Senior PROM //e, //c Version 3.0 Documentation
Copyright 1986, 1987 by Cutting Edge Enterprises

February 1987

2087O

Table of Contents

Apple usually refers to the Apple II+, //e or //c computer, and is a trademark of Apple Computer, Inc. Merlin is a trademark of Southwestern Data Systems.

Chapter 1: Senior PROM Capabilities and Description.

A Senior PROM is a firmware device designed for use exclusively in the Apple //e and //c computers. It replaces your standard Apple Monitor ROM(s) with a small printed circuitry board. This doubles the amount of ROM (Read Only Memory) in your computer, without consuming a peripheral card slot (on the //e) and without losing compatibility with ANY present or future Apple //e or //c programs.

The Senior PROM provides the following functions in ROM, where they are instantly available to the computer user at any time:

   *   A complete menu of functions available from the Reset key:
       -   Ability to enter the system monitor at any time to move, examine, or alter programs.
       -   Reboot a diskette with destroying memory.
       -   Copy the entire contents of memory to auxilary memory.
       -   Move memory from the "language card" to accessible memory.
       -   Move memory from the auxiliary card to main memory.
       -   Write a "test pattern" into memory to study disk boots.

   *   Enter a complete set of DOS 3.3 RWTS-based utilities:
       -   Initialize diskettes.
       -   Copy diskettes.
       -   Enter ROM-resident Sector Editor and Memory detective.
       -   Disable RWTS error-checking to diagonse disk problems.
       -   Alter DOS recognition bytes to examine protected software.

   *   Non-maskable interrupts fully supported which allows:
       -   Interrupt a program, view or change memory, and restart the program.
       -   Capture to a DOS 3.3 disk the entire 64k of Main RAM.
       -   The ability to re-load and restart a captured program up to 128k from disk.
       -   Release software from its protected disk.
       -   Swap back and forth between two program (using up to 64k each) instantly.

   *   Enhanced monitor capabilities that include:
       -   ASCII display conincident with disassembly or memory display, including 65C02 support.
       -   Full Step and Trace functions to examine, debug and correct assembly language programs.
       -   A "Mini-Assembler" that supports the 65C02
       -   Full support of screen editing commands and Ctrl-S commands
       -   Instant return to RWTS-based menu via Ctrl-C.

Since these functions are available in ROM, they are there when you need them, and are transparent when you don't. The mere flip of a switch and a press of a key will make these utilities appear. Likewise, flip the switch again and press a key and they disappear!

Chapter 2: Senior PROM System Requirements.

Special //e Notes about "Double ROM" and "Single ROM" Logic Boards:

Starting in November 1986, Apple changed their //e logicboards to a single Monitor ROM labeled the "CF ROM" (see installation diagrams, //e figure 2). Before November 1986, all //e logic boards had two monitor ROMs labeled "CD" and "EF" ROMs (see installation diagrams, //e figure 1). When we ship a //e Senior PROM, we ship the "Double ROM" version for pre-November 1986 //e logicboards, unless otherwise specified. If you have the newer "Single ROM" //e logicboard (as in //e installation figure 2), please return your Senior PROM (just the hardware please) for an immediate and free replacement.

The Senior PROM //e Requires the following:

-   An Apple //e, any revision.

and the following hardware is Recommended to use your new Senior PROM //e to its fullest:

-   An Extended 80 Column Card (this gives your system 128k).
-   Two disk drives.

The Senior PROM //c Requires the following:

-   An Apple //c, any revision.

and the following hardware is Recommended to use your new Senior PROM //c to its fullest:

-   A Second 5.25" Disk Drive.

No other hardware or software is required.

UniDisk Compatability on the Apple //c:

Starting in March 1987, the Senior PROM //c comes with the new 32k Apple Unidisk ROM with Apple Talk compatability (Monitor ROM part number 042-0033-A). This is the newest //c Monitor ROM available from Apple.

If you have an older //c (before November 1986) with the UniDisk Senior PROM //c installed, it is not necessary to have your local dealer upgrade your //c for use with the UniDisk 3.5 inch drives. The UniDisk Senior PROM //c has already done this for you.

If you own an older Senior PROM, you can upgrade to the UniDisk Senior PROM. Please contact Cutting Edge for details.

Chapter 3: Hardware Description of Senior PROM.

The Senior PROM //e (Please note the "Single ROM" and "Double ROM" discussion on the previous page). The Senior PROM //e consists of two EPROMs mounted on a small printed circuit board that replaces the existing CD and EF Monitor ROMs (or CF Monitor ROM) on the Apple //e logicboard. These two PROMs are the standard Apple //e monitor ROM code (CD and EF ROM code) and a whole new set of utilities and monitor ROM. These new utilites and monitor ROM are available at any time with a flip of a switch.

Also included on the board is a push button (or momentary toggle switch) NMI (Non Maskable Interrupt) switch, which is fully supported (and debounced), allowing the user to interrupt a program in memory, examine or change the code, and restart it. In addition, the program can be captured to a normal DOS 3.3 disk and re-started at a later time.

The standard Apple ROM code in the Senior PROM is available in either Standard Monitor ROM configuration or the new "Enhanced" //c compatible Monitor ROM configuration for use with the 65C02 and the new character ROM.

Note that the Senior PROM is not a replacement for the //e upgrade kit from your Apple dealers. If you order your Senior PROM with the "enhanced" configuration, it will only work on a //e that has had the upgrade done by your Apple dealer. This is because the new character ROM and a 65C02 are required for the "upgraded" monitor ROMs.

If at some point in the future you update your present //e to Enhanced ROMs, you may return your Senior PROM to Cutting Edge Enterprises for an upgrade. The cost is $10.00 to cover postage and handling.

The Senior PROM //c: The Senior PROM //c consists of two EPROMs mounted on a small printed circuit board that contains all the Senior PROM ROM code. The standard Apple //c Monitor ROM is removed and the Senior PROM //c is plugged directly into the //c motherboard.

Just like the Senior PROM //e, a separate switch board comes out of the Apple //c case and it attached to the exterior of your //c. This switch activates the Senior PROM and the NMI.

Copyright Infringements: By purchasing and using the Senior PROM, you <u>are not</u> infringing any copyrights owned by Apple Computer Incorporated, or any other company. Since the Senior PROM can only be used in an Apple which **YOU ALREADY OWN** and have the legal rights to modify, you have already bought an individual license for the firmware in the Apple's monitor ROMs.

Warranty Information.

Your Senior PROM is warranteed against defects for a period of 90 days. This <u>does not</u> include user abuse or damages incurred by user installation or removal.

Note: Bringing your Apple in for Authorized Service. The Senior PROM <u>will not</u> void your Apple //e or //c service warranty **providing you remove it <u>before</u> bring your Apple in for service,** and providing you have not visually damaged the computer upon installation or removal.

Chapter 4: Installation of the Senior PROM in an Apple //e.

USER NOTE: Please be extremely careful in installing your new Senior PROM. **CUTTING EDGE ENTERPRISES IS NOT RESPONSIBLE FOR ANY DAMAGE INCURRED BY INSTALLATION TO EITHER YOUR COMPUTER OR THE SENIOR PROM.** Each unit is tested carefully at the factory before shipping to ensure it will properly function in your Apple //e computer.

1) Turn off the power switch on the back of your Apple //e computer.

2) Remove any disk drives or monitors from the top of the Apple //e computer case, and generally clear the area.

3) Gently grab the back edge of the lid of your computer, and pull (gently) up.

4) Slide the top lid out of the computer case, away from the keyboard.

5) Refering to //e Installation figures 1 or 2, determine if you have a Single or Double Monitor ROM logic board. Locate the CD and EF ROMs (or the CF ROM) on the Apple //e logicboard.

6) Using a Chip Remover or a small screwdriver, carefully remove both the CD and the EF ROMs (or the CF ROM) from the Apple //e logicboard (it's often helpful to pry gently at both ends of each chip).

7) For both Single and Double ROM logicboards, carefully remove the Senior PROM from its packaging and plug the board into the CD and EF sockets (for double ROM Senior PROMs), or the CF socket (for single ROM Senior PROMs) on the //e logic board. **Position the Senior PROM so the ribbon cable is AWAY from the keyboard** (see //e installation figure 3). Make sure that all the pins on the Senior PROM go into the sockets on the logic board. Press the Senior PROM into the socket(s) gently, but firmly. There should be very little resistance. Failure to follow this step correctly can cause damage to your Senior PROM! Be extremely carefully not to damage the pins on the Senior PROM during installation!

8) Locate to Pin 6 of the 6502 micro-processor chip on the Apple //e logic board. Refer to //e Installation figure 3 to help find the 6502 chip and pin 6. If any peripheral cards are blocking the 6502 chip, remove them first.

9) Carefully push the end of the Micro-Probe (attached to the Senior PROM) so the small metal clip is exposed. Connect this metal clip to pin 6 of the 6502 Microprocessor chip. Make sure the metal contact is not touching any other pins of the 6502 Microprocessor.

10) String the external board with the two switches out the back of the Apple //e case, and attach the board (using the adhesive backed tape) to the **right side** (RESET key side) of the Apple case, preferably close to the reset key. Move the toggle switch to the back (deactivated) position.

11) Recheck steps 7 to 10 and replace any removed peripherial cards and Replace the Apple //e lid and any assorted hardware.

Your Senior PROM is now installed!

Installation of the Senior PROM in an Apple //c.

USER NOTE: Please be extremely careful in installing your new Senior PROM. **CUTTING EDGE ENTERPRISES IS NOT RESPONSIBLE FOR ANY DAMAGE INCURRED BY INSTALLATION TO EITHER YOUR COMPUTER OR THE SENIOR PROM.** Each unit is tested carefully at the factory before shipping to ensure it will properly function in your Apple //c computer.

If your //c has any internal user installed boards (such as a ZRAM), the Senior PROM will not fit inside your //c. For users with this problem, an External Senior PROM //c Adapter is available for $25.00 from Cutting Edge. This mounts the Senior PROM outside of your //c, since there is not ample room inside.

Starting in March 1987, the Senior PROM //c is fully compatable with 3.5" UniDisk drives and Apple-Talk.

Opening the Apple //c.

The Apple //c is unique among the Apple II family because it is a "closed" package. By following the procedure given here, anyone can safely open a //c without voiding the warranty or risking damage to the system.

The //c case is assembled by snapping a top cover onto the base, like one of those "snap-together" toys that were designed to prevent glue-sniffing in the younger generation. Removing the top cover with no damage requires undoing the assembly by following a simple sequence of events. Because the case is made from very flexible plastic, some bending and minor twisting of the pieces is possible to ease the disassembly process. Even if you become careless or impatient during the process, and break off one of the small plastic tabs, the //c will still close securely with no visible evidence of injury. The two halves are held together by tabs located in five areas (see //c Figure 1), and after removing 6 screws the best tool for opening is a small, flat-bladed screwdriver about 1/8" to 3/16" wide. So with a 1/8" to 3/16" flat blade screwdriver and a medium sized phillips screwdriver in hand, you are ready to begin.

