

A journal and exchange of Apple II discoveries

Smarter Smartports

By Dennis Doms

Some months ago (January 1987, to be precise) Tom Weishaar wrote an article for *A2-Central* (then called *Open-Apple*) entitled "How to keep from getting pregnant" about the Apple SmartPort protocol that allows connecting multiple devices to the Apple II. (While you're at it, you may want to check out the letters entitled "Smartport lines missing" and "Smartport Follow-ups" in the Feb. 1987 issue.) The SmartPort has been used as a protocol for connecting disk devices to Apple IIs, although support for character devices is also defined. A wide range of disk devices, including RAM disks, floppy disks, and hard drives, have been supported through various SmartPort controllers such as the IIgs disk ports, the Apple SCSI interface, and the Apple II Memory Expansion Card.

The new Apple SuperDrive Controller is also based on that SmartPort protocol, but adds a new twist: it supports removable media (floppies) that can be formatted in a range of physical (low-level) formats. This information wasn't in our original article or in most of the Apple manuals; the best reference for the updated information is in *The Apple IIgs Firmware Reference Update* available from (ahem) Resource Central. Here's a quick look at how you can pick a format for a multiple-format floppy.

Most procedures are the same as in the original program.

First, we do a slot scan that looks for the SmartPort identification bytes. If they check out, we calculate the SmartPort entry address (the slot's base firmware address plus the ProDOS entry address offset in \$CnFF plus 3). At the same time, our program grabs the SmartPort ID byte at \$CnFB (where "n" is the slot number) which we'll use to check some features of the interface.

Incidentally, the *Firmware Reference Update* mentions that the interpretation of that byte has changed somewhat. The following bits are "true" for the associated conditions: bit 7, the interface driver supports extended commands (extended commands are used to access the interface more efficiently from IIgs 16-bit applications); bit 6, the driver supports additional commands ("additional" meaning the additional calls defined in the update); bit 1, the driver supports SCSI devices; and bit 0, the driver supports a RAM card (RAM disk).

Using the Status call, we find out how many devices are attached to the SmartPort interface. The call is performed by a subroutine POKEd into memory in lines 2270-2300; the subroutine is a little different than the original:

JSR	SPEntry	jump to entry address
DC	11'Command'	Command number
DC	A2'ParmList'	Parameter list
ROR	ErrFlag	Use to save error (carry) status
STA	ErrCode	save error code (if any)
CLC		make BASIC happy
RTS		
ErrFlag	DC	11'00'
		space for error flag

ErrCode	DC	11'00'	space for return code
ParmList	DC	11'00'	space for parm count
			(various parameters follow)

Our first Status call uses a subcommand of \$00; from this we get a general status byte that we check to see if the device is a block device. If it is, we do another Status call with a subcommand of \$03 to return the device's Device Information Block (DIB); from this we get several pieces of information, including the device's name string.

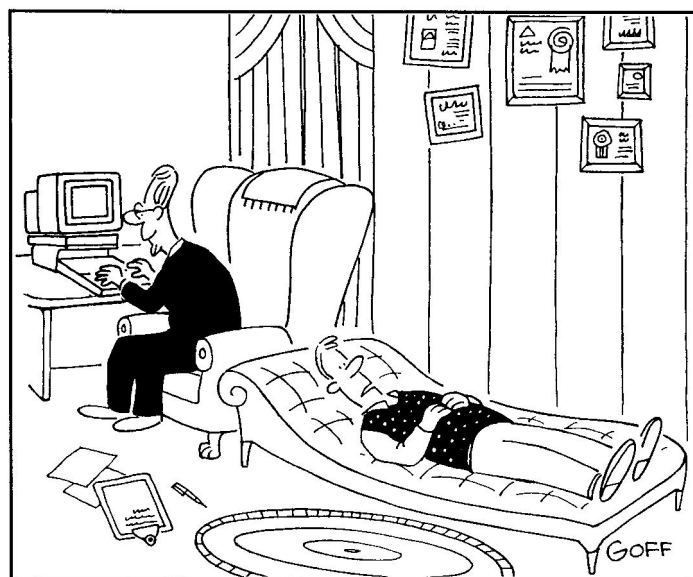
If it's a high-density 3.5, then let's check the format options. Here is where we get into some of the additional calls, specifically the GetFormatOptions and SetFormatOptions calls.

While our program scans the devices, it looks for a device that identifies itself as a "DISK 3.5HD", which is a high-density 3.5 drive. If it finds one, it sets up the SmartPort routine to use the command for GetFormatOptions, the new command to return a list of possible formats for the devices.

The GetFormatOptions parameter list consists of a count of 3, the unit number (byte), the buffer location (\$2000), and a buffer size (at least 256 bytes is recommended; we used 512).

The Firmware Reference Update is pretty confusing in its description of something it calls a "media" list. The media list is shown as a parameter list item (the fourth of three?) but actually seems to refer to the data returned in our buffer. Working from that, we come up with the following layout.

First is the Entry Count (word); each format entry is eight words long. Next is the Display Count (word), the number of entries to dis



"I JUST CAN'T STAY AWAY FROM MY APPLE COMPUTER, DOCTOR. DO YOU THINK THAT'S UNHEALTHY? DOCTOR?"

play for the user. The next word is the reference number for the recommended option (if any) for the current media. Next is the reference number for the format of the current media (if the device determines and returns it). After this come the format options.

The first word of each format option is an arbitrary (firmware specific) reference number that can be passed back to SetFormatOptions as an identification for the format. The next word is supposed to be a link reference to another entry; but for the 3.5 controller we found these all to be "\$0000". Following this is a "flag word"; bits 15-4 are reserved, bits 3-2 indicate the size units ("00" is bytes, "01" is kilobytes, "10" is megabytes, and "11" is gigabytes), and bits 1-0 indicate the "recipient type" ("00" is a Universal format that accepts any file system, "01" is for Apple or universal file systems, "10" accepts non-Apple or universal file systems, and "11" is reserved). Next is a long word (four bytes) that contains the block count for a format operation. Following is a word indicating the block size in bytes (often 512 to support ProDOS devices). Then we have a word indicating the interleave factor, and finally a word for the total volume size expressed in the Size Units indicated by the flag word. So let's get and display this information; at the same time we'll save the format reference numbers for use in a moment.

Okay, now we know what the options are. Let's try using the "flip side" command SetFormatOptions to prepare for and execute a SmartPort format command.

Obviously, formatting will destroy all information currently on the disk, so you shouldn't try this with a disk that means anything to you. Less obviously, the "format" we're performing is only a low-level physical format that places recording markers on the disk so that an operating system can perform block reads and writes to it; it doesn't build and write the logical formatting (such as a ProDOS 8 directory) that is required for a file system (such as ProDOS) to recognize it. This is (as they say) left as an exercise. Normally a formatting utility for an operating system will perform the logical formatting after physically formatting a disk.

