

# PAGINATOR

R F C / Sirius  
212 Westbrook Drive  
Raleigh, NC 27609

## INSTALLING THE PAGINATOR ON THE APPLE II\* COMPUTER

NOTES: Installation of the PAGINATOR includes removal of integrated circuit chips and insertion of integrated circuit chips in sockets. You should use appropriate removal and insertion tools to avoid bending or breaking the leads on the chips. You should be aware that integrated circuit chips have slots, marks, or holes in one end to identify it as the end which has pin number 1. Most sockets also have some means of identifying pin 1. The sockets used on the PAGINATOR have a small indentation on the end for this purpose. Locations given in the instructions below (i.e., K1) refer to the grid coordinates on the APPLE II board as shown in the APPLE II REFERENCE MANUAL, January 1978, page 123.

SUMMARY: The PAGINATOR is composed of a main board in the shape of a "C" and an attached socket. The main board replaces the 74LS257 chip at location J1; The chip which was removed is then plugged into the empty socket on the main board. The attached socket replaces the 74LS259 (or the 9334) at F14; The chip which was removed here is then plugged back into the empty socket attached to the PAGINATOR. After all this has been completed, a checkout routine will verify proper operation, and a section on programming considerations will explain how to use the PAGINATOR to best advantage.

PREPARATION: You will need good lighting, an IC removal tool and an IC insertion tool.

DETAILED INSTRUCTIONS: (See diagram on page 3.)

1. Turn off the APPLE II if it's on, and unplug it from the electrical outlet.
2. Remove the top cover from the APPLE II.
3. To allow better access, unplug the power supply plug from the socket at location K1 on the APPLE II circuit board (you must squeeze the locking tabs on the plug front and back).
4. Find the pin 1 slot or marker on the front end of the 74LS257 chip at location J1 just in front of the power plug at K1. Using an IC removal tool, remove the 74LS257 chip from its socket.
5. Using an IC insertion tool, plug the 74LS257 into the empty socket on the main board of the PAGINATOR. If you hold the PAGINATOR so that the copyright is going up the left edge (the board will roughly approximate a "C"), pin 1 is toward the bottom edge. There is a small indentation on that end of the socket, slightly to the right of center.
6. Plug the PAGINATOR into the empty socket from which the 74LS257 was removed. Be very careful to align the pins on the PAGINATOR with the holes in the socket. The cutout on the right side of the PAGINATOR allows visual alignment, but you may need to remove any cards plugged into the first few peripheral slots in order to see well. You will also need good lighting, but it is very important to avoid misalignment. When the PAGINATOR is in place, the 74LS257 will be in approximately the same place and will have approximately the same orientation as it had before it was removed.

7. Replace any cards removed from the peripheral slots in step 6.
8. Replace the power supply plug into the socket at K1 on the APPLE II board.
9. Remove the 74LS259 or 9334 from socket F14 (slightly left and forward from the COLOR TRIM adjustment) after noting the location of the pin 1 slot or mark.
10. Plug the PAGINATOR socket into the empty socket at F14 (wires go in front).
11. Plug the 74LS259 or 9334 into the PAGINATOR socket - pin 1 in front.
12. Replace the top cover and plug the APPLE into the electrical outlet. Do not turn it on.
13. Turn on the video monitor or TV attached to the APPLE II and allow it to warm up if necessary before proceeding any further. When you turn on the power to the APPLE II, be prepared to turn it back off if you do not see a cursor and a prompt as usual. Now turn on the APPLE II and check for the prompt. If not present along with the cursor, turn off power immediately and recheck all work.