1) Turn the power switch to "0" and remove the power cord and any peripherals and cables plugged into your //c.

2) Place your //c on a firm table with the bottom facing up (keyboard down) so the mounting screws are viewable. Referring to //c Figure 0, remove the 6 screws shown with a phillips head screwdriver.

3) Turn the //c over (keyboard up). Holding the screwdriver as shown in //c figure 2, at position 1 in //c figure 1, push down and in on the tab which descends from the top cover until the front of the cover "pops" up as the tab is released. Note: You are prying the bottom part of the case out, and then pushing in with the screwdriver. If you mar the case you are applying too much force in the wrong place. The pop should pop easily, without any force.

4) Turn the //c around and refer to //c Figure 3 for the proper sequence in this area. This part is a little trickier, since the "flaps" at either side of the back are tabbed into both the sides of the bottom cover and the bottom of the top cover (points B, C, D and E in //c Figure 3), and stay with the cover when it is lifted off.

5) Lower the handle ("A" in //c Figure 3), and then gently pry the flaps out and down at all four points (B, C, D, & E) with the screwdriver. The plastic is flexible, and a flap will sometimes snap back while the other is being pried out.

6) When both flaps are out (//c Figure 4), pry out gently at point "F" until the center of the back clears the D-shaped connectors and pops out and up.

7) Referring to //c Figure 5, pry gently from below at points A and B (one inch on either side of the disk latch opening), and the entire top will pop free from the base.

8) Push the keyboard gently towards the disk drive and lift the front up slightly, and then pull the keyboard forward until it lifts free. Fold the keyboard over onto the disk drive to expose the motherboard.

9) Touch the metal case on the disk drive to ground your self to the //c and to discharge any static electricity.

10) Refer to //c Figure 6 for the layout of the Apple //c mother board. Locate the ROM memory chip at location D-19, and carefully remove it with a chip puller or by lifting under each end with a small screwdriver. Using a sharp razor blade or a soldering iron, cut the W1 double triangle shaped jumper so the triangles no longer meet, as shown in figure 6. Also, the jumper at W2 should be connected with the triangles touching (see //c installation figure 6). The best way to do this is using a soldering iron and some solder. The Senior PROM //c will not work unless these jumpers are modified. If these jumpers are not present on your motherboard, no cutting is necessary (refer to figure 6).

11) Identify and mark pin 6 of the 65C02 CPU chip at E-19 (pins are numbered starting from 1).

12) Align the Senior PROM with the arrow pointing to the front of the computer (keyboard end), and carefully insert the pins into the empty socket at D-19. Use extreme care not to bend or break any of the pins – if there is resistance to insertion, lift the board and carefully realign it before inserting.

13) Referring to //c Figure 7, examine the Senior PROM and note if there is a chip present in front of the arrow (labeled "Apple PROM" in the Figure 7). Depending on your revision of the Senior PROM //c, there may or may not be a chip at this location. If no chip is present on the Senior PROM, plug the //c ROM chip you just removed in step 10 into the Senior PROM at this position. Be extremely careful to install the chip with the "notch" towards the volume control, as shown in //c Figure 7, and not to bend any of the pins while pushing the chip into the Senior PROM. If a chip is already present in the Senior PROM at this location, store the chip you removed in step 10 in a safe place.

14) Attach the micro-clip to pin 6 of the 65C02 CPU chip as shown.

15) Dress the switch cable around the board and towards the right rear of the computer, laying it flat over the top of the disk drive. Bring the cable out the back of the computer at point "E" of //c Figure 3.

16) After carefully reviewing the installation, flip the keyboard back over the front of the computer, and gently push it back into position, noting the tabs that slide into the metal disk drive cover. Install the plastic top cover by reversing the "unsnapping" process above, starting at point 5 (//c Figure 1) back to point 1. It may be necessary to leave the rear flap slightly open at point "E" of //c Figure 3 to allow the switch cable to exit the computer. This small opening will cause no problems in operation of the //c.

17) Route the external switch cable underneath the //c and attach the external switch board to the front right side of the //c case. Make sure the keyboard switch is in the "QWERY" position.

Installation in your Apple //c is now complete!

Preventing Problems.

Carefully review installation steps 9 thru 17 making sure the Senior PROM is not installed backwards or incorrectly. Make sure the pins on the Senior PROM go into the motherboard socket without bending or crushing. Do not force the Senior PROM in place. The Senior PROM should plug in easily, without any resistance.

Alternative //c Installation.

It has come to Cutting Edge's attention that many Apple Service Centers use an alternative method for opening an Apple //c for repair. Cutting Edge Enterprises does not endorse this procedure, and will not be responsible for any damage resulting from it. It is presented as an alternative method of opening a //c.

1) Turn the power switch to "0" and remove the power cord and any peripherals and cables plugged into your //c.

2) Place your //c on a firm table with the bottom facing up (keyboard down) so the mounting screws are viewable. Referring to //c Figure 0, remove the 6 screws shown with a phillips head screwdriver.

3) After removing the necessary screws from the //c case, turn the unit back over (keyboard up) and unfold the handle down so the //c is propped at an angle, with the keyboard end resting on the case, and the back connector end resting on the handle.

4) Pivoted the unit on the keyboard end by lifting the back of the unit 2 to 3 inches off of the table. Note the keyboard end of the //c is still touching the table, and the back connector end is 2 to 3 inches off the table.

5) The back of the unit is then "dropped" squarely on the table from this position. This procedure may have to be repeated 1 or 2 times, but generally the //c case will spilt cleanly, and without damage, at the intended seams.

6) Proceed with installation steps 7 through 17, as previously described.

Removing the Senior PROM from your Apple //e or //c

If you ever need to remove the Senior PROM from your Apple, **be extremely careful**. It is very easy to break one (or more) of the pins on the Senior PROM that plug into the logic board.

After disassembling your computer, the best way to remove the Senior PROM is to use a small flat head screw drive, and gently pry the Senior PROM board lose out of the motherboard sockets. When the Senior PROM is the out of the logic board sockets most of the way, use a **gentle** rocking motion to completely remove the board.

Don't forget to remove the micro-probit from pin 6 of the 6502 microprocessor!

If you do break one (or more) of the pins on the Senior PROM that plug into the logic board, the unit must be replaced or repaired. Send the Senior PROM back to Cutting Edge Enterprises with a check for $20 for a replacement or repair. Send First Class U.S. mail only (U.P.S. can not be accepted). Note if you require a change in ROM code (from Standard to Enhanced ROMs), this can be done at the same time for no additional cost.

Once again, Cutting Edge would like to stress that you be **extremely** careful and gentle when removing a Senior PROM as it is easy to damage the unit when removing it.

Turning your Computer on with the Senior PROM and Switch Description.

Notice the two switches attached to the external board (you may also refer to the External Switch Board Diagram). The toggle switch activates and deactives the Senior PROM. The push button (or momentary toggle) switch is the Senior PROM NMI switch. The Senior PROM is **DEACTIVATED** when the toggle switch is pushed **all the way towards the back** of the computer. The Senior PROM is **ACTIVATED** when the toggle switch is pushed **all the way towards the front** of the computer. The **middle or transparent position** keeps the Senior PROM in its previous switch position. In the middle position when the NMI switch is pushed, the Senior PROM automatically becomes active regardless of the switch's previous position (refer to **Using the NMI and Resurrect Programs** for more detail on the middle transparent position). Note that switch position assumes you have installed the external switch board on the right side of the computer case (suggested).

Please ignore the push button (or momentary toggle) NMI switch until you have read the Chapter "Using the NMI and Resurrect Program". **If the switch is pushed at the wrong time it can cause unpredictable results**.

Now that your Senior PROM is installed, and the external switch board is attached to the right side of your computer case and in the back (deactivated) position, turn your computer on. On the //c make sure the keyboard switch is in the "QWERTY" position. Hold down both the Open Apple and Closed Apple keys (they're next to the space bar), and press CTRL - RESET simutaneously. On the //e, this will perform the standard Apple diagnostics, and in a few moments, an "OK" message should appear. On the //c, this is a RAM exerciser in which the hi-res pages will change, and no "OK" message will appear (note: on //c's with a ZRAM, the hi-res pages will change twice, and then an 80 column text screen will appear). This will test the Senior PROM's ROMs, and your installation. If any error message appear, replace the Senior PROM with your standard ROMs and repeat the diagnostics. If the same error is not produced, your Senior PROM is damaged.

Press CTRL - RESET to exit the diagnostics (or //c exerciser), and your Apple should now operate as nothing has changed. If the hi-res page is displayed and the computer acts as it is locked-up, move the Senior PROM toggle switch to the back position and press CTRL - OPEN APPLE - RESET (press all three simultaneously). The Apple should re-boot and the drive should be spinning. Your Apple will act normally until you move the Senior PROM toggle switch to the front position.

Installation Problems (or Don't Panic Yet!)

If your computer is still locked up, regardless of the position of the Senior PROM toggle switch and how many times you hit CTRL - OPEN APPLE - RESET, you may have installed the Senior PROM incorrectly. But please don't panic, you probably have done no damage to your computer!

Check that the Senior PROM was not installed backwards. On the //e, make sure the ribbon cable is exiting the Senior PROM board towards the back of the computer, away from the keyboard. On the //c, the word "Front" and associated arrow should be pointing towards the front (keyboard) of the //c.

If you installed the Senior PROM backwards, or your computer is still not working, re-install your original Apple ROM(s) into the motherboard and test the computer. If the computer works fine with the original ROM(s), send the Senior PROM back to Cutting Edge Enterprises with an explanation of what did not work. ONLY SEND YOUR SENIOR PROM BACK USING FIRST CLASS U.S. MAIL. U.P.S. can not be accepted by Cutting Edge. Depending on the damage done, repair costs range from no charge to $20.00.

Chapter 5:   Senior PROM Reset Functions.

Activate the Senior PROM by flipping the toggle switch towards the back of the computer. Now press CTRL - RESET and you will hear a distinctive "beep". Your Apple //e or //c will seemingly lock up, but not really! The Senior PROM is in the **Wait Mode**, waiting for you to tell it what to do next.