Having collected the format options with GetFormatOptions, we can now tell the device how we want to format a disk by feeding it a valid reference number with SetFormatOptions (command code \$0A). Again, the parameter list in the Firmware Reference seems confused, of three parameters, the first three are the same as for GetFormatOptions. But the following word is identified as the format reference number, and the word after that (fifth of three parameters?) is the interleave (we used \$0000 to accept the format's default).

With the options set, we can do the real format. The command code is 3; the parameter list only needs the device unit number to receive the format.

With that, let's bring on the program:

```
1000 REM — play with SmartPort, 1992 —
1010 GOSUB 2260: REM install ML for SmartPort calls
1020 REM now define some constants and arrays
1030 LOWMEM: 16384
1040 DIM SP(7),SI(7): REM save for entry addresses, ID byte
1050 DIM B(7),B$(1): REM to parse byte to bits, state (you'll see)
1060 DIM FR(10): REM format reference number array
1070 B$(0) = " NO":B$(1) = " YES"
1080 CM = 771:EF = 782:EC = 783:PL = 784: REM and define a few locations
1090 DIM DC(7): REM for device counts

1110 REM — locate SmartPort slots and characteristics —
1120 GOSUB 2130: REM find SmartPorts
1130 POKE PL + 2,0: POKE PL + 3,32: REM statlist at $2000
1140 FOR S = 1 TO 7
1150 PRINT "Slot ";S;": ";
1160 : IF SP(S) = 0 THEN PRINT "<not SmartPort>": GOTO 1200
1170 : PRINT "Entry address = ";SP(S); SPC( 3);
1180 : BYTE = SI(S): GOSUB 2200
1190 : PRINT "ext = ";B$(B(7));": addnl = ";B$(B(6));": SCSI = ";B$(B(1));":
```

```
RAM = ";B$(B(0))
1200 : NEXT S

1220 REM — find device count —
1230 FOR S = 1 TO 7
1240 : IF SP(S) = 0 THEN 2100
1250 : PRINT "— Slot ";S;": ————"
1260 : POKE 770, INT (SP(S) / 256): POKE 769,SP(S) - PEEK (770) * 256: REM
call vector
1270 : POKE CM,0: REM status
1280 : POKE PL,3: REM parm count
1290 : POKE PL + 1,0: REM unit num
1300 : POKE PL + 2,0: POKE PL + 3,32: REM buffer at $2000
1310 : POKE PL + 4,0: REM return device status
1320 : CALL 768
1330 : IF ( PEEK (EF) >= 128) THEN PRINT "Error #"; PEEK (EC);" occurred on
Status (controller) call."
1340 :DC(S) = PEEK (8192)
1350 : PRINT "There are ";DC(S);" devices..."
1360 : IF NOT (DC(S)) THEN 2100: REM no devices

1380 REM — scan info on all devices —
1390 : FOR D = 1 TO DC(S)
1400 :: POKE CMD,0: REM status call
1410 :: POKE PL,3: REM parm count
1420 :: POKE PL + 1,D: REM device number
1430 :: POKE PL + 2,0: POKE PL + 3,32: REM buffer
1440 :: POKE PL + 4,0: REM return device information block (DIB)
1450 :: CALL 768
1460 :: IF ( PEEK (EF) >= 128) THEN PRINT "Error #"; PEEK (EC);" occurred on
status (device) call."
1470 :: BYTE = PEEK (8192): GOSUB 2200
1480 :: IF ( NOT B(7)) THEN PRINT SPC( 5);"Device ";D;" is not a block device;
skipping...": GOTO 2090: REM next device
1490 :: POKE PL + 4,3: REM get DIB
1500 :: CALL 768
1510 :: IF PEEK (EF) >= 128 THEN PRINT "Error #"; PEEK (EC);" occurred on
status (DIB) call."
1520 :: N$ = "":L = PEEK (8192 + 4): FOR I = 1 TO L:N$ = N$ + CHR$( PEEK (8192
+ 4 + I)): NEXT I
1530 :: PRINT SPC( 5);"Device ";D;" calls itself ";N$;"."
1540 :: IF N$ <> "DISK 3.5HD" THEN GOTO 2090: REM not HD floppy
1550 :: PRINT SPC( 7);"type: "; PEEK (8192 + 21);": stype: "; PEEK (8192 +
22);": vers: "; PEEK (8192 + 23);": "; PEEK (8192 + 24)

1560 :: REM — get and report format options —
1570 :: REM Notice we haven't checked for extended options generically.
1580 :: REM (But we did check the type of device via the name.)
1590 :: POKE CM,11: REM GetFormatOption
1600 :: POKE PL,3: REM parm count
1610 :: POKE PL + 1,D: REM unit number
1620 :: POKE PL + 2,0: POKE PL + 3,32: REM still use $2000
1630 :: POKE PL + 4,0: POKE PL + 5,4: REM $1000 bytes for buffer size
1640 :: CALL 768
1650 :: IF PEEK (EF) >= 128 THEN PRINT "Error #"; PEEK (EC);" occurred on
GetFormatOptions call."
1660 :: FC = PEEK (8192) + PEEK (8192 + 1) * 256: REM formats
1670 :: FU = PEEK (8192 + 2) + PEEK (8192 + 3) * 256: REM fmts to display
1680 :: FD = PEEK (8192 + 4) + PEEK (8192 + 5) * 256: REM rec. option
1690 :: FT = PEEK (8192 + 6) + PEEK (8192 + 7) * 256: REM cur. format
1700 :: PRINT SPC( 7);"Format options (";FU;" of ";FC;":)"
1710 :: FOR F = 1 TO FC
1720 :: BASE = 8192 + 8 + 16 * (F - 1)
1730 :: FR(F) = PEEK (BASE) + PEEK (BASE + 1) * 256: PRINT SPC( 9)"Ref:
";FR(F);": ";
1740 :: PRINT "Link: "; PEEK (BASE + 2) + PEEK (BASE + 3) * 256;": ";
```

```

1750 :::BYTE = PEEK (BASE + 4) + PEEK (BASE + 5) * 256: GOSUB 2200
1760 :::SU$(0) = "B":SU$(1) = "K":SU$(2) = "M":SU$(3) = "G"
1770 :::RT$(0) = "UF":RT$(1) = "AF":RT$(2) = "NA":RT$(3) = "xx"
1780 ::: PRINT "Blks: "; PEEK (BASE + 6) + PEEK (BASE + 7) * 256 + PEEK (BASE
+ 8) * 256 ^ 2 + PEEK (BASE + 9) * 256 ^ 3;"; ";
1790 ::: PRINT "Size: "; PEEK (BASE + 10) + PEEK (BASE + 11) * 256;"; ";
1800 ::: PRINT "I/L: "; PEEK (BASE + 12) + PEEK (BASE + 13) * 256;"; ";
1810 ::: PRINT "FSze: "; PEEK (BASE + 14) + PEEK (BASE + 15) * 256;SU$(B(3) *
2 + B(2));
1820 ::: IF F = FD THEN PRINT " RECOMMENDED";
1830 ::: IF F = FT THEN PRINT " CURRENT";
1840 ::: PRINT
1850 ::: NEXT F

1870 :: REM — try to format (on request) —
1880 :: INPUT "Format this device? (Y/N): ";A$
1890 :: IF LEFT$(A$,1) <> "Y" AND LEFT$(A$,1) <> "Y" THEN GOTO 2090
1900 :: INPUT "OK...which option (above)? (Ref. num.): ";A
1910 ::FM = 0
1920 :: FOR I = 1 TO FU: REM verify reference number is valid
1930 :: IF (FR(I) = A) THEN FM = 1: REM okay to try format
1940 :: NEXT I
1950 :: POKE PL + 0,3: REM still 3 parms (??)
1960 :: POKE PL + 1,D: REM same unit number
1970 :: POKE PL + 2,0: POKE PL + 3,32: REM same buffer
1980 :: POKE PL + 4,0: POKE PL + 5,2: REM same length
1990 :: POKE PL + 6,A: REM format ref num
2000 :: POKE PL + 7,0: REM interleave; use ref default
2010 :: POKE CMD,10: REM SetFormatOption
2020 :: CALL 768
2030 :: IF PEEK (EF) >= 128 THEN PRINT "Error #"; PEEK (EC);" occurred on
SetFormatOption call."
2040 :: POKE CMD,3: REM format
2050 :: POKE PL,1: REM only one parameter
2060 :: POKE PL + 1,D: REM device to format
2070 :: CALL 768
2080 :: IF PEEK (EF) >= 128 THEN PRINT "Error #"; PEEK (EC);" occurred on
Format call."
2090 :: NEXT D
2100 :: NEXT S
2110 END : REM main loop

2130 REM — find smartport —
2140 FOR S = 1 TO 7
2150 :BA = (12 * 16 + S) * 256: REM base address of $Cn00
2160 : IF ( PEEK (BASE + 1)) = 32 AND ( PEEK (BASE + 3) = 0) AND ( PEEK (BASE
+ 5) = 3) AND ( PEEK (BASE + 7) = 0) THEN SP(S) = BASE + 3 + PEEK (BASE
+ 255):SI(S) = PEEK (BASE + 251): REM entry addr, ID byte
2170 NEXT S
2180 RETURN

2200 REM — parse byte to bits —
2210 FOR B = 7 TO 0 STEP - 1
2220 :B(B) = 0: IF BYTE >= 2 ^ B THEN B(B) = 1:BYTE = BYTE - 2 ^ B
2230 : NEXT B
2240 RETURN

2260 REM — ML for SmartPort (standard) call —
2270 FOR I = 768 TO 783: READ X: POKE I,X: NEXT I
2280 DATA 32,0,0,0,16,3,110,14,3,141,15,3,24,96,0,0
2290 RETURN