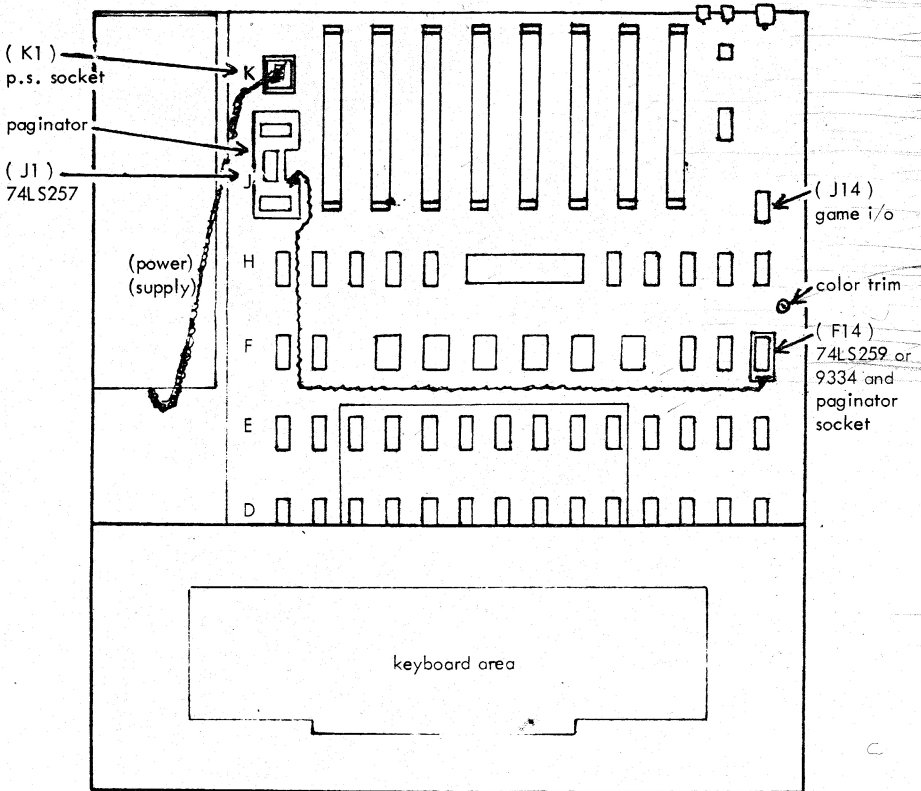
CHECKOUT: Verify proper operation by typing in the monitor commands given below. The asterisk at the beginning is the monitor's prompt, you must type in the command which follows it and give a carriage return. The computer should react as explained in the comments.

## YOUR DIALOG WITH COMPUTER

## COMMENTS

-----	-----
*0800:0 N0801<0800.BFFEM	Clears memory locations \$0800-BFFF to zeros
*C053:0 NC054:0 NC057:0 NC050:0	Sets switches for Mixed Mode, Page 1, Hires, Graphics
*C054:0 NC058:0 NC05A:0	Selects page 1(A) at 8K-16K It should be blank now
*2000:FF	Puts one short line at top left if all is working properly
*C054:0 NC059:0 NC05A:0	Selects page 1(B) at 24K-32K *It should be blank now
*6000:FF 00 FF	Puts two short lines at top left if all is working properly
*C054:0 NC058:0 NC05B:0	Selects page 1(C) at 40K-48K It should be blank now
*A000:FF 00 FF 00 FF	Three short lines at top left if all is working properly
*4000:FF 00 FF 00 FF 00 FF	Set up page 2 before going there because we will lose our text
*C055:0 NC052:0	Go to page 2/no text - we should see 4 short lines at top left
*C058:0 NC05A:0 NC054:0 NC051:0	You will not see this until you type the return. This restores page 1, text mode and ends the test.

View of APPLE II with top cover removed



## PROGRAMMING CONSIDERATIONS:

1. Page selection for the new pages is controlled by two "annunciator" signals in the APPLE II. Since APPLE does not use these signals, the original monitor rom does not initialize the signals to any particular value (on or off). As a result, the HIRES page 1 which will be selected when you power up the APPLE II is not predictable. If you have the original monitor rom, you will have to put code in your disk "HELLO" program to select the page you want (or simply poke in the appropriate values from the keyboard if you desire). If you have the newer rom however (the "autostart" rom), it will initialize the annunciators used by the PAGINATOR to be both off whenever you turn on power or whenever you push the RESET key. This will select the original location (8K-16K) for page 1.



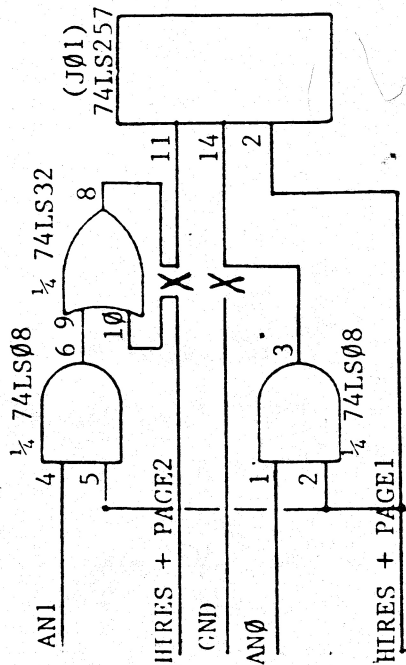


Figure 1 shows the new logic added with the PAGINATOR. The "X" marks indicate places where existing signals were interrupted. (Note: the diagram on page 146 of the APPLE II Reference Manual has a few errors; one of those is the pin number for the HIRES+PAGE1 signal. (says 5, is 2)