The Senior PROM will react to the following keys while in the Wait Mode (after Ctrl - Reset is pressed):

Key Pressed After Reset	Resulting Action
DEL	Enter the Monitor via $FF59. (i.e. old style Monitor reset).
=	Toggle between page 1 and page 2. Strictly for viewing, has no effect on memory.
–	Toggle between text and graphics mode. Strictly for viewing, has no effect on memory.
0	Toggle between lo-res and hi-res page. Strictly for viewing, has no effect on memory.
9	Toggle between text and graphics, and full graphics.
8	Move main upper 16k RAM memory (the language card) to accessible main RAM memory at $2000-5FFF.
7	Move auxiliary upper 16k RAM memory to accessible main RAM memory at $2000-5FFF.
6	Move main RAM from $00-8FF to $2000-28FF.
5	Move all 64k of main motherboard RAM memory to the auxiliary RAM card memory (the extended 80 column card).
4	Prepare to deactivate the Senior PROM (by flipping the toggle switch) and re-enter the normal //e environment.
3	Go directly to the Sector Editor.
2	Go to the Secondary Menu, the Memory Management menu.
1	Go to the Main Menu, the RWTS Utilities.
ESC	Reboot the disk drive via $C600 without disturbing any memory.
A	Alter an Auxiliary memory location.
B	Bank select upper 16k RAM.
C	Clear memory from $0100-BFFF and reboot the disk drive via $C600.
D	Disassemble 10 instructions before and after Program Counter.
G	Goto a specified address on Restart (instead of restarting at current running address).
Q	Quick Restart from NMI.
R	Restart from NMI.
S	Stack display.
W	Wait to deactivate Senior PROM before restarting.
X	eXchange program in Auxiliary memory with program in Main memory, and restart.
CTRL P	Print an image of Text Page 1 on the printer in slot 1 (//e only).

NOTE: When in the Wait Mode, a menu of the above items are not displayed on the screen. The reason for this is simple: to display a menu on the screen would require altering the memory space from $400 to $7FD (text page 1). This may be crucial memory that you do not want to disturb. To alliviate any confusion while in the Wait Mode, we have provided a keyboard overlay that you may place over your keyboard displaying the choices available while in the Wait Mode.

DELETE: Enter the Monitor.

From the Wait Mode, the DELETE key will put you into the monitor. This is the equivalent of having an "old Style" F8 Monitor ROM, as the old Apple II's had. In other words, the DELETE key will **ALWAYS put you into the Monitor,** regardless of where the Reset vectors are pointing (at $3F2-3F4). You may want to use this option to break out of a protected program, where (for example) the Reset key has been trapped to clear memory and reboot on a normal Apple //e.

To explain more clearly, the Reset key is a "programable" key. This means that when you press Reset on a normal Apple //e or //c, it jumps to the locations pointed to by $3F2 and $3F3. Hence if these two locations contain a $BF and a $9D, respectively, when reset is pressed you will jump to $9DBF (notice the "backwards" order). A jump to $9DBF is normally where Reset jumps in normal DOS 3.3 (it's the DOS warm entry point).

In many situations, a programmer will "trap out" the reset key. Normally the Reset key is a good way to exit a program, but it can be programmed (by changing locations $3F2-3F4) to point ANYWHERE, thus dictating exactly what reset will do. In many copy protected programs, this means the programmer has trapped the reset key to point to the beginning of the program, or to clear memory and reboot the disk. So if you press Reset, instead of ' exiting to the Applesoft prompt, the program starts over or memory clears and the disk reboots. This, of course, prevents curious eyes from examining the program code.

With the Senior PROM active, pressing CTRL - RESET and then the DELETE key will ALWAYS exit you to the monitor (thru $FF59), regardless of the contents of $3F2-$3F4. Thus you may examine or change any memory locations using standard Monitor commands.

You may also examine protected APPLESOFT programs using the Senior PROM to exit to the Monitor. To do this, reset into the Monitor, type "D6:00", deactivate the Senior PROM switch, press RETURN, and then CTRL-C. Assumming no secondary protection, you may list out any Basic program that may be in memory.

=:   Toggle the Current Display Mode Between Page 1 and Page 2.

While in the Wait Mode, the = key will toggle the display mode between page 1 and page 2. This has no effect on any memory regions or locations. In text and lo-res mode, page one memory is from $400-7FF, and page two memory is from $800-BFF. In the hi-res mode, page one memory is from $2000-3FFF, and page two memory is from $4000-5FFF. This option does not exit the Wait Mode.

-:   Toggle the Current Display Mode Between Text and Graphics.

While in the Wait Mode, the - key will switch the display mode between text and graphic modes. This has no effect on any memory regions or locations. This option does not exit the Wait Mode.

13

0: Toggle the Current Display Mode Between Lo-res and Hi-res.

While in the Wait Mode, the 0 key will switch the display mode between lo-res and hi-res graphics, providing you used the - key to toggle the graphics mode. Used in conjuction with key -, you can display either lo-res page 1 or 2, or hi-res page 1 or 2. This option does not exit the Wait Mode.

9: Toggle the Current Display Mode Between Mixed Text and Graphics, and Full Graphics.

While in the Wait Mode, the 9 key will switch the display mode between full graphics, and graphics mixed with four lines of text at the bottom of the screen, providing you used the - key to toggle the graphics mode. This option does not exit the Wait Mode.

8: Move Main Bank Switched Memory to $2000.

While in the Wait Mode, the 8 key will move the main upper 16k of RAM memory (sometimes called the language card or 16k RAM card area) down to $2000-5FFF, and ' then exit into the monitor.

This option has the advantage of allowing you to easily view the upper 16k "bank switched memory" (BSM) on the Apple. To explain, we need to discuss bank switching.

When the Apple was originally developed, RAM memory was very expensive. Original Apples came with only 16k, expandable to 48k. At the time, 16k was considered "lots" of memory. As the price of RAM chips decreased over the years, it became desirable to expand the Apple to 64k (which is the maximum memory that can be directly addressed by the 6502 micro-processor).

The problem with expanding the Apple past 48k of RAM had to do with its original architecture. RAM memory was set aside for the lower 48k memory from $00-BFFF, but the peripheral and ROM memory space was allocated to the upper 16k, $C000-FFFF. If you wanted to expand the Apple to 64k of RAM, you had to share the upper 16k with the ROM space, and "bank switching" was used.

In fact, only the top 12k of upper RAM is bank switched with $D000-FFFF. This is because memory from $C000-CFFF is reserved for the peripheral cards. There are two RAM card banks which occupy the same space: a 4k bank (occupying $D000-DFFF) and a 12k bank (occupying $D000-FFFF), thus giving you 16k total.

One of the hassles in accessing the upper 16k RAM memory is that in turning on the extra RAM memory, you must turn off the motherboard ROM memory. If you try and do this from the keyboard (the immediate mode), you will lock up the computer (it can be done from program control only). The reason is that you are turning off the ROM memory, which includes the Monitor ROM, which "monitors" your Apple's activities. Essentially, your computer becomes helpless and "dies" until powered off and back on.

To avoid this problem you must first put an image of the Monitor ROM into the upper most 2k ($F800-FFFF) of RAM memory before deactivating the ROMs. The problem with this is that you write over any memory in your upper RAM memory from $F800-FFFF. Here is the procedure (for the sake of completeness) from the monitor prompt (the notation "[RET]" represents pressing the RETURN key):

14

```
*C081 N C081 [RET]                                      Read from ROM, write to RAM.
*F800<F800.FFFFM [RET]                                  This moves the Monitor ROM to RAM.
*C080 [RET]         This turns on the upper 16k RAM and turns off the motherboard ROMs.
```

NOTE: You may access the RAM card without having an image of the Monitor ROM present. The way to do this is exclusively through program control, and never through the immediate mode. This is exactly how the Senior PROM moves the upper RAM memory down to accessible lower RAM ($2000-5FFF).

To summarize: After pressing CTRL – RESET and then the 8 key, the upper 16k or main RAM is moved down to $2000-5FFF for easier access (as explained above). Here are the exact locations of the two upper RAM memory banks after they are moved to $2000-5FFF:

> $D000-FFFF (bank1)   moved to $2000-4FFF
> $D000-DFFF (bank2)   moved to $5000-5FFF

7: Move Auxiliary Bank Switched Memory to Main Memory at $2000.

While in the Wait Mode, the 7 key will move the Auxilary upper 16k of RAM memory to $2000-5FFF of main RAM memory. This does the same thing as the 8 key, but instead of moving the upper 16k of Main RAM memory, it moves the upper 16k Auxiliary RAM memory to Main memory at $2000-5FFF. Note that 128k of RAM is required (an Extended 80 Column card for the //e) for this fuction to work.

For the same reasons you would want to move the upper 16k of Main RAM memory to lower Main memory, you would also want the ability to move the Auxiliary 16k of upper RAM memory to lower main memory. This saves you the hassle of having to deal with using the upper 16k of RAM (as described in the 8 key section) and also having to deal with the Auxiliary RAM memory bank.

Here are the exact locations of each of the two Auxilary RAM memory banks after they are moved to main memory $2000-5FFF:

> $D000-FFFF Auxiliary (bank1)   moved to $2000-4FFF
> $D000-DFFF Auxiliary (bank2)   moved to $5000-5FFF

6: Copy Locations $00-8FF to $2000-28FF.

While in the Wait Mode, the 6 key will copy unaltered all the memory from $00-8FF up to $2000-28FF.

The advantage to this option is that you may examine a program running in the "volatile" memory areas, $00 to $7FF. For example, many protected programs use text page 1 ($400-7FF) as a program area to protect valid code from easy viewing. A normal //e or //c reset will destroy this memory area.

The reason a normal Reset destroys this memory is whenever you type a command or do a standard reset, something gets output to the text screen, thus destroying any program living there before the Reset. Publishers know this, and to prevent you from looking at their code, they sometimes use this memory space for valid code. You have probably seen this before when you are playing a hi-res game and you hit reset. Suddenly, the screen is filled with text

garbage. The "garbage" is really program code, and that's how it looks in ASCII format on the text screen. Since you were in a game viewing the hi-res screen, you never saw the "garbage" text screen until you pressed Reset.

Likewise, the keyboard input area ($200-2FF), the stack ($100-1FF), and many zero page locations ($00-FF) are destroyed upon a normal reset. Thus if there was any program code there, it would be destroyed.

With the Senior PROM active, after pressing CTRL - RESET and the 6 key, the first action done is to move all the registers to $2900-2904, and then copy memory from $00-8FF up to $2000-28FF. Then a normal "old style" reset is performed via $FF59, and you are exited into the monitor. The registers are saved at the following locations:

    Accumulator = $2900
    X-Register = $2901
    Y-Register = $2902
    Stack Pointer = $2903
    Status Word = $2904

You now may list and view the $00-8FF code at $2000-28FF. The code there is lower memory just before you pressed reset. This proves to be invaluable in understanding the works of some copy-protected programs using volatile memory areas.

A final note on this option: This routine also moves memory from $800-8FF up to $2800-28FF, even though a normal reset will not destroy this memory area (it is not "volatile" memory). The reason is so you may boot a 48k DOS 3.3 slave disk without losing any memory from $00-8FF. Remember, when booting a 48k DOS 3.3 slave disk, memory from $800-8FF is destroyed. Thus if we didn't move $800-8FF upto $2800-28FF you would have to do so manually if you wanted to boot a DOS disk and save $800-8FF (and $00-7FF) to disk.

5: Copy all 64k of Main RAM to Auxiliary RAM.