```

Oddities. One unexpected problem we ran into with the Apple 3.5 Controller and the AMR 3.5 drive is that the two apparently won't cooperate in formatting an 800K disk. The AMR will format an 800K disk fine on the IIgs disk port but not on the new controller. Apparently the SetFormatOptions command is not recognized by the AMR because when we select to format an 800K disk the AMR instead tries to format it as a 1,440K disk.

The 720K format reported is intended for use with the normal "double-density" MS-DOS disk format. These disks cannot normally be read on Apple 3.5 drives because they use a different electronic encoding than the Apple 800K drives, but the high-density drives support both encoding schemes. It is possible now to read physical blocks from MS-DOS disks using normal block read and write commands; the 3.5 Controller is intelligent enough to adjust itself for the format in use. Blocks on 1,440K disks can also be read; the low-level physical format for these disks is the same for either Apple or MS-DOS disks.

Apple Writer: Built in-menus

by Ron Evry

As you have seen in the previous two articles, *Apple Writer* is fast and easy to use. You may never need to learn more than you have already. The following features of this freeware program may be difficult to memorize, but as you use them, you will find them becoming second nature.

Apple II's are frequently configured with a single 5.25 inch disk drive. Users of these machines can be frustrated by all of the disk swapping required by many programs. This is not a problem for *Apple Writer* users. Since this freeware program is almost entirely memory-resident, once booted, it can be removed from the drive altogether. Incredibly, there are three built-in menus featuring an array of functions that are instantly available without accessing the disk.

The first is the ProDOS Commands menu. To access it, press CONTROL-O. The text screen will temporarily disappear and be replaced by a menu. Some features of this menu are indispensable; we will examine them all. Pressing the A key will allow you to catalog any online device. At the prompt "Enter Volume Name:" you must type in the correct ProDOS pathname you wish to catalog, complete with slashes (ex: /DATADISK/LETTERS). If your default prefix has already been set, pressing RETURN at the prompt will catalog that particular volume or subdirectory. If you wish to catalog whatever volume is in the drive, you can use a period before a slash (./) as a shortcut.

If you do not know what volumes are available to load or save to, pressing F (List Volumes On-Line) will read every drive and give the volume names. If a drive is empty, it will list a slash with no name. Then, by pressing H, you can set a volume or a more complete pathname as your default.

Pressing G will enable you to create a subdirectory. At the prompt, either type in a complete pathname with slashes, or if your prefix has already been preset, just the name of the subdirectory you are creating without the the slashes. Subdirectories are a useful way of organizing your text files by categories, and on large storage devices such as 3.5 drives and hard drives, are a vital way of getting by the 51 file limit in the main volume directory.

Other options in the ProDOS Commands menu include Rename File (press B), which asks for the name of your old file and then asks how you wish to rename it, Lock File (press C), which protects a file from being accidentally overwritten, Unlock File, (press D), in case you have changed your mind about the previous option, and Delete File (press E), which will permanently clear a file from a disk (this option will not work on a locked file).

Pressing I will enable you to format a volume (a handy feature if you have a text file and nowhere to save it). The J key will set up a printer or modem interface, which may be necessary if you have an esoteric printer or you have put your interface card in any slot besides 1 or 2. It will also set up parity, baud rate and echo for your modem if you are using the program for that purpose. *Apple Writer* was my first telecommunications program, and I quickly realized that

it has way too many shortcomings to be practical for this purpose. There are much better telecommunications programs available commercially, in the public domain, or as shareware. If you are using a patched copy of *Apple Writer* on a IIgs, these last two features will be disabled.

Pressing return at the prompt in the ProDOS Commands menu will put you back where you left off in your text document.