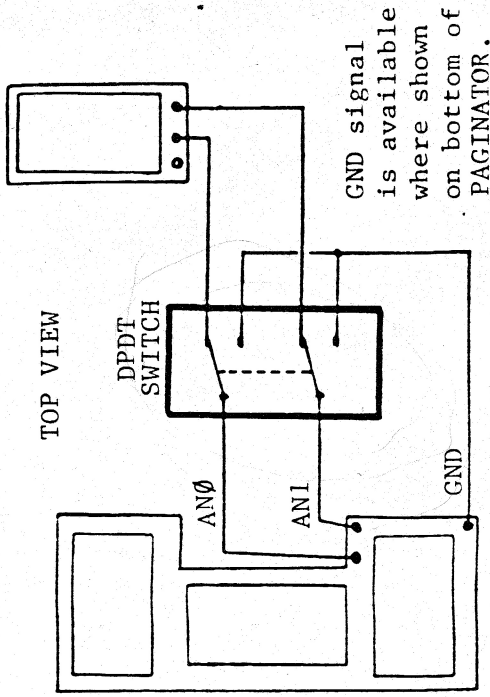


Figure 2 shows how a double-pole, double-throw switch can be added to the circuit. This DPDT switch will allow the PAGINATOR to be completely disabled. There is no need for this unless you have to use HIRES page1 and AN0 or AN1 at the same time.

2. The PAGINATOR uses all of the regular HIRES graphics switches ("POKES") plus two extra ones AN0 and AN1 to select which page 1 memory area will be used. AN0 and AN1 are not normally used in any APPLE software, but may be used by other "add-on" devices or applications. AN0 and AN1 may still be used in any manner desired EXCEPT when you are using page 1 HIRES graphics. The following illustrates the use of AN0 and AN1 to select the desired memory area. Notice that anytime you address location SC058, you will clear AN0 (turn it off); addressing location SC059 will set it (turn it on). Location SC05A will clear AN1; and location SC05B sets it. In BASIC, the decimal equivalent of the addresses are -16296, -16295, -16294, and -16293 respectively.

FROM BASIC	FROM MONITOR
> POKE -16296,0	*C058:0
> POKE -16294,0	*C05A:0
AN0 clear / AN1 clear = page 1(A) at 8K-16K (\$2000-4000)	
> POKE -16295,0	*C059:0
> POKE -16294,0	*C05A:0
AN0 set / AN1 clear = page 1(B) at 24K-32K (\$6000-8000)	
> POKE -16296,0	*C058:0
> POKE -16293,0	*C05B:0
AN0 clear / AN1 set = page 1(C) at 40K-48K (\$A000-C000)	
> POKE -16295,0	*C059:0
> POKE -16293,0	*C05B:0
AN0 set / AN1 set = page 1(D) (results unspecified)	

3. The PAGINATOR provides the means for the hardware to use different parts of memory as the area displayed on your screen when you request HIRES graphics page 1. The PAGINATOR does not make the software place the information in that part of your memory however. This step can be accomplished in one of several ways that will be described in the following paragraphs. First, it will help to understand how the software knows where to put the data. In APPLESOFT, the HGR command puts a constant into page zero (the value of the constant is \$20 00) as the base address for the screen data, then the data is moved into memory indirectly using the page zero constant as a reference in order to clear the screen and later to use as a reference for any other data to be displayed in page 1 HIRES graphics. The INTEGER BASIC "INIT" routine does a similar thing, but uses other addresses outside of page zero to store the constant data (\$20 00).

If you are using RAM versions of APPLESOFT (either cassette, disk, or language card) you can easily "POKE" a new constant into the routines before you use them and, except for selecting the matching page 1 in hardware with AN0 and AN1, there is nothing else to do to your programs. This is also true for all RAM based routines for INTEGER BASIC as provided by APPLE COMPUTER, INC. Unfortunately you cannot alter the ROM based versions as easily, so all "HGR" commands in APPLESOFT and all "CALL INIT" (or similar statement) in the INTEGER BASIC routines must be removed entirely from your program and replaced with subroutines to accomplish the same things. The subroutines below will place appropriate constants (\$20 00, \$60 00, or \$A0 00) into the right place in memory,

\*APPLESOFT is a trademark of Apple Computer, Inc.

clear or set AN0 and AN1 so that the hardware will use the same address, and then call the "HGR" or "INIT" routine at a point after the base address would have been set by the routine. As an added feature, the routines provided will allow you to set the mixed text/graphics mode at the same time.