While in the Wait Mode, the 5 key will copy all Main 64k of motherboard RAM memory to the Auxiliary 64k RAM memory (extended 80 column card), including all the registers pushed on the stack. You are returned to the Wait Mode after this is complete, and an inverse "S" will be displayed at the lower right of you screen. **Because of this, 128k is required in your Apple //e for this option to work.**

This allows you to take a "snapshot" of a program running in the main 64k memory area of your Apple, and later save it to a normal DOS 3.3 disk in file form. When the NMI switch is used (the push-button switch next to the toggle switch) to interrupt the program rather than CTRL - RESET, the program can be captured in memory or on a normal DOS 3.3 disk and later restarted. A whole chapter has been dedicated to the NMI functionality and follows in this manual.

4: Enter BASIC.

From the Wait Mode, the 4 key will allow easy entry back into the environment you were in before you activated the Senior PROM. When pressing this key, you will hear a "beep" signifying you should move the toggle switch to the deactivated position.

For example, say you have DOS 3.3 booted and you want to format a disk without DOS. You can do this from within the Senior PROM environment by first activating the Senior PROM and using the ROM resident utilities to format a disk without DOS.

To re-enter the DOS 3.3 environment from the Senior PROM by simply pressing CTRL – RESET and then press the 4 key. You will hear the familar Senior PROM "Beep", and may deactivate the switch. After you do so, you will be right back in your DOS 3.3 environment, as though you never left it. If DOS was not present, you will be in Basic, as though you pressed CTRL B from the Monitor.

3: Go to the Sector Editor.

While in the Wait Mode, the 3 key will bring you directly into the Sector Editor disk and memory edit utilities. Note that using this direct method only disturbs zero page, page 1 and page 2. To exit the Sector Editor, you may press either CTRL – C or 1 to enter the Senior PROM Main Menu, or press DEL to exit to the Monitor.

For more information regarding the Sector Editor, please refer to that specific chapter.

2: Go to the Memory Management Menu System.

While in the Wait Mode, the 2 key will take you to the secondary Senior PROM menu, the Memory Management Menu. This menu lists all the memory management options available from within the Senior PROM. This menu will be discussed fully in a following chapter.

1: Go to the Main Menu System.

While in the Wait Mode, the 1 key will display the main Senior PROM internal menu: the RWTS Utilities Menu. This menu will list all the RWTS (Read – Write – Track – Sector) functions available from within the Senior PROM. The RWTS Utilities Menu is explained fully in a following chapter.

ESC: Reboot the disk drive.

From the Wait Mode, pressing the ESC key will boot drive 1, Slot 6 automatically without disturbing any memory. This is (almost) the equivalent of the CTRL – OPEN APPLE – RESET that most of you are familar with on the Apple //e and //c, with one exception.

When you press CTRL – OPEN APPLE – RESET on a normal //e or //c, a routine is jumped to in the monitor that wipes out 2 bytes of each page in memory. You may notice this by watching the text screen closely when you hit CTRL – OPEN APPLE – RESET. Notice the appearance of four text characters in a diagonal on the lower right side of the screen. This is the Monitor destroying several bytes in the text page memory area, $400-7FF.

While in the Wait Mode, the ESC key will reboot drive 1 <u>without distroying any memory locations</u>. This has the advantage of being able to reset out of a protected program, and then booting a 48k DOS 3.3 slave disk, which will not write over memory from $900-95FF and from $D000-FFFF. You may then examine or save these memory areas as desired.

A practical example of this would be if you were using a hi-res program, and you wanted to save or print the hi-res screen for later reference. All you would have to do is to activate the Senior PROM, press CRTL - RESET and then ESC, and boot a 48k DOS 3.3 slave disk. Then you could save the hi-res page to disk with a standard DOS command (i.e., for hi-res page 1, type]BSAVE PIC,A$2000,L$2000).

NOTE: Just after pressing the ESC key, and while the drive is recalibrating, you should deactive the Senior PROM switch. You should do this because after DOS is loaded, the program booted may try to access BASIC (any normal DOS 3.3 disk does this). Because of this, the Senior PROM must be deactivated for the boot to continue.

If you forget to deactivate the Senior PROM when booting a normal DOS 3.3 disk, in a few moments you will exit into the monitor with a beep. You may continue loading the disk without re-booting by deactivating the Senior PROM switch, and then typing "9D84G". This will continue the load as normal.

A Senior PROM Deprotection Trick: If you want to stop a program from loading that is using some sort of modified DOS, boot the disk with the Senior PROM activated. This will allow the program to load its DOS, but will break into the monitor before the DOS can load the initial BASIC "Hello" program. Now you can easily examine the DOS memory before it is manipulated by any of the programs on the disk.

CTRL P: Print Image of Text Page 1 on Printer (//e version only).

While in the Wait Mode, pressing **CTRL P** will print an image of Text Page 1 on your printer (text page 1 is the text screen viewed immediately after pressing NMI or Reset), providing there is a printer peripheral card in slot 1. You will then be returned to the Wait Mode. Note this option uses pages 0 and 1 (the Stack), and only works on the //e.

C: Clear Memory and Reboot.

This option will write a test pattern into memory from $100-BFFF (the lower 48k) and reboot the disk drive in slot 6. The test pattern writes a single byte value of each page into all 48k of memory. For example, page $08 will be filled with repeated $08's, page $7F will be filled with repeated $7F's, and so on. After writing the test pattern into memory, the disk drive in slot 6 is booted.

For the remainder of the Wait Mode options (A, B, D, G, Q, R, S, W, X), please refer to Chapter 9, **Using the NMI and Resurrect Programs.**

Chapter 6: The Main Menu - RWTS Utilities.

The Senior PROM Main Menu displays the available RWTS related functions available. RWTS (Read - Write - Track - Sector) is the portion of DOS that does the direct reading and writing of sectors on a disk and lives from $B700-BFFF in a 48k Apple. Note that with the Senior PROM installed you do not have to load DOS in your Apple to use these functions. The reason is that the Senior PROM has RWTS built into it, so to format, copy or edit a disk, you only have to turn you Apple on and use the Senior PROM. You do not have to boot a disk first!

In addition, if you have a DOS and associated RWTS already present in your Apple, you may "protect" it so the Senior PROM does not overwrite the current RWTS in memory. The advantage to this is if you have a certain DOS environment that you don't want to disturb, you can still use the Senior PROM and your RWTS. This will prove to be very valuable when trying to determine copy protection schemes used on a diskette.

What follows is a list and description of the functions available from the Senior PROM Main Menu. If an asterisk (*) appears before the function, the Senior PROM will use its own RWTS unless you first "protect" yours. For a further discussion of "protecting" your RWTS, see option 7, Protect RWTS in Memory.

Invoking the Senior PROM Main Menu - RWTS Utilities.

While in the Wait Mode (after pressing Ctrl - Reset), you may enter the main Senior PROM menu system (the RWTS Utilities) any one of three ways:

1) In the Wait Mode, the 1 key will put you directly into the Senior PROM main menu system.

2) From the Wait Mode enter the Monitor by pressing the DELETE key or any other key that will eventually place you in the monitor. Next press CTRL C and then the RETURN key to enter the Senior PROM main menu system.

3) From the Wait Mode enter the Monitor by pressing the DELETE key or any other key that will eventually place you in the monitor. Type "E003G" and then the RETURN key to enter the Senior PROM main menu system.

After performing any of the above three sequences, the Senior PROM Main RWTS menu will appear on your screen:

```
-------------------------------------------------
!   Senior PROM v3.00 @1986 - Enter Choice:  !
!                                             !
!    1) Format a Disk                         !
!                                             !
!    2) Copy a Disk                           !
!                                             !
!    3) Go to the Sector Editor               !
!                                             !
!    4) Enter BASIC                           !
!                                             !
!    5) Disable DOS Error Checking at $B942   !
!                                             !
!    6) Alter Prolog Bytes                    !
!                                             !
!    7) Protect RWTS in memory                !
!                                             !
!    8) Enter Memory Mgmt Menu                !
!                                             !
-------------------------------------------------
```

Any single key press will result in the desired fucntion (the RETURN key is not neccessary). Pressing the DEL key will exit into the Monitor. Note that the DEL key is also the conventional exit key to abort the current function.

*Option 1 - Format a disk: Pressing the 1 key from the main menu will prompt you to enter the drive number in which you want to format a disk. If you do not wish to format a disk, you may press the DEL key to exit this function. Otherwise, place the disk to be formatted in the appropriate drive and press either the "1" or the "2" key to format the disk.

Next you will be prompted for the volume number for the disk. If you just press the RETURN key, you will receive the default volume number of $FE (254 in decimal). If you don't press the RETURN key, you may enter any volume number (including $00), in hex. Note the prompt will take only numbers from 0 to 9, and letters $A to $F.

The disk you placed in the drive will now be formatted to that volume number, and a DOS 3.3 VTOC (Volume Table of Contents) added to the disk on track $11 (17 in decimal). NOTE THAT NO DOS IS WRITTEN TO THE DISK! Also note you may at a later time add DOS 3.3 to the disk (using the Senior PROM and any DOS disk), since tracks $00 - 02 were allocated in the VTOC for that purpose.

If you wish, you may enter the Sector Editor and use it to un-allocate the DOS tracks, thus giving you $20 (32 in decimal) extra sectors for storage space.

*Option 2 - Copy a Disk: Pressing the 2 key from the main menu will prompt you to copy a disk from drive 1 to drive 2 or from drive 2 to drive 1. **Note that two disk drives are required.**

If you wish to exit this function press the DEL key to exit. If you want to switch the drives (copy from drive 2 to drive 1), press "D". Otherwise, press any other key to begin.

At this point, you should place the original disk in the source drive and the destination disk in the destination drive. It doesn't matter whether the destination disk has already been formatted or not.

You will now be asked for the starting track to copy from, inclusive. If you want the default starting track of $00, just press the RETURN key. Otherwise you may enter any track from $00 to $23 in hex only (two digits must be entered). This is the track where the copy will start, including that track.

Next you will be prompted for the ending track to copy, inclusive. If you want the default ending track of $22, just press the RETURN key. Otherwise you may enter any track from $00 to $23 in hex only (two digits must be entered). This is the track where the copy will end, including that track.

This allows you to select exactly what tracks you want to copy. If you want to do a standard disk copy, just insert the disks and press the RETURN key at the starting and ending track prompts. This will do a standard disk copy much like COPYA does.

If the destination disk is already formatted, the copy function will not re-format the disk. If your destination disk was not previously formatted, you will be prompted to enter a volume number for the destination disk. If you press the RETURN key, the destination disk will be formatted with the original disk's volume number. You may over-ride this feature by entering the desired volume number in hex.

If a write error occurs during the copy process, you will be prompted to format the destination disk. After formatting the destination disk the copy will start over, using the starting and ending tracks you specified. This is because formatting a disk wipes the disk clean, and the copy must start over.

If you wish to stop at any point during the copy, press the DEL key. The copy process will stop and you will be returned to the main menu.

If you wish to force the destination disk to be formatted (regardless of whether it was previously formatted), you may do so by leaving the destination disk drive door open when the copy process starts. This will cause a write error, and you will be prompted for the volume number for the destination disk. Be sure to close the drive door before continuing the copy.