Pressing CONTROL-Q will bring up the Additional Functions menu. These functions are utilized less frequently than the ProDOS Commands, but from time to time, they can be quite useful. Earlier on, we used the CONTROL-T function to set up tabs. You may want to save those tab settings for later reuse. In that case Additional Function B. Save Tab File will do this for you. Simply write the filename you wish to save the tab settings under at the prompt. You may give a tab file the same name as a document file you are working on, since *Apple Writer* automatically adds the suffix ".TAB" to it. If you name your tab file "SYS", it will replace the "SYS.TAB" file on the *Apple Writer* disk and be loaded in on boot-up. Since this will destroy the existing default tab settings you will be asked if you want to do this at the prompt.

Loading in a tab file is simplicity. Choose option A in the Additional Functions menu and type in the tab filename. Do not include the ".TAB" suffix. If you need a catalog, enter a question mark at the prompt. These tab settings will need to be reloaded every time you boot up up *Apple Writer*, if you need them.

Option G in this menu will toggle the carriage return display. When it is on, you can see a bent arrow at every carriage return. This feature is useful when laying out columns of type, preparing documents that need carriage returns after every line (such as for GEnie email), or for setting up text files to use in your programs.

Option H toggles the top bar of the screen. This seems to be a redundant feature, since pressing the ESCAPE key will do the same thing. However, having this option does open the way for a clever programmer to utilize the ESCAPE key for some other function in a patch.

If you press option I, you can use your printer just like a typewriter, which makes addressing labels or envelopes easier. When in the typewriter mode, you must type RETURN at the end of each line. Everything you type will be printed, even if you back over and change a line. Interestingly enough, you may enter this mode while you are in the middle of writing a document, and when you quit (ESCAPE followed by the Q key) you will return right where you left off.

This option will connect you to your modem also but not with the patched IIgs version. You must first set your destination slot with the CONTROL-O, J option. You cannot upload or download files, or emulate any terminals with this program. It is only useful for sending and receiving text.

Option J will enable you to quit *Apple Writer*. As a safety measure, you will be given the chance to back out and save your document in memory first. Once you quit, everything you have not saved will be erased. Quitting puts you back into ProDOS, or the Finder, if you booted from there.

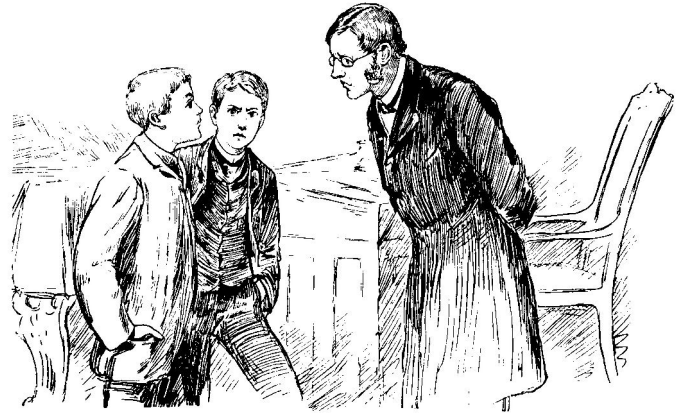
Options C and D load and save Print/Program value files, much in the same way options A and B worked with tab files. The default Print/Program Command menu is called "SYS.PRT" and using option D to save one of these files will automatically place the ".PRT" suffix on the filename.

If you are wondering what the Print/Program Value File is, press CONTROL-P and enter a question mark at the prompt. There you will see an entire menu of printing and page formatting commands. These can be changed on a temporary basis just for a session, or changed and saved as a loadable value file, or as the default file. Each value should be examined. Once you are familiar with them, you do not have to bring up the value menu to change any of them.

Simply press CONTROL P to enter in the new value at the prompt (ex: (P)rint/Program: LM10). Remember, these settings are consistent throughout a document. To make changes for only part of a document, you need to use embedded characters, which we will touch on at another time.

The values "LM" and "RM" refer to the left and right margin settings, respectively. The default setting for the left margin is zero and the right margin is 78. You can set the right margin to any number up to 240 (assuming your printer can handle that many characters per line).

"PM" refers to the paragraph margin setting. This is the number of spaces indented from the left margin after a carriage return. If you



wish to avoid tabs in your files, but want to see them in your print-outs, then this feature was made for you.

Incidentally, there are two good ways to see what your document will look like when printed out. If the only formatting commands you have are "LM" and "RM" changes, then the CONTROL-A (adjust margins) command will reformat the screen to accommodate them. If you've made other formatting adjustments, you may preview the print-out by first changing your print destination to the screen (CONTROL-P then pd0), followed by a standard print command (CONTROL-P then np). This will display your document on the screen exactly as it will look on the print-out. To stop and start the display as it rolls by, press CONTROL-S. Do not forget to set the print destination back to slot 1 (or wherever your printer is) with "pd1".

Other settings include the number of blank lines from the top of the page before the first line is printed ("TM") and the number of blank lines between the last line of text and the bottom line ("BL"). The default setting for these values is 1. The top line is called the header and the bottom line is called the footer. It is possible to use these lines to print title or page information that will run page by page through your documents, or to print footnotes on individual pages.

The standard setting for "PL" (printed lines) is 58. This is the total number of lines on the page from the top line down to the bottom line, including space for footnotes. Generally regular 11" long paper can stand a few more lines on the page, if you think it may be necessary. This setting will probably need to be changed if you alter the "PI" (page interval) value, which is based on six lines of text per inch (for example, since a standard sheet of paper is 11" long, there are 66 total lines on a page, which is the default "PI" setting). If you are using legal sized paper (14" long), you may wish to set this figure at 84, then set "PL" at 76 or so. Please note that setting "PL" at 0 or 1 will cause *Apple Writer* to hang up on you and you will have to re-boot. So don't do that.

As we mentioned in the first part of this tour, the "LI" setting will add line feeds if your printer needs them. The standard setting of "li0" works with my Imagewriter, and if I need to double space a document, then I change to "li1". Also you may notice that the default

setting for "CR" (carriage return) is 1. This issues a line feed to your printer at the end of each line. I have adjusted my default setting for "cr0", which works for my Imagewriter. You may have to check your printer to see which is best.

Another setting is "SP", which allows you to use single sheets of paper instead of pinfeed. The default setting is 0. Change it to 1 if you want to use single sheets or envelopes.

"UT" refers to the underline token. Using it once turns on the underlining, and using it again shuts it off. The default token is a backslash ("\"), but you may wish to change it to another character. Just type "UT" and whatever character you want to use. If you download a lot of text files, you will probably want to get rid of the token altogether, as your printer will go berserk if it runs into a token and then has to print twenty underlined pages. To do this, just type "UT" and press RETURN. By the way, underline tokens actually are printed as a space, so use one instead of a space between words. If your token is the first space on a line, it will not print anything at all.

You may choose from four justifications for your documents. This feature controls how text fills the space between the margins. The default setting of "LJ" (left justification) will run all of the text flush against the left margin of the document. "RJ" will do the same with the right margin, leaving the left side ragged. For setting up smooth columns of type, you can combine "FJ" (full justify) along with tightened margins. You can use the preview mode to see your columns before you print them and adjust them without wasting paper.