-----  
 INTEGER BASIC "BEFORE"

10 INIT = 2048

370 CALL INIT

560 END

-----  
 INTEGER BASIC "AFTER"

10 INIT = 2048

370 PG = 2: MX = 0: GOSUB 32000

560 END

32000 POKE -16300,0: REM PAGE 1  
 32001 POKE PG MOD 2 -16296,0: REM AN0  
 32002 POKE PG/2 -16294,0: REM AN1  
 32003 POKE -16301-MX,0: REM MIXED  
 32004 POKE -16297,0: REM HIRES ON  
 32005 POKE 768+PEEK(INIT+3),32+PG\*64  
 32006 CALL INIT+11: REM GO CLEAR IT  
 32007 RETURN

-----  
 APPLESOFT II "BEFORE"

370 HGR

560 END

-----  
 APPLESOFT II "AFTER"

370 PG = 2: MX = 0: GOSUB 32000

560 END

32000 POKE -16300,0: REM PAGE 1  
 32001 POKE PG-INT(PG/2)\*2-16296,0: REM AN0  
 32002 POKE INT(PG/2)-16294,0: REM AN1  
 32003 POKE -16301-MX,0: REM MIXED  
 32004 POKE -16297,0: REM HIRES ON  
 32005 POKE 230,32+PG\*64: REM ADDR HI BYTE  
 32006 CALL 62447-PEEK(-3085)/253\*51204  
 32007 RETURN

NOTES ABOUT THE SUBROUTINES:

PG selects the page 1 memory area as follows:

PG = 0 for page 1(A) 8K-16K (\$2000-4000)

PG = 1 for page 1(B) 24K-32K (\$6000-8000)

PG = 2 for page 1(C) 40K-48K (\$A000-C000)

all other values for PG are illegal - results unspecified

MX selects mixed text/graphics on or off as follows:

MX = 0 for mixed text/graphics on (normal)

MX = 1 for mixed text/graphics off (no text)

The complete address for the beginning of the HIRES data area can be calculated from PG as follows (for "BLOAD'ing" screen images):

BASE ADDRESS = (8192+SGN(PG)\*16384)\*SGN(3-PG\*2)

Since the subroutines on the preceding page can be used with all revs of both RAM and ROM routines for HIRES graphics, you should use them whenever you must be compatible with systems other than your own. In fact, because they allow for setting the mode in a simple way (mixed text/graphics on or off) and because they adjust both the hardware switches (AN0 and AN1) and the software base address whenever the "PG" variable changes, many people prefer to use the subroutines for everything. This is especially true if you are using two or three of the available page 1 areas in the same program. In other cases though it is easier to set the switches and base address directly and not use the subroutines. This might be appropriate for many existing programs when you do not want to search out all of the "HGR" statements (in APPLESOFT) or the "CALL INIT" statements (in INTEGER BASIC). You can do this ONLY if your HIRES graphics routines are in RAM versions (either from cassette, disk, or the language card). HIRES routines which are in ROM will clear memory from 8K to 16K to all zeros everytime "HGR" or "CALL INIT" is used, therefore the following can not be used with ROM versions.

ALTERING APPLESOFT IN RAM AT RUNTIME is accomplished by POKE'ing a new value into the routines for use as a constant by them. For all RAM versions except the one for the language card, first make sure that APPLESOFT is in RAM, second select the correct hardware page 1 with AN0 and AN1 as shown on page 4, third fix the base address with a POKE 11231, xx where xx has one of the following values to match the hardware page:

xx = 32 for page 1 (A) at 08K to 16K  
 xx = 96 for page 1 (B) at 24K to 32K  
 xx =160 for page 1 (C) at 40K to 48K

Note that anytime you reload APPLESOFT (as with a "FP" command) you must redo the poke.

ALTERING APPLESOFT IN LANGUAGE CARD is similar to the above for APPLESOFT in RAM, but also includes steps for "write enable" to the card and then "write protecting" it at the end. First, select the hardware page 1 as shown on page 4 using AN0 and AN1, second write enable the card with TWO PEEKS to address 49283, third POKE 62435, xx where xx is a value from the table above to match the hardware page selected previously, finally PEEK location 49280 to "write protect" the card again. If you want to change the value on the diskette itself, it is at byte offset 483 in block offset 23 (or block offset 17 within the APPLESOFT file). All numbers decimal - first block & byte is zero.