During the copy process, the current track and sector (in hex) that are being processed are displayed in the lower right hand corner of the screen. If the source drive is active, this is the current track and sector being <u>read</u>. If the destination drive is active, this is the current track and sector being <u>written</u>.

If there is a disk error encountered on the read or write, the drive will recalibrate (the grinding noise) and the speaker will beep, and the track and sector where the error occurried can be noted at the lower bottom right of the screen for a short duration. This display will remain on the screen till the next usable sector is encountered.

As a final note, memory from $800 to $95FF is used as a copy buffer. Even though additional memory could have been used as a buffer, this choice of a copy buffer will not disturb DOS, if present (from $9600-BFFF)

*Option 3 - Go to the Sector Editor: Pressing the 3 key will place you in the Sector Editor fuction. These utilities will allow you to view, edit, and modify any sector on a disk. Please refer to the Sector Editor portion of this manual for instructions.

Note that you may press CTRL - C or the 1 key to return to the Senior PROM Main Menu from the Sector Editor.

Option 4 - Enter BASIC: Pressing the 4 key allows you to enter BASIC cleanly from the Senior PROM to the previous environment you were in before you activated the Senior PROM. When pressing this key, you will hear a "beep" signifying you should move the toggle switch to the deactivated position.

After you do this, you will be in BASIC, and DOS will also be active if the correct recognition bytes are present. If you did not use the "protect" option of the Senior PROM, memory from $B700-BFFF will have been overwritten.

If DOS was present before you activated the Senior PROM and you "protected" RWTS, your BASIC environment should be the same as before you activated the Senior PROM.

Regardless of whether you used the "protect" option or not, some zero page, page 1, page 2 and page 3 locations have been changed from their previous values.

Option 5 - Disable/Enable DOS Error Checking at $B942: Pressing the 5 key will toggle the DOS error checking routine at $B942 on or off (by altering the byte at $B942), depending on its previous state. For example, if the DOS error checking is currently on (enabled), pressing the "5" key will turn it off (disabled). Likewise, if the DOS error checking is currently off (disabled), pressing the "5" key will turn it on (enabled). The screen is then updated as to what would happen if the "5" were pressed again.

To explain the DOS error checking at $B942, if a read or write error occurs the current routine is routed to location $B942. Here the Carry Flag is set (hex $38 or the mnemonic SEC), and the routine returns. DOS tracks the Carry Flag and if it is set, it knows a disk error has occured.

To disable this routine, the Senior PROM puts a Clear the Carry Flag instruction (hex $18 or the mnemonic CLC) at location $B942 so the Carry Flag never gets set if a read error occurs. This will allow you to copy a disk that has been modified or damaged, although the data read from the damaged sector might be distorted.

Another very good use of this option is on copy-protected disks that use different address or data epilogue bytes. Epilogue bytes are a unique two byte sequence that DOS looks for to verify it has read the correct amount of data. Many copy protection schemes change this two byte sequence from the normal DOS 3.3 format of $DE AA to something else. This defeats any standard disk copy program. See the enclosed manual "Deprotection Methods and Techniques Using the Senior PROM" for a complete discussion of Data and Address Epilogue and Prolgue bytes.

Option 6 - Alter Prolog Bytes (and also Epilog Bytes): Pressing the **6** key will allow you to change the normal DOS 3.3 Prolog bytes from their standard three byte sequence. This will allow you to read many protected programs as if they were in a normal DOS 3.3 format, for easy backups, modifications and editting.

To copy a protected disk with modified Prolog bytes, you must first know what the modified prolog address and data byte sequences are. You can use the Nibble read command of the Sector Editor to help you determine these (see the chapter on the Sector Editor).

After determining the disk's modified prolog bytes, select option 6 from the Senior PROM Main Menu, "Alter Prolog Bytes". Now you will be prompted to enter the modified Address Prolog bytes. Each byte is entered as two hex digits without a RETURN keypress. The normal DOS three byte sequence is $D5 AA 96, and if you press the RETURN key three times, you will get the normal DOS sequence.

Next you will be prompted to enter the new Data Prolog bytes. The normal DOS three byte sequence is $D5 AA AD, and if you press just the RETURN key three times, you will get the (default) normal DOS sequence.

Now you can select option 2, "Copy a Disk", and copy the protected disk with modified Prolog bytes to a normal DOS 3.3 disk. This will read the protected disk with the modified Prolog bytes you specified, and write to the blank disk using normal DOS Prolog bytes. The copy will be in normal DOS 3.3 format, and you may copy, edit and alter this disk as you wish.

Defeating the Epilog bytes: Note that if you want to defeat only the Epilog byte check routines for copying a disk with modified Epilog bytes, you may use this option. Simply select option 6 (change Prolog bytes), and press the RETURN key to accept all six default normal DOS Prolog bytes. This will automatically defeat the epilog byte check routines and allow you to make a disk copy of many protected programs.

Copying a disk in this manner with modified Epilog bytes has a advantage over defeating the DOS error checking at $B942 (option 5) and copying the disk. The advantage is that the DOS error checking is still intact, so the data read from the original disk is checked by RWTS for validity.

Changing Back to Normal DOS 3.3 Format: After modifying the Prolog bytes, you will notice that the menu option has changed to "normalize Prolog bytes". Select the "Alter Prolog Bytes" option again, and you may copy disks with normal DOS Prolog bytes. You will see option 6 change back to "Alter Prolog Bytes".

Option 7 - Protect/Unprotect RWTS in Memory: Pressing the **7** key will allow you to use the current RWTS already in memory from $B700-BFFF. This will protect the current RWTS (and associated DOS) from being overwritten by the RWTS image that is contained in the Senior PROM. Unless the option is selected, the Senior PROM RWTS will be moved down to $B700-BFFF by all of the above options preceded by an asterisk (*).

Protecting the current RWTS has the advantage of allowing you to make certain changes to RWTS and not have them overwritten by the Senior PROM Copy, Format or Sector Editor fuctions, or to use an altered RWTS read in from a copy protected disk. This also allows you use the Senior PROM and to return to your previous DOS environment without any surprises. In addition, you may "unprotect" the RWTS in memory and use the RWTS contained in the Senior PROM. To do this, press the "7" key once more to unprotect the current RWTS. <u>Beware that using any of the "*" options will overwrite memory from $B700–BFFF once RWTS has been "unprotected".</u>

Option 8 – Go to the Memory Management Menu System: Pressing the **8** key will place you in the Memory Management Menu system. This set of options are described in the next section. Press the "8" again to return to the Main Menu.

Chapter 7: Memory Management Menu.

The Senior PROM Memory Management Menu displays the available memory related options. These routines will be helpful when trying to capture, view or alter memory portions of a program.

Invoking the Memory Management Menu.

While in the Wait Mode (after pressing CTRL – RESET), you may enter the Memory Management Menu any one of two ways:

1) While in the Wait mode the 2 key will put you directly into the Memory Management Menu system, disturbing only the memory normally disturbed during a reset and text page 1 ($400–7FF).

2) While in the Main Menu RWTS Utilities menu, you may press the "8" key. This key is a toggle between the two menu systems, the RWTS Utilities and the Memory Management Menu.

After performing one of the above two sequences, you will see the Senior PROM Memory Management Menu appear on your screen:

```
-------------------------------------------------
!                                               !
! Memory Mgmt Menu                              !
!                                               !
!                                               !
! 1) Copy all of main to aux                    !
!                                               !
! 2) Transfer pages 0-85 from aux to main       !
!                                               !
! 3) Transfer pages 86-BF from aux to main      !
!                                               !
! 4) Move 0-8 to 20-28                          !
!                                               !
! 5) Move main BSM to pages 20-5F               !
!                                               !
! 6) Move aux BSM to pages 20-5F (main)         !
!                                               !
! 7) Write a test pattern into main mem         !
!                                               !
! 8) Return to main menu                        !
!                                               !
-------------------------------------------------
```

Any single key press will result in the desired function (the RETURN key is not neccessary). Pressing the DEL key will exit from this menu into the monitor.

Option 1 – Copy all of main memory to aux 64k: Pressing the 1 key from the
Memory Management menu will copy all of Main 64k of motherboard RAM memory to the Auxiliary 64k RAM memory (extended 80 column card).

This option has the most functionality when used through the reset key or NMI because no orignal RAM memory will have been disturbed. By going through the Memory Management menu, original memory from $400-7FF (text page 1) and many zero, $01, $02, $03 page locations are disturbed.

If these memory areas are of no consequence to you, this menu item will work fine. You may want to refer to the NMI portion of this manual for additional uses for this function. Note that this performs the same function as the CTRL – RESET, "6" sequence.

Option 2 – Transfer pages $0-85 from Auxiliary to Main Memory: Pressing the 2 key from the Memory Management menu will transfer locations $00-85FF of the Auxiliary memory to Main memory. This option is used after you have NMI'ed or Reset, and moved all of Main memory to Auxiliary memory. This will take memory in the Auxiliary 64k (Extended 80 Column Card in the //e) from $0-85FF and move it to $1000-95FF of Main memory .

After moving the memory from the Auxilary memory back to Main memory at $1000, you can boot a 48k slave disk and save this memory area by typing:

]BSAVE MEM$0-85FF,A$1000,L$8600

You may then examine this memory at your leisure.

Option 3 – Transfer pages $86-BF from Auxiliary to Main Memory: Pressing the 3 key from the Memory Management menu will transfer $8600-BFFF of the Auxilary memory to Main memory. This option is also used after you have NMI'ed or Reset, and moved all of Main Memory to Auxilary memory. This will take memory in the Auxilary 64k (Extended 80 Column Card in the //e) from $8600-BFFF and move it to $1600-4FFF of Main memory.

After moving the memory from the Auxilary memory back to Main memory at $1000, you can boot a 48k slave disk and save this memory area by typing:

]BSAVE MEM$8600-BFFF,A$1600,L$3A00

You may then examine this memory at your leisure.

Option 4 – Move Main Memory $00-08 to $20-28: Pressing the 4 key from the Memory Management menu will transfer $00-08FF of Main memory up to $2000-28FF of Main memory (equivalent to the infamous "Krakrom" of the Apple II and II+). This option is used after you have NMI'ed or Reset, and you want to save the volatile memory areas in the text page, stack, zero page and input buffer for further examination. It has little use from this menu since these volatile memory areas have already be disturbed.