One other justification choice, "CJ" (center justification) is pretty useless on an entire document, unless you want to set up weird poetic-looking type. This justification mode is terrific for centering titles though, so what is needed is a way to enter values for part of a document and then revert back to the defaults. This can be done by embedding printer commands in a document. If you wanted to center a title and then switch back to standard left justification, you would simply type ".CJ" (without the quotes) at the far left margin of your document, above the line to be centered. Then type ".LJ" on the line after the title.

You can also move your left and right margins around in a document as well this way, making it possible to lay out the pages of a screenplay, for example. If you do not relish the idea of typing ".LM40" and ".RM80" every time you want indented columns, these commands can be put into glossary entries that will do the work for you. More about those in a bit. Just remember, that whatever embedded formatting commands you print last will stay on the menu until you change them, even if you load in and print a brand new file.

The settings for "TL" (top line) and "BL" (bottom line) will put titles, page numbers or other information at the top or bottom of each page. The headers and footers are laid out in the same manner. Press CONTROL-P then type "TL" or "BL" followed by a delimiter (you can actually use any character you want for a delimiter here), what you want to see on the left of the page, another delimiter, the center text, another delimiter, the far right text, then a final delimiter. Here's an example:

```
[P]rint/Program:BL/Ron Evry/Apple Writer, the Guided Tour/A2-Central/
```

This will appear on the bottom line of every page of your document when printed. If you'd like to put page numbers on your document, put the "#" symbol where you want your page number to show up (ex: TL/Evry/A2-Central/page #/). If you want to skip page one in your-page numbering, use backslashes as your delimiters. If you need to underline in your header or footer, you will have to choose a different character than a backslash, if you are also skipping page numbering on the first page. Got it? Good.

Now suppose you have a really long text file and it is broken up into more than one file. You've printed the first thirty pages from one file and now you want to print the next file, starting on page thirty-one and you want to number the pages where you left off. No sweat. Press

CONTROL-P and at the prompt type PN31 (or PN whatever number you're on) and press RETURN. Your header or footer will start page numbering from wherever you say.

Here is a handy hint for those long, long documents. If you want to find out exactly what page your cursor is on, press CONTROL- (yes, that is Control-hyphen). Move around with your arrow keys and the information displayed will change. To exit from this mode, start typing again.

Please remember that a footer is not a footnote. *Apple Writer* does have a dandy footnote feature available though, and here is how it works. You do not have to figure out where the bottom of the page is going to be. Simply put your cursor just to the right of the footnote reference in the body of your document. Then type a parentheses and a less-than sign (ie: (<), the text of the footnote, and end with a more-than sign and parentheses (ie: >)), followed by a space. The footnote will show up at the bottom of the same page as your reference, automatically.

The text of your footnote can not be any longer than the space between your margins, but you can have more than one line of footnotes, as long as there is a space between each of them. You do need to limit your footnotes to no more than 1024 characters on a single page. And if you are running a long WPL program along with footnotes, they could get in each other's way. But that is something that can wait until we get to WPL. Two or more footnotes can be separated by a footnote that only contains one or more blank spaces. Finally, if your footnote is on the last page of your document, you must put in an embedded form feed command, or it will not print.

When you print, there are three very important printer control characters that will only work by being embedded in your text. They all need to be placed at the left margin to work. First (more logically, last) is the ".FF" (form feed) command, which ejects the paper in your printer out to the end of the page. This is the easiest way to put page breaks in where you want them, and combined with *Apple Writer's* preview mode, enables you to polish your page layouts nicely.

Second, there is the ".EP" command, which is really two commands in one. ".EP1" tells your printer to print from that point onward. ".EPO" tells your printer to not print from that point on, until it reaches an ".EP1". If you put an ".FF" just before an ".EP1", it will begin printing at the top of the next page. Just remember that your page numbers will be unaffected by ".EP" commands, so you may wind up printing page one followed by page three.

The third of the major embedded characters is the ".IN" command. You can write messages to yourself following this command and your printer will stop, put the message on the screen ONLY, and resume printing when you hit the RETURN key. The message will not print. As a matter of fact, you can put in messages to yourself by starting a line (after a carriage return) with a period. It will not stop the printer and it will not print. End your embedded messages to yourself with a carriage return, of course, and if you have a need to print a line that starts with a period, put a space in first.

Another way to embed commands is to use the Control-V option. Any control characters you insert between two Control-V's will actually go into your document and not be executed immediately. This is a great way to put in printer specific control character commands for boldface, italics, and whatever else your printer is capable of doing. Control characters not preceded by a period can be placed at any point in a document, not just at the beginning of a line. *Apple Writer* comes with a glossary on the disk called SPECIAL which has a number of Imagewriter specific control characters. When SPECIAL is loaded, all you need to is press the appropriate Open Apple key combination to embed the instructions to your printer. More on glossaries in the next installment.

Miscellanea

Quality Computers bills *Six Pack*, by talented programmer Bill Tudor, as "a refreshing collection of utilities." I found it to be more than refreshing, I found it to be more like exhilarating. *Six Pack* is designed for use with System 6 for the Apple IIgs and a hard drive. If you haven't upgraded to System 6, you don't need *6 Pack*. But if you have upgraded to System 6 and you have at least 2 megabytes (you'll need 3 megs to load all the modules at once), you really shouldn't miss *6 Pack*.

Six Pack contains 15 utilities; 10 FinderExtras and 5 New Desk Accessories (NDA's). One of my favorites includes *FilePeeker*. It allows you to peek inside just about any type of file to see its contents. Click on a graphics file and you see what the graphic is. Click on an icon file and see what the icons look like. It's addicting! Another is *PrintCatalog*; it prints a catalog of any window or disk icon to your screen, printer or a disk file. It's very flexible and allows to you determine exactly what type of information you want archived. *IR* is a FinderExtra that allows you to activate other FinderExtras, Desk Accessories (NDAs and CDAs), inits and drivers from the Finder instantly without having to reboot the computer. It is the one part of *Six Pack* not written by Bill Tudor but by Matt Deatherage of Apple Computer Inc.'s Developer and Technical Support Department. *OpenSesame*, a new desk accessory, takes this concept one step further in that it allows you to do all these things from any desktop application. Both are real handy to have around. With these two goodies you can also enable seldom used drivers. *HotKeys* is another hot item that can easily spoil you. It allows you to use one or two key strokes to open often accessed folders on your hard drive. For example, hit 2 on your numeric keyboard and your System/Desk.Accts folder opens. *HotKeys* comes preconfigured but you can assign the keys as you wish. *MoreInfo* gives you more information about files than the Finder's Icon Info menu item. It also allows you to change that information (file types and creation dates, for example) and to lock/unlock files or enable/disable system files. For more information, contact Quality Computers, 20200 Nine Mile Road, St. Clair Shores, Mich, 48080, 800-777-3642.

