAGE

Page 3 is the memory from $300 to $3FF (768.1023 decimal). ProDOS reserves everything from $3D0 on, leaving 208 bytes free for programmers. Here’s how BIG U divides it up:

Program	Hex	Decimal
CAT.STEPPER	300.34A	768.842
CAT.DATER	34B.35D	843.861
CAT.FIXER	35E.375	862.885
Unused Memory	376.3B0	886.944
KEYCAT (Reset Versian)	3B1.3CF	945.975
PRODOS	3D0.3FF	976.1023

INDEX

(Deleted due to the page numbers no longer making sense.)