Option 5 – Move Main BSM to Pages $20-5F: Pressing the 5 key from the Memory Management menu will transfer Main Bank Switched Memory (BSM) from $D000-FFFF (bank 1) and from $D000-DFFF (bank 2) to $2000-5FFF of Main memory. This option allows you to view the upper 16k of Main Bank Switched Memory easily. Here is where each of the two banks are transfered to in lower main memory:

$D000-FFFF (bank1)	moved to $2000-4FFF
$D000-DFFF (bank2)	moved to $5000-5FFF

Option 6 - Move Aux BSM to Pages $20-5F: Pressing the 6 key from the Memory Management menu will transfer Auxilary Bank Switched Memory (BSM) from $D000-FFFF (bank 1) and from $D000-DFFF (bank 2) to $2000-5FFF of Main memory. This option allows you to view the upper 16k of Auxilary Bank Switched Memory easily. Here is where each of the two banks are transfered to in lower main memory:

$D000-FFFF (bank1)	moved to $2000-4FFF
$D000-DFFF (bank2)	moved to $5000-5FFF

Option 7 - Write a Test Pattern into Main Memory: Pressing the 7 key from the Memory Management menu will write a test pattern into all 64k of Main Memory. This test pattern writes the single byte value of each page into all 64k of memory. After writing the test pattern into memory you are placed in the Wait Mode.

For example, after selecting this option page zero page will be filled with repeated $00's, page $01 will be filled with repeated $01's, and so on. In banked switched memory, the same applies except pages $D0-DF of bank 2 are filled with the bytes $C0-CF, respectively.

You may now press any of the standard Senior PROM reset function keys for the desired function (most likely the "=/+" key to boot a disk, or one of the memory move functions to study its effect.

The purpose of this option is to allow the user to see what portions of memory gets loaded by a particular program. If this option is selected just before booting a particular program, you may boot the program and at the desired instance, reset into the Monitor. Now enter the Sector Editor via $D000 or through the RWTS Main Menu (use the "Protected DOS" option of the Main RWTS Menu so as not to write RWTS into $B700-BFFF), and use the "<" and ">" keys to view memory. From this you can tell what portions of memory a program has loaded into.

If you wish to write a test pattern into Auxiliary memory, you may do so using this option too. Simply select option 7 from the Memory Management menu, and then press the "X" key to move all 64k of Main memory into Auxiliary memory. This will move the test pattern written in Main memory into Auxiliary memory. Now you will be in the monitor with a test pattern in all 128k of both Main and Auxiliary memory.

Option 8 - Return to Senior PROM Main Menu: Pressing the 8 key from the Memory Management menu will return you to the Senior PROM Main RWTS Menu.

Chapter 8: Using the Sector Editor.

There are three ways to enter the Sector Editor. Here are the available choices:

1) From the Wait Mode, press the **3** key. Note that this will put you in the sector editor with whatever RWTS is currently present in memory.

2) If you are at the monitor prompt with the Senior PROM active, you may enter the Sector Editor by typing "D000G".

3) If you are at the Senior PROM main RWTS menu, press the **3** key. This will put you into the Sector Editor with the Senior PROM RWTS present at $B700-BFFF. If you wish to use whatever RWTS is currently in memory at $B700-BFFF, use the "Protect RWTS" option of the Senior PROM.

What is a Sector Editor?

A sector editor allows you to examine or edit the data that is either in your //e's or //c's memory, or on a disk. You can also edit and examine disk with non-standard DOS formats.

Generally you can edit a program on a disk by loading the file, changing it, and then by re-saving it. But what if you want to change DOS, for example, so it BRUNs the hello program instead of RUNing it? You can not BLOAD DOS, edit it, and then BSAVE it. Instead you must use a sector editor to load the sector directly from the disk that you want to change, edit it, and then write the sector back out. This is just one small job a sector editor can perform.

Sector editors load in information on a sector by sector, track by track basis, instead of a file by file basis.

Some other chores commonly performed by sector editors are locating a certain byte sequence on a disk, altering the current DOS on a disk, editing a program directly on a disk, finding a string in memory or on a disk, comparing two disks, reading or writing half tracks, etc.

Upon entering the Sector Editor, you should become familiar with the display. The top two lines are the status display. This shows the current track, sector, memory buffer, drive and slot that will be read, written and displayed.

The four digit number on the second line, far right (just below the drive number) is called the checksum. This number gives a checksum of the current memory buffer. This is useful when comparing memory pages or disk sectors.

Down the left side of the screen are the starting byte numbers, incremented in $10's. This means the upper left byte is byte $00, followed by byte $01, etc.

Reading a sector: Insert a normal DOS disk in a drive and press the "R" key. This will read the track and sector (displayed at the top of the screen) from the disk in the drive and slot number displayed, into the memory buffer displayed. (Note that each sector on a disk equals exactly one memory buffer page.)

You can press the "A" key to display the memory buffer in either hex or ASCII. If you want to read a different track and sector, press "T" and then enter the track desired in a two digit format. For example, if you wanted track 1, enter "01". To change the sector, press "S", and enter the sector desired with a single key press. If you wish to read this track and sector into a different memory buffer, press "B" and enter the page number as two digits ("1A for the memory buffer $1A00-1AFF). Also, the greater and less than sign keys increment or decrement the buffer page by one, respectively.

To edit a byte in the current memory buffer, first press "A" to select either Hex or ASCII display format. This displays the buffer in the chosen format and allows you to edit the buffer in that format. Now press "E" and enter the starting byte to edit (in two digit format). Now enter the hex byte or ASCII character (depending on the mode selected) you want to change that location to. In the Hex mode bytes are entered in a two digit format, and the current value of the location is printed after the colon, and is also hi-lighted in the buffer display in inverse text. The **Space Bar** may be pressed to leave the current location unchanged and to proceed to the next location. In the ASCII mode, characters are entered as a single key press, and the location being changed is hi-lighted in the buffer display between an inverse greater-than and less-than signs. After entering the byte desired, you will be immediately prompted to edit the next consecutive byte. If you wish to stop editing, press RETURN. If you wish to continue, enter the next Hex byte or ASCII character.

If you are editing in the ASCII mode, you may enter the ASCII as either Flashing or Inverse. To do so, press "E" (for edit) and then the starting byte number to edit. Now press CTRL F for Flashing, or CTRL I for Inverse. Then type in the ASCII characters you wish to enter. Press RETURN to exit the edit mode when desired.

After editing a memory buffer as desired, you may write it back to a disk. First make sure the track, sector and drive number are correct. Then press CTRL W to write the current memory buffer to the track and sector specified of a disk.

These are some basic chores you can use a sector editor for. A summary of all the features of the Sector Editor follow.

Finding Strings in memory or on a disk: Using the "F" (Find) and "L" (Locate) commands you can find either an Ascii or Hex string in memory, or on a disk. The Ascii Find will locate the entered Ascii string in any possible format: flashing, inverse, lowercase or control characters. Another feature of the Sector Editor's Find and Locate commands is the ability to use "Wildcards". For example, say you want to locate all JMP's to memory in the $2000-20FF range. This is easy to do with the Wildcard. Just enter the string to search as "4C = 20". This will locate all jumps to the $2000-20FF range. Refer to the summary of Sector Editor's Command for an exact description of the procedure and syntax.

Summary of Sector Editor's Commands:

Display a sector in either ASCII or HEX:
Press "A" to change viewing from HEX to ASCII. Press "A" again to switch from ASCII to HEX.

Specify desired track to read or write:
Press "T" and enter the desired track as two Hex bytes.

Specify desired sector to read or write:
Press "S" and enter the desired sector, as a single Hex byte.

Reading a sector or block:
Press "R" to read the sector specified.

Writing a sector:
Press CTRL W to write the sector specified.

Reading next sector:
Press "=" to read next sector.

Reading next (incrementing) sector:
Press "SHIFT =" to read the next incremental sectors continually, to be interrupted by pressing any key.

Reading previous sector:
Press "−" to read previous sector.

Reading previous (decrementing) sector:
Press "SHIFT −" to read the next decrementing sector continually, to be interrupted by pressing any key.

Editing a byte or consecutive bytes:
Press "E" and then enter the appropriate HEX byte to edit. Now enter Hex or ASCII, depending on whether the screen is currently displaying HEX or ASCII (use the "A" key to toggle). If in the Hex mode, the byte being editted in the buffer will be displayed in inverse. If in the ASCII mode, the byte being editted in the buffer will be displayed between two inverse arrows. You may then type in the value to change this location to. In the Hex mode, two keystrokes are accepted per location representing a Hex value between $00 and $FF. In the ASCII mode, the key typed will appear in the buffer display. You will be immediately prompted to edit the next byte. Press RETURN to exit the edit mode.

If editing in the Hex mode, the **Space Bar** may be pressed to leave the current location unchanged and to proceed to the next location.

If editing ASCII characters, after entering the starting byte to edit, you may press CTRL I or CTRL F to enter Inverse or Flashing characters, respectively.

Changing memory buffer page to View or Read/Write:
Press "B", enter the desired memory page as two digits. Or press "<" or ">" to decrement or increment buffer page, respectively.

Reading/Writing incrementing sectors and pages:
First select the track, sector and buffer to read or write to. Press "R" to read, or CTRL W to write. Press CTRL N to increment sector and memory page, and to read or write successive pages to successive sectors of the disk.

Disassembling code and ASCII Equivalents:

Select appropriate memory page (using "B" command). Press CTRL D to disassemble starting at selected memory page. Press "Q" to toggle between fast and slow display mode. Use the SPACEBAR to temporarily interrupt the scrolling.

HEX/ASCII Dump:

To view a HEX and ASCII dump from the current buffer pages, press "H". The "Q" key will toggle between fast and slow display mode.

Finding a HEX or ASCII byte sequence in memory or on a disk:

Press "F" to to invoke the find routine. The prompt "Hex, Ascii, Same:" will then be printed on the screen. This is asking to find a Hex or Ascii string, or use the Same string as previously entered.

If "S" is chosen, the string previously entered from Find will be displayed (proceeded by an "A" or a "H" designating it was either an Ascii or Hex string), and the string will immediately be searched for.

If "A" or "H" is chosen, enter the Ascii or Hex string desired, with a maximum or 16 characters or bytes, respectively. A RETURN will end the string, and you will be asked to search for this string in "Memory, Disk". If "M" is entered, the string is immediately searched for in memory, and the starting location of each occurance (in Hex) is printed on the screen (Find will find all Ascii stings found, including flashing, inverse, lowercase or control character equivalents).

If "D" was entered to find the string on disk, the main display will appear with an inverse "L" on the upper right of the screen. This indicates that Locate is turned on. Now set the track and sector to the appropriate values, and use the continue read command to read the disk ("+" to increment sectors, "_" to decrement sectors read automatically). When (and if) the string is located, the speaker will beep and the locations in the current buffer at which the string was found will be printed. If you wish to continue searching the disk, press "+" or "_" again.

Wildcards: If you are searching for a Hex string, you may enter a "Wildcard" (this is not available for Ascii strings). This enables you to search for an ambiguous string. To do this, choose "H" for a Hex string, and enter the string. To designate a wildcard, enter a "=". This will match all the characters in the string exactly, except for the "=". The byte represented by the "=" is the wildcard, and can be any byte. This is useful for finding JMP's or JSR to approximate locations and memory pages.

Switch disk drive to use:

Press "D" to switch from drive 1 to drive 2, or vice versa.