Following on the heels of Apple Central Expo and Apple East Expo comes Apple West Expo. This event will take place on April 23-25, 1993 in Brooks Hall, San Francisco, Calif. Travel arrangements and discounts have been made through American Airlines. For hotel information and other details get in touch with Event Specialists at 17 Lilac Road, Sharon, MA 02067, 800-955-6630, 617-784-4531, 617-784-1830 (fax).

Due to numerous requests, II Productive's (1008 Ridgmont Street, Round Rock, Texas 78664) TrueType Font Collection now includes a hard copy containing samples of every font in the collection. The examples are small (they all fit on two pages) but large enough to see what the font looks like. If you would like a copy write to Marvin Gershowitz at the above address and include a dollar for shipping and handling.

Contrary to the review in the February issue of A+InCider, Contacts GS by Simplexity Software does not require HyperCard IIgs. It's a new desk accessory that keeps an address and phone book at hand, easily accessible from any desktop program. What makes this one different from others is the fact that it will dial your phone for you. It will allow you to import the ASCII files it contains to any data base program for formatting and printing. Contact Simplexity Software at 13045 Chapman Avenue, Suite 302, Orange, Calif. 92668, 714-283-3957.

Playing with the *Switch-It!* Demo (file #19997 in GENIE's A2 library) was really frustrating! After a short romp with the demo, I wanted the real thing. The demo allows you to load two desktop applications into memory then switch back and forth between them. The real program allows for running many more programs depending

on the amount of memory you have installed in your computer. The demo has a ten minute limit, a really clever way to leave the user begging for more! *Switch-It!* was programmed by Jawaid Bazayr of Procyon, Inc. and will be distributed by Sequential Systems (1200 Diamond Circle, Lafayette, Colo. 80026, 800-759-4549). Resource Central will be among the many mail order houses who are sure to carry *Switch-It!*

MD-BASIC 2.0 is the latest version of a fully integrated Apple-soft BASIC development environment for Apple IIgs. New features include a desktop interface with pull down menus, windows and mouse support. This new desktop interface includes the option of multiple windows, full text editing, choice of fonts and tabs, a complete search and replace feature, printer support, integrated processing, and Applesoft-to-source conversion. *MD-BASIC* can now be used as a stand alone application or as an external tool for command line environments. *MD-BASIC* retails for \$89.00. Upgrades from previous versions are \$30.00 plus \$5.00 shipping and handling. For more information contact The Morgan Davis Group, 10079 Nuerto Lane, Rancho San Diego, Calif. 91977-7132, 619-670-0563, 619-670-9643 (fax), 619-670-5379 (bbs).

More evidence that the Apple II really is alive and well, contrary to numerous obituaries: Softdisk, Inc. is actively looking for additional programmers for their Apple II group. Qualifications include proficiency in 8-bit assembly and/or the desktop environment. Knowledge of the Macintosh would not be held against you. <grin> Interested parties who would be willing to relocate to Shreveport, LA should send their resumes (including languages and environments) to Lee Golden, Softdisk Inc., PO Box 30008, Shreveport, LA 71130-0008, Fax resume to: 318-424-0174, America Online: Bryan Zak, AppleLink: SOFTDISK, GENIE: SOFTDISK.INC, internet: soft-disk@applelink.apple.com.

In honor of their 20,000th upload, the A2 Roundtable on GENIE gave away an armful of prizes to a very surprised Tom Smith of Willowdale, Ontario. Mr. Smith was the first person in to upload a file after the 20,000th file had been contributed. The promotion was not announced ahead of time but the prizes were solicited in advance from just about every prominent Apple II company in existence. There are still approximately 16 prizes to be given away in smaller lots during selected Real Time Conferences on the A2 RoundTable.

HyperStudio junkies take note. The first annual HyperStudio Festival is scheduled for July 8-10, 1993 in San Diego, Calif. Sponsored by Roger Wagner Publishing (who else?), it is being billed as a combination of hands-on workshops (for both the Apple IIgs and the Macintosh versions) and hourly conferences empowering the user to get the most out of multimedia with their computers. Festival attendees will have the opportunity to explore all sorts of hypermedia hardware including scanners, video digitizers, laser disk players, still video cameras, MIDI instruments, and VCRs. For further information contact Roger Wagner Publishing at 1050 Pioneer Way Suite P, El Cajon, Calif. 92020, 800-421-6526, 619-442-0525 (fax).

The Turbo IDE card is a high-speed DMA hard disk controller for the Apple II line of computers. This card will allow IIgs and IIe users to make use the IDE type of hard drive as opposed to those with a SCSI interface. Its designers claim that it is fully compatible with all DMA capable hardware such as the Zip GSX, Transwarp GS, PC Transporter, and the usual range of RAM cards. It supports up to 8 partitions on one drive with 32 megabytes per partition. It carries a one year guarantee and free software updates (utilities and ROM) via GENIE. We don't have an IDE drive inhouse, so it was impossible for us to test it. For more information contact Joachim Lange, Schoenstrasse 80a, 8000 Muenchen 90, Germany, telephone: 49-89-6505150 (19.00-23.00 CET), GENIE: J.Lange7.-edr



Ask (or tell) Uncle DOS

Corrections and Amplifications

In last month's issue we mistakenly attributed the upcoming **HardPressed** disk compression utility to Seven Hills Software. We all know that WestCode and programmer Andy McFadden are the ones to (evidenced in the November 1992 issue of **A2-Central**) eventually thank for this one. Contact them at 15050 Avenue of Science, Suite 112, San Diego, Calif. 92128, 800-448-4250.

Alternative drives

I was seriously looking into buying a SyQuest drive, and I've been looking at features/price of them vs. Flopticals for the past couple months or so. I'd almost made a final decision, when I started reading and hearing more reports about the Flopticals.

Right now, SyQuest technology is very "mature." It's been around quite a while, and it's pretty reliable. One drawback to this, however, is that SyQuest prices are not likely to drop appreciably in the near future. They're about as low as they're going to get for a little while. On the other hand, while Flopticals are new and unproven, their current, just-introduced prices are equal or slightly less than the prices of 44 Meg SyQuest drives. Also, there is a lot of excitement about Flopticals, and I'm seeing more and more people buying them. The media cost of Flopticals is also a lot lower than that of SyQuest cartridges (about \$1.15/Meg vs. \$1.55/Meg — that adds up, when buying many megabytes), and that's just the introductory prices. Right now, the cost of Floptical media is actually in between the cost of name-brand 800K disks (about \$1.50/Meg) and bulk 800K disks (about \$.60/Meg). I expect that Floptical media prices will see a drop similar to the one that 800K disks have had since they were introduced. The Floptical drives should also see a similar drop eventually.