MAKING A PERMANENT ALTERATION TO APPLESOFT IN RAM for a particular purpose may be helpful in some cases (note: see paragraph above for language card version). To do this, get into INTEGER BASIC, do a HIMEM:16384, then LOAD APPLESOFT from disk or cassette (do not use "FP"), then POKE 15328, xx where xx is a value from the table above, finally SAVE APPLESOFT. Remember that when you use this copy of APPLESOFT, you must set AN0 and AN1 to agree with the page 1 area you have put in the software.

ALTERING INTEGER BASIC HIRES ROUTINES IN RAM is similar to the above procedures. First load the routines, then set INIT=zzzz (usually 2048), third POKE INIT+1, xx where xx is a value from the table above, then use or save the routines (note: these routines are usually found from 2048-3071 in memory - \$0800-\$0BFF)

4. A major consideration when using HIRES graphics is to insure that nothing else is trying to use the same area of memory at the same time (i.e., programs or variables, etc.,). This means that you may need to set HIMEM or LOMEM sometimes to control memory usage by your programs. The PAGINATOR was designed especially to allow the HIRES memory page to be placed at the top of memory in 32K and 48K systems. Without the PAGINATOR,

the HIRES memory divides the user area into two blocks separated by the 8K used by the HIRES data. The PAGINATOR allows us to place the HIRES memory at the top of a 32K or 48K system and have a contiguous area of memory for user space. In systems which do not have a disk, all that is needed is to set HIMEM at 24K or 40K respectively to prevent programs from overwriting the HIRES data and vice-versa. If your system does have a disk, the DOS 3.2 normally resides at the top of memory. If we left it there, we could not use the area for HIRES. Fortunately, the DOS does not actually load at the top of memory, it RELOCATES ITSELF after it is "booted". Before relocation, the DOS "sizes" memory by looking for the most memory that it expects anyone to have (i.e., 48K). If it finds any memory at that address, it then relocates there; if it does not find any memory there, it lowers its expectations (and the value of a variable) and tries again. This continues until some RAM memory is located and then the DOS relocates itself to that area of memory. If we alter the value at which the DOS begins looking for RAM, we alter the maximum address to which the DOS can relocate itself. The "RELOCATE 3.2" program below is used to accomplish this. Briefly, the program asks how much memory you want the DOS to be able to find. Then it saves the value in location zero and loads the "UPDATE 3.2" program provided by APPLE. Before running the "UPDATE 3.2" program however, it POKES an RTS instruction into it at a point after "UPDATE 3.2" has loaded the DOS image. The RTS gives control back to the "RELOCATE 3.2" program which then POKES the desired value into the DOS image rather than the 48K value originally in the image. Finally the "RELOCATE 3.2" program does a little housekeeping to restore the "UPDATE 3.2" program (where it had POKEd the RTS) and an instruction to make sure the CARRY FLAG is cleared as expected, then "RELOCATE 3.2" re-enters the "UPDATE 3.2" program approximately where it left it with the RTS instruction. As you can see, "RELOCATE 3.2" requires that "UPDATE 3.2" be resident on the same disk to work. One final note, the PAGINATOR does not affect page 2 HIRES graphics at all and does not prevent the use of "HGR2" wherever you would otherwise use it. The PAGINATOR simply adds extra page 1 areas so that a 48K system now has three page 1 areas and one page 2 area.

#### INTEGER BASIC LISTING FOR RELOCATE 3.2

```

10 REM RELOCATE 3.2
20 REM COPYRIGHT 1979
30 REM JAMES P REECE
40 REM ALL RIGHTS RESERVED
900 D$="": REM CTRL-D
910 PRINT "HIGHEST MEMORY ADDRESS"
920 PRINT "USED BY DOS 3.2 (IN K)"
930 INPUT "(RANGE 20 TO 64) ",K
940 IF K<20 THEN 930
950 IF K>64 THEN 930
960 POKE 0,K*4-1: REM SAVE K
970 PRINT D$;"BLOAD UPDATE 3.2"
980 POKE 2093,96: REM RTS INSTR
990 CALL 2048: REM GOTO UPDATE 3.2
1000 POKE 2092,24: REM CLC INSTR
1010 POKE 2093,173: REM ORIG VALUE
1020 POKE 7172,PEEK(0): REM K
1030 CALL 2092: REM GOTO UPDATE 3.2
1040 END: REM JUST FOR FUN

```

or "UPDATE 3.2.1" if appropriate

for DOS 3.2.1 poke 6916, peek(0) instead