Verifying a disk:

Insert appropriate disk, set the track and sector, and press "+" or "_" (depending if you want to increment or decrement the current track and sector, respectively). If the error message DISK ERR appears on the screen just below the drive number, the current track and sector can not be read and is assumed bad.

Compare and Verify two disks:

Insert the first disk, press CTRL V. When done, insert the second disk and press CTRL V. Different sectors will be displayed with a "+".

Map of used sectors on DOS 3.3 disk:

Press "M" and used sectors are marked on the map by "+".

Nibble read of disk:

Enter the track number you wish to Nibble read (sector numbers doesn't matter as the entire track is read). Press "N", and that track will be read. A hi-res display of the track is shown. Press a key and a hex dump of the track will be displayed. Use the Space bar to stop and start the Hex listing, "Q" to toggle fast and slow display. The RETURN key exits.

Exclusive OR Buffer:

You may EOR the entire buffer displayed with any value by pressing CTRL-E, and then entering the the value you wish to EOR the buffer with. This will make editing flashing or inverse text much easier. To return the buffer to normal, EOR the buffer with the same value once again.

Blank Current Buffer:

Pressing CTRL B and the current memory buffer will be blanked out with $00's.

Convert HEX to Decimal and Decimal to HEX:

Pressing "C" will put you in the convert calculator mode. If you wish to convert HEX to decimal, enter the number preceded by a "$" sign. Press RETURN to exit this mode.

File Follower:

Set the track and sector to the first Track/Sector list of a DOS 3.3 file. Press CTRL F and all the file's sectors will be mapped as they exist on the current disk.

Screendump (//e version only):

Pressing CTRL P from any prompt will print the current screen to your printer. Note you may get a printout of any screen within the sector editor. A printer in slot 1 is assumed. If you do not have a printer connected, this will hang the Sector Editor, allowing you to only use RESET to exit.

Exit to the Monitor:

Press the DEL key to exit to the monitor.

Exit to the RWTS Main Menu:

Press CTRL C to exit to the RWTS Main Menu.

Reading Half Tracks:

Press CTRL Z to read or write half tracks. The current drive will recalibrate and an inverse "H" will be displayed at the upper left of the screen. You may use the Read and Write commands normally, except these operations will be on half tracks. Press CTRL Z again to return to the normal full track mode.

Determining a Protected Disk's Prolog Bytes:

Many protected disks use modified Prolog bytes as part of their copy protection. The Prolog bytes are "road signs" to DOS, telling it what sector it is trying to read, and where the data begins for that sector. If they are changed from normal DOS format, copying the disk becomes very difficult.

Providing that the basic format of the protected disk is 16 sectors per track (DOS 3.3 uses the format), you can usually copy the disk to a normal DOS format by using the **Alter Prolog Bytes** option of the Main Menu, and then by copying the disk using option 2, **Copy a Disk.**

But before you can do this, you must determine what the Prolog bytes have been changed to on the protected disk. The best way to do this is to use the Nibble read option of the Sector Editor.

First, put the protected disk in the disk drive and using the "T" command, change the track to the track desired (a good starting track is $03). Press "N" again for the nibble read, and press the SPACE BAR to halt the track dump.

What you are viewing is the contents of the entire track 03, in a raw format. This differs significantly from the standard Sector Editor display in that the data is raw: this is how the data is stored on the disk.

We are looking for the address and data Prolog bytes. The address Prolog bytes tell DOS what track, sector and volume it is trying to read (this is called "soft sectoring"). Next there are the data Prolog bytes which tell DOS that the data for the sector begins here.

The normal address and data prolog bytes are $D5 AA 96 and $D5 AA AD, repectively. But the copy protection has changed these to something else, and they probably won't be too easy to identify to the inexperienced. Luckily, most only change one byte in the sequence, so this helps.

To help locate the Prolog bytes, look for a series of FF's. These are "Sync Bytes", which act as fillers between the data and address fields. Look right after the FF's for the Prolog bytes. It is easy to distinguish between data and address Prolog bytes since the address field only contains eight (8) bytes of information following it, and then the Epilog bytes (Epilog bytes, $DE AA EB in normal DOS, and tell DOS that the data stops here). Next, several more Sync Bytes ($FF) follow.

Then the data Prolog bytes follow, and 342 bytes of information. This is the actual data that get read into memory. The 342 bytes ends up as 256 bytes in RAM memory when DOS reads them. This is called 6+2 encoding (assuming DOS 3.3 format), and has to do with disk drive storage techniques. After the 342 bytes of data comes the Epilog bytes again. Then the whole sequence repeats for the next sector in that track.

To get familar with finding the address and data Prolog bytes, it is suggested you do some nibble reads on a normal DOS 3.3 disk. Look for the FF's (Sync Bytes), and then the 3 byte sequence of $D5 AA 96, or $D5 AA AD, which are the Prolog address and data bytes, respectively. In a picture here is the format:

FF FF FF FF D5 AA 96 FF FE AA AA AA AA FF FE DE AA EB
   (1)   (2)   (3)   (4)   (5)   (6)   (7)

FF FF FF FF D5 AA AD -342 bytes- XX DE AA EB
   (8)   (9)   (10)   (11) (12)

1.   Sync Bytes (4 bytes in above example, but varies).
2.   Address Prolog Bytes (3 bytes)
3.   Disk Volume Number (2 bytes)
4.   Sector Number (2 bytes)
5.   Track Number (2 bytes)
6.   Check Sum to verify disk integraty (1 byte)
7.   Epilog Bytes (3 bytes)
8.   Sync Bytes (4 bytes in above example, but varies)
9.   Data Prolog Bytes (3 bytes)
10.   342 bytes of Data
11.   Check Sum to verify disk integraty (1 byte)
12.   Epilog Bytes (3 bytes)

After you feel comfortable with finding the address and data Prolog bytes on a normal DOS 3.3 disk, try finding them on your protected disk. Then you may use the "Alter Prolog Bytes" option of the Main menu, and copy the disk to normal DOS format. From there you may edit, change or copy the disk as needed.

Chapter 9: Using the NMI and Resurrect Program.

//E USERS PLEASE NOTE: To use your Senior PROM's NMI functions to their fullest, you must have 128k (an Extended 80 Column Card) installed in your Apple //e. These are available from any Apple dealer. They are also available from other sources for under $50.00 (check the "Computer Shopper").

What Do I Use an NMI for?: The Senior PROM NMI will allow you to interrupt any program running in the main 64k of RAM, save it to a normal DOS 3.3 disk, and restart the program any time you wish. This effectively circumvents many copy protected disks with a minimal of effort by copying memory, instead of the copy protected disk itself.

Description: A "NMI" is a Non-Maskable Interrupt, and like the name indicates, it "interrupts" a program no matter what it's doing. Technically, the NMI line is pin 6 of the 6502 microprocessor, that when momentarily connected to ground, does an interrupt.

The beauty of the NMI is that during the interrupt, all the conditions of the 6502 microprocessor are saved on the stack, as is the Program Counter. To restart an interrupted program, all you must do is reload the registers (Accumulator, X-reg, and Y-reg), the stack pointer, and do a "Return from Interrupt" (which is the 6502 mnemonic RTI). This will restart the program from where it was interrupted. The NMI proves very valuable when trying to backup copy protected programs. We can allow a program to load from a copy protected disk and into memory, interrupt it, and save memory to a normally formatted DOS 3.3 disk. Then at any time we can reload our saved program from the normal DOS disk and restart the program. This is the theory behind most of the copy cards (Wildcard, Replay card, etc.). In addition, you can save the interrupted program in RAM, view or change the code, and restart it.

If you wanted to perform a NMI on a normal Apple, you could crudely do it by inserting a switch that crosses pin 29 (NMI) and pin 26 (ground) of any of the peripheral card slots. This will perform a very crude NMI because your NMI would not be "de-bounced".

To elaborate, when you press the switch to perform the NMI, the contacts of the switch actually slam against each other many times before coming to rest together (this happens VERY fast). This has the effect of doing many NMI's, where each NMI interrupts the previous NMI, and so forth, until the switch contacts come to rest (remember, computers are FAST). Now the stack (where the program counter and 6502 status word are stored) is distorted from doing the mulitple NMI's. This makes doing a RTI (Return from Interrupt) difficult, if not impossible. Hence we use a "de-bounced" NMI switch.

The de-bounced NMI switch will do one NMI, regardless of how many times the switch contacts slam together in that mili-second. This will give you a good clean NMI, allowing predictable RTI's. The de-bounced NMI switch requires the use of a "chip", in conjunction with the switch.

You have overcome the hardware aspect of a NMI with a debounced switch. Now you have a software problem. Just like the RESET key, the NMI is programmable in the Apple. Whenever you do a NMI, the NMI is executed and then program control is passed through location $3FA and $3FB.

Because of this, software publishers often "trap out" the NMI so that location $3FB points to a routine that wipes memory and reboots. Thus your effort at using a NMI are defeated.

Introducing the Senior PROM NMI features.

The Senior PROM has been specially modified for NMI's. Instead of always jumping through locations $3FA and $3FB after the NMI, you have a variety of options to choose from (much like the Senior PROM reset functions) after pressing the NMI button:

Key Pressed After Reset	Resulting Action
DEL	Enter the Monitor via $FF59. (i.e. old style Monitor reset).
=	Toggle between page 1 and page 2. Strictly for viewing, has no effect on memory.
–	Toggle between text and graphics mode. Strictly for viewing, has no effect on memory.
0	Toggle between lo-res and hi-res page. Strictly for viewing, has no effect on memory.
9	Toggle between text and graphics, and full graphics.
8	Move main upper 16k RAM memory (the language card) to accessible main RAM memory at $2000-5FFF.
7	Move auxiliary upper 16k RAM memory to accessible main RAM memory at $2000-5FFF.
6	Move main RAM from $00-8FF to $2000-28FF.
5	Move all 64k of main motherboard RAM memory to the auxiliary RAM card memory (the extended 80 column card).
4	Prepare to deactivate the Senior PROM (by flipping the toggle switch) and re-enter the normal //e environment.
3	Go directly to the Sector Editor.
2	Go to the Secondary Menu, the Memory Management menu.
1	Go to the Main Menu, the RWTS Utilities.
ESC	Reboot the disk drive via $C600 without disturbing any memory.
A	Alter an Auxiliary memory location.
B	Bank select upper 16k RAM.
C	Clear memory from $0100-BFFF and reboot the disk drive via $C600.
D	Disassemble 10 instructions before and after Program Counter.
G	Goto a specified address on Restart (instead of restarting at current running address).
Q	Quick Restart from NMI.
R	Restart from NMI.
S	Stack display.
W	Wait to deactivate Senior PROM before restarting.
X	eXchange program in Auxiliary memory with program in Main memory, and restart.
CTRL P	Print an image of Text Page 1 on the printer in slot 1 (//e only).

Many of these options are also available from Reset, and have the same functionality when used with the NMI (refer to chapter 5 for a discussion of options **DEL** to the **ESC** key).