So, while I was almost decided on buying a SyQuest drive, I've now decided to wait a few months, to see where these Flopticals are going. I wouldn't be surprised if they got to be quite popular. Sure, the 128 Meg 3.5" Magneto-Optical drives are coming, and at the cost/Meg ratio they're even better than 800K floppies. The problem is that they're already a bit more mature than Flopticals, so their prices are going to be dropping a bit slower (the hardware, anyway — their media is really cheap to make, comparatively). I also expect that they're always

going to be at least twice as expensive as a Floptical drive (they're about three times as expensive, right now), and I just don't have that much money. They're also probably not going to become as commonplace in the Apple II world as Flopticals or SyQuests. I know maybe a dozen other Apple II users and developers who have SyQuests. I already know three or four people that have Flopticals (only counting Richard Bennett and none of the other Australians who bought Flopticals to Kfest '92). I expect that Flopticals will become commonplace enough that media exchange between Apple II users who have them will become as common and feasible, if not more so, as media exchange between SyQuest users right now, within just a few months. The availability of the AE Floptical drive actually quite promotes this, if only in making Apple II users more aware of and interested in Floptical technology.

Actually, the Apple FDHD 1.4M disks are formatted as MFM, the same as MS-DOS 1.4M disks. So the Floptical **can** read and write Apple (Mac and ProDOS) 1.4M disks. It just can't read Apple 800K disks, since those are formatted as GCR. (It won't work with the earlier Applied Engineering 1.6M disks, either. From what I understand, AE has stopped making those versions of its drives, in favour of the standard 1.4M.)

I've actually seen a bit more excitement about Floptical drives in the MS-DOS world than in the Apple (Mac) world. I would guess that it's because the Mac world has already adopted SyQuests as a "standard" mass storage removable media, while the MS-DOS world doesn't really have a standard, yet. I'd really hate to see the two worlds standardize on two different formats, in the long run, now.

One other consideration that came to light, recently, is that SyQuest has announced a new version of its 88M drive, which can now **write** to 44M cartridges, as well as read them, like before. However, it cannot **format** those 44M carts. So, using the new 88M drive you need to either buy the 44M carts you're going to use pre-formatted (I don't know if that's available) or find someone else with a 44M SyQuest to format them for you. It takes about 3 times as long for this new drive to write to 44M cartridges as to 88M cartridges. That greatly limits its usefulness, in my mind, for doing serious exchange of data (I would want to keep my own data on 88M carts, but trade with other people on 44M carts, as 44M drives are MUCH more common). So in all, to the new SyQuest 88M drive the 44M carts are still essentially a foreign medium that is only marginally supported.

I wonder whether M-O drives actually read CD-ROMs and/or Kodak Photo CDs? If not, how long is it likely to be before they can? A single 5.25 M-O drive that could use CD-ROMs and Photo CDs as well would be very attractive, though probably way out of my price range.

Lunatic E'Sex
Palo Alto, Calif.

InSyde secrets

I have a problem with my Cirtech SCSI controller card. It has a novel method of accessing disk space beyond the first 32 meg by running DIVER.SYSTEM when you boot. This installs slot 1 drive 1 as the remaining 8 meg on my hard disk. Unfortunately, when AppleWorks 3.0 attempts to access this drive it crashes into the monitor. If I do not run DRIVER.SYSTEM, AW 3.0 has no problems (but of course, I lose 8 meg of disk). Apple recommends against remapping (except slot 2 → slot 5 and the Smartport remapping found in ProDOS 8 version 2.01) but I can see no other way of accessing the rest of my drive.

I also have problems with my Sequential Systems 1 meg card in my PAL IIe. The problem is that the first 7 pixels on the first scan line of 80 column text, double lores and double hires do not come from aux bank 0 location \$800 (or \$2000 for hires) but instead come from aux bank 3 location \$800/\$2000. This usually is all zeros resulting in a black corner. If I save information to the RAMdisk it eventually reaches bank 3 and puts a random bit pattern in the top left corner. I have still not heard back from Sequential Systems since September 21 when they faxed me that they were investigating my problem and suggested that memory refresh was intimately tied to screen refresh and suspected a problem with PAL. Has Resource Central sold any Meg-80z cards to PAL IIe users and have you heard of any problems?

What method does AppleWorks 3.0 use to access the Auxslot RAM? I am using the RAM disk driver that came with Prosel 8 rather than the one that is supplied with the Sequential System card as it uses slot 4 and removes my hard disk from the drive table. When I do a catalog of /RAM after quitting from AppleWorks 3.0, the RAM disk is in a mess.

David Wilson
Wollongong, Australia

The good news is that the Cirtech InSyder SCSI interface will support ProDOS 8 2.0's remapping scheme for multiple volumes. ProDOS no longer has to be patched to use more than two volumes from ProDOS 8.

For this to work, you need to set the jumpers on the InSyder for "P" off and "F" on. This configures the card to appear as a SmartPort interface; this should fix the problems.

Also remember to remove Driver.System; having two utilities trying to remap the same volumes will cause problems. (And may be causing the crashes.)

We weren't expecting the PAL IIe RAM problems with the Sequential Card: We've sold only a handful of the cards and you're the only PAL customer as far as I can see. If the problem can't be resolved and you need to return the card for credit that's fine; let us know that you're returning the card because of a problem that wasn't resolved so we can credit you properly.

Finally, AppleWorks 3.0 versus the RAM disk: the general problem is that with third-

party drivers (as are used for 11e auxiliary slot memory RAM disks) AppleWorks doesn't know about the RAM disk's memory use and the RAM disk software doesn't intrinsically see AppleWorks's use of memory, so each of the two has to be patched to avoid using banks of memory that the other does.

Specifically for your case, ProSel includes a utility to limit the memory use of the RAM disk, but its method is incompatible with Sequential's patches to limit AppleWorks's desktop. If the major problem is a slot/drive clash, it's much easier to modify the Sequential software to put the /MEG volume in a different place than it is to rewrite the memory management sections of ProSel's RAM.Drv or AppleWorks 3.0 to make them compatible. With /MEG in a usable location, you can use the compatible Sequential AppleWorks patches.

I'm assuming that AL.SYSTEM (the autoloader utility) is what you're normally using. Try the following patch on a copy (never modify the original; you may need it later in pristine form!) where "S" is the desired slot and "D" is the desired drive:

```
BLOAD AL.SYSTEM,TSYS,AS2000
POKE 10898,(S+(D-1)*8)*16 : new slot/drive to check
POKE 10910,(S+(D-1)*8)*16+8 : new slot/drive to use
POKE 10917,(16*D)+(2*S) : 1st byte of driver vector
POKE 10922,(16*D)+(2*S)+1 : 2nd byte of driver vector
BSAVE AL.SYSTEM,TSYS,AS2000
```

The first POKE changes the slot/drive setting that will be checked for an existing entry in the ProDOS (drive) devices table. The original value for AL.System is 64 (\$40) for slot 4, drive 1. (Bitwise, the byte value is in the form DSSS0000, so the value for the second drive in slot 3 would be \$B0.)

The second POKE inserts the new device ID we plan to use in the same form. We add "8" to set bit 3 of device ID byte to identify the new device as a RAM disk. (Under current conventions, the high-order bit of the lower nibble generically indicates a RAM disk. Apple uses "F" for the lower nibble to indicate an Apple-brand RAM disk.)

The third and fourth POKES change the address that the RAM disk driver vector is assigned to. Initially, the vector for slot 4, drive 1 is used, and its address is at \$BF18-\$BF19. The drive table starts at \$BF00 but we actually only need to POKE two one-byte values for the low-order bytes of the table address (initially \$18 and \$19 respectively); we use a bit of math to calculate the new location for the vector from the drive and slot values.

I tried this patch using slot 2, drive 1 with the Sequential-patched AppleWorks and everything seems to work. You can use a different slot/drive; you'd just want to pick something that seems to be habitually "empty" since the slot/drive to use is hardcoded into the program. (It'd be easy to write a BASIC program to make the patches if manually changing the values becomes tedious.)

If you want to make similar patches to Launcher.System on the diagnostics side of the disk, the same code starts at \$1200 bytes into the file, which means if you loaded Launcher.System at \$2000 the patches should be:

```
BLOAD LAUNCHER.SYSTEM,TSYS,AS2000
POKE 12902,(S+(D-1)*8)*16 : new slot/drive to check
POKE 12914,(S+(D-1)*8)*16+8 : new slot/drive to use
POKE 12921,(16*D)+(2*S) : 1st byte of driver vector
POKE 12926,(16*D)+(2*S)+1 : 2nd byte of driver vector
BSAVE LAUNCHER.SYSTEM,TSYS,AS2000
```

Be sure to patch a copy and try installing the RAM disk from it to see if /MEG shows up in the right slot. If all goes well, go ahead and patch up AppleWorks and you should be up and running.—DJD

(r)Ejected

The eject mechanism on my Apple 3.5 disk drive has stopped working. It does not work when I press the eject button or when software (i.e. the Finder) attempts to eject the disk. I have to eject the disk with a paper clip.

I would rather not resort to bringing the drive in for servicing because I need to use it frequently. Can you please tell me how to fix the mechanism?

Raymond Koc
San Francisco, Calif.

The eject motor is shot. Unfortunately, Apple exchanges entire innards, not parts like the motor. But if you have a screwdriver and patience you can swap the innards yourself.

Last time in A2-Central-on-disk:

Directory: /A2.ON.DISK.9301/

Filename	Blocks	Description
General.Stuff	1	
.Andover.Kendall	13	Monospaced fonts
.Chg.A.File.4.16	54	File repair/convert util
.ProBasic.1.1	115	Programming environment
.ProBasicLibrary	81	Extras-Library #2
Iigs.Stuff	1	
.HyperStudioDemo	492	Version 3.1i runtime
.Letter.Slide	705	Educational shareware

Last time in Script-Central

Directory: /ScriptC.Jan93/

Script.Cntrl.10	932	The illustrious FE
ArtWerk2.4	157	More clip art for you
Genealogy	300	Trace family tree
Biorhythms	54	Pretty raw--fix it up
Hypersonic2.4	112	Use in your stacks
TicTacToe2.4	42	Hyperized version
CompView	68	View pictures in Dr. Who
TheLounge	263	New bookmark feature
CursorAnims2.4	28	25 new cursors
SysSnds.NCmd	76	Play rSounds within stack
Companions	261	Final piece for Dr. Who
election92	577	Results of pres election

Last time in TimeOut-Central

Directory: /TimeOut.Cent.14/

GENIE	1	This stuff from GENIE
.U4.CMS	3	Docs for a help screen
.FILMS	479	Movie directory database
.RTC.BRANDT.UMA	26	RTC with Randy
BUG.CITY	5	Documented bugs
TO.U4.CMS	19	
DAN.CRUTCHER	1	
.MENUFILES.MACRO	17	Dan's MenuFile program
JOHN.TEGELAAR	3	TONS of TAPL programs
WALLY.BRADFORD	1	
.HEXDEC.40	6	HexDec lister
.SEARCH.40	9	Search macro
CLEAR.NIGHT	1	From Dan Verkade
.SF.DBOUBLER	3	Patches for SuperFonts
.RW.DBOUBLER	3	Patches for ReportWriter
JEM.SOFTWARE	1	Convert \$ amount to
.DOLLARTEXTMACRO	8	English text
NELKEN.SERIES	1	UltraMacros 4 series
.U4.3	73	
.U4.D	11	
.ULTRA.NOTES	12	
.U4.4	37	
.U4.E	8	
.SS.SAMPLE.4	11	
.U4.5	28	
.U4.6	18	

A2-Central™

© Copyright 1993 by
Resource Central, Inc.

Most rights reserved. All programs published in A2-Central are public domain and may be copied and distributed without charge. Apple user groups and significant others may obtain permission to reprint articles from time to time by specific written request.

Publisher: **Tom Weishaar**
Editor: **Ellen Rosenberg**
with help from: **Denise Cameron**, **Dennis Doms**, **Sally Dwyer**,
Dean Esmay, **Peter Haxton**, **Mary Johnson**,
Jeff Neuer, **Jan Weishaar**

A2-Central—titled Open-Apple through January, 1989—has been published monthly since January 1985. World-wide prices (in U.S. dollars; airmail delivery included at no additional charge): \$34 for 1 year; \$60 for 2 years; \$84 for 3 years. All back issues are currently available for \$2 each; bound, indexed editions of our first six volumes are \$14.95 each. Volumes end with the January issue; an index for the prior volume is included with the February issue.

The full text of each issue of A2-Central is available on 3.5 disks, along with a selection of the best new public domain and shareware files and programs, for \$90 a year (newsletter and disk combined). Single disks are \$10. Please send all correspondence to:

A2-Central
P.O. Box 11250
Overland Park, Kansas 66207 U.S.A.

A2-Central is sold in an unprotected format for your convenience. You are encouraged to make back-up archival copies or easy-to-read enlarged copies for your own use without charge. You may also copy A2-Central for distribution to others. The distribution fee is 20 cents per page per copy distributed.

WARRANTY AND LIMITATION OF LIABILITY. We warrant that most of the information in A2-Central is useful and correct, although drivel and mistakes are included from time to time, usually unintentionally. Unsatisfied subscribers may cancel their subscription at any time and receive a full refund of their last subscription payment. The unfilled portion of any paid subscription will be refunded even to satisfied subscribers upon request. OUR LIABILITY FOR ERRORS AND OMISSIONS IS LIMITED TO THIS PUBLICATION'S PURCHASE PRICE. In no case shall our company or our contributors be liable for any incidental or consequential damages, nor for ANY damages in excess of the fees paid by a subscriber.

ISSN 0885-4017 Genie mail: A2-CENTRAL