The Transparent (middle) Switch Postion and the NMI.

The Transparent toggle switch position is very important when using the NMI. When the toggle switch is moved from the deactive to the middle position, the Senior PROM remains in the deactive state until the NMI button is pushed. This keeps the Senior PROM deactivated (and the Standard Apple ROM code active) until the moment when you press the NMI button. This is very important for software that checks your ROMs for unmodified code, and for programs running in Basic.

For this reason, all programs should be booted with the Senior PROM deactivated. Then when using the NMI to interrupt the program, move the switch to the middle position and press the NMI button.

NMI'ing a Program using up to 64k of Main Memory.

At anytime, you may move the Senior PROM toggle switch from the deactivated to the middle position and press the NMI push button to interrupt the program currently running in memory. After doing this, your Apple should "beep" and enter the Wait Mode, with acomplete menu of options available, much like Reset (on the //c the screen will "flash"). Note there are a few rare cases when the NMI button will not freeze a program. See the section entitled "What to do if the NMI Button Does Not Interrupt a Program" for a further discussion.

Saving the Program Currently in Main 64k memory to Auxiliary Memory.

After pressing the NMI switch, the first action should always be to press the **5** key (the only exception to this is when using the eXchange command, or Reverting, both described later). This will move all of Main 64k to Auxiliary memory (the Extended 80 column card in the //e). To notify you that this action has occurred, an flashing **S** will appear in the lower right hand corner of your monitor screen. After this, the Senior PROM will return to the wait mode, ready for your next command.

=, -, 0, 9: Displaying the Proper Viewing Page.

Using the **=**, **-**, **0**, or **9** keys, you may view any combination of the hi-res and text pages. This is helpful for viewing any hidden information, or to determine what view page the program was using when interrupted. This information will be needed when restarting the interrupted program. Note that using many of the other NMI features will overwrite text page 1. Because of this, right after pressing the **5** key is a good time to examine any of the viewing pages, especially text page 1:

    = Toggle between page 1 and page 2.
    - Toggle between text and graphics pages.
    0 Toggle between lo-res and hi-res pages.
    9 Toggle between mixed text & graphic, and full graphic page.

D: Displaying the Program Counter and a Disassembly of Associated Code, and Disply the Registers.

After moving all of Main memory to Auxiliary (by pressing the **5** key), you may wish to see exactly where the program was running when you pressed the NMI button (know as the Program Counter). This information can be invaluable, for example, when trying to find exactly where in memory a nibble count is running. This is also invaluable when debugging your own program written in 6502 when it "jumps to never, never land".

Pressing the **D** key will display in inverse text exactly where the program was executing when the NMI button was pressed. Also displayed is a disassembly of (approximately) ten instructions before and after the program counter. This gives a quick idea of what the program was trying to execute when you interrupted it. The bottom of the screen will give a display of the registers at the time of the NMI. NOTE YOU ARE VIEWING THE PROGRAM LAST SAVED TO AUXILIARY MEMORY USING THE 5 KEY!

S: Veiwing the Stack - What Subroutines have been Executed.

Pressing the **S** key will display the contents of page 1 in the vincinity of the Stack Pointer. The Processor Status Word, the low byte of the Program Counter, and the high byte of the Program Counter are display in inverse. Note the contents of the Program Counter, registers, Processor Status Word and Stack Pointer are displayed at the bottom of the screen. Here is an example screen:

		Comments (not shown on screen)
01E3:	E0	Contents of location $7FE
01E4:	F2	Contents of location $7FF
01E5:	93	Value of Y-register
01E6:	1F	Value of X-register
01E7:	42	Value of Accumulator
01E8:	**BB**	**Value of Processor Status Word**
01E9:	**05**	**Low byte of program counter**
01EA:	**0A**	**Hi byte of program counter**
01EB:	08	Low byte of last subroutine return address
01EC:	22	Hi byte of last subroutine return address
01ED:	AA	Low byte of next to last sub. return address
01EE:	41	Hi byte of next to last sub. return address
01EF:	7F	
01F0:	90	
01F1:	EF	
01F2:	90	
01F3:	88	
01F4:	09	
01F5:	83	
01F6:	09	
01F7:	91	
01F8;	0A	

```
  AT NMI:
  PC=0A05  A=42  X=1F  Y=93  P=BB  S=E8
```

From this display you can tell the program was running at location $0A05 when interrupted, and that the program will return to locations $2209, 41AB, 9080, 90f0, 0989, 0983 and 0A92 as successive RTS's (Return from Subroutine) instructions are encountered. Note too, that the routine that was executing at the time of the NMI was called by a JSR (Jump Subroutine) instruction located at $2206.

Remember, the return address on the Stack is always one less than the actual address to which the program will return upon an RTS. Note that page 3 of Main memory is used as a buffer for this option, thus overwritting any existing code there. REMEMBER YOU ARE VIEWING THE LAST PROGRAM SAVED IN AUXILIARY MEMORY USING THE **5** KEY.

R, Q: Restarting an Interrupted Program.

After using the NMI to interrupt a program, and pressing the **6** key to save the program to the Auxiliary 64k, you can easily restart the program one of two ways:

1. **Total Restart:** Pressing the **R** key will do a total restart, moving all of Auxiliary memory back to Main memory and restarting your program.
2. **Quick Restart:** Pressing the **Q** key will do a "quick" restart of your program. This will move Auxiliary memory $00-08FF to Main memory, and restart your program.

The Quick restart moves only pages $00–8FF from Auxiliary memory back to Main memory. This lets you enter the Monitor to change and view memory from $900–BFFF, go back to the Senior PROM wait mode (via Reset or NMI), and restart your program. Doing this allows the effects of your changes to memory to be viewed when the program restarts.

The Total Restart moves all of Auxiliary 64k memory back to Main memory and restarts your program. This has the advantage of "reverting" back to where the program was interrupted and last saved to Auxiliary memory using the 5 key.

For example, say you are playing an arcade game, and you have reached the highest and most difficult level. Before starting the level, you press the NMI button, press the 5 key, set the viewing page, and do a Quick restart. Moments later, you are killed by an overbearing alien. You may now interrupt the program (with either Reset or NMI), set the viewing pages, and do a Restart (note the 5 key is not pressed). The game is resurrected to the point you originally NMI'ed and saved the program, right before starting the new level and being destroyed by the alien. You may practice this level over and over if desired, from this point, using the Senior PROM restart.

A: Altering Memory While in the Wait Mode.

Using the Quick restart option allows a user to easy modify memory between $900–BFFF by entering the Monitor and changing the appropriate locations. The program may then be restarted.

You may use the Alter command while in the Wait Mode to alter memory when doing a total Restart where the whole program resides in Auxiliary memory, or if memory between $00–8FF needs to be changed. Pressing the A key while in the wait mode produces an "@" prompt. You may now enter the location to be changed, followed by pressing RETURN. A ":" prompt will appear. Enter the value to change this location to, and press RETURN. This will Alter the appropriate location in Auxiliary memory so when the program is moved back to Main memory and restarted, the altered memory is intact.

G: Forcing the Program to Goto a Predefined Location.

When restarting a program, you may force the program to Goto a specified address, instead of the address the program was interrupted at. Pressing the G key will produce a "G" prompt to appear on the screen. Enter the address desired where the program will Goto when restarted, and press RETURN. All the registers will be reloaded just as they were when the program was interrupted, but instead of going to the address where the program was running at, it will Goto the address specified.

Note that after entering the address the program will Goto when restarted, you may use the "D" command (Disassemble) to view ten instructions before and after the address you specified with the G command.

Using the G key to Disassembling Volatile and Other Memory Areas.

A common protection practice is to put copy-protection code over text page 1 (memory from $400-7FF). This way, if the program is interrupted (via Reset or NMI), the whole text page would scroll up at least one line, and any typing would further destroy this memory area. Using the Disassemble command, you may see exactly what the code was doing in this volatile memory area, or any other memory area from $200-BFFF without exiting the Senior PROM wait mode.

To do this, press the **S** key, and make a note of the current Program Counter. Now press the **G** key and enter the location you wish to disassemble. Finally, press the **D** key to see the disassembly.

When you are done disassembling memory, press the **G** key again, and enter the original value of the Program Counter that you previously noted. If you don't do this step the program will restart at the last address entered with the **G** command. Also remember you are viewing a disassembly of Auxiliary memory, so all disassemblies will be of the program that was last saved to Auxiliary memory (using the **5** key). Also note that memory from $00-1FF can not be disassembled.

B: Turning on the Proper Bank Switched Memory (BSM) when Restarting.

When restarting some 64k programs, it may be required to set the Banked Switched Memory (the upper 16k of the 64k Main memory) in its proper mode. Pressing the **B** key, followed by a number between 0 and 4, will select the 16k BSM in the following manner when restarted:

0 =   Deselect BSM (equiv. to C081 N C081 from the Monitor). Note this is the default if **B** is not selected.
1 =   Bank 1 of BSM read enabled, write protected (C080).
2 =   Bank 1 of BSM read enabled, write enabled (C083 N C083).
3 =   Bank 2 of BSM read enabled, write protected (C088).
4 =   Bank 2 of BSM read enabled, write enabled (C088 N C088).

When you use the Bank option, an inverse **B** will appear at the bottom right of the screen, followed by the number (0-4). This signifies which configuration of the BSM will be active when the program is restarted.

Note that you may not want to use this option all the time, but only when absolutely necessary. The reason being is the **B** option overwrites memory from $280-294 (20 bytes). If your program uses this memory area it might cause unpredictable results.

W: Forcing the Program to Wait till the Senior PROM is Deactivated before Restarting.

Many times a program will not properly restart if the Senior PROM is activated. An example of this is a program running in Basic, or some graphics programs that use Basic's graphics routines. Since both the Basic code and the Senior PROM code can not be active at the same time, a Basic program must be restarted with the Senior PROM deactivated.

This is easily accomplished using the **Wait** option. By pressing the **W** key, the Senior PROM will wait until you deactivate it using the toggle switch before restarting the program. Note when you use the Wait option, a flashing **W** appears at the bottom right of your screen signifying the Wait option is active. Then when **R** or **Q** are pressed to restart the program, the Senior PROM will "beep". This signifies you may now deactivate the Senior PROM, and the program will restart.

Note that you may not want to use this option all the time, but only when absolutely necessary. The reason being is the **W** option overwrites memory from $280-28D (13 bytes). If your program uses this memory area it might cause unpredictable results.

What to do if the Program Does Not Restart.

When Restarting a program, a variable may cause the program not to work properely. This could be attributed to several variables such as:

1. Does the program use the Basic ROM code?

2. Does the program require the BSM to be configured correctly?

If the program requires the Basic ROM code or an exact image of the all your Apple //e or //c's ROM code, use the Wait option before restarting the program. This will require you to deactivate the Senior PROM (and activate the normal ROMs) before the program will restart.

Some programs use the Banked Switched Memory (BSM) and require it to be configured properely before the program will restart. Use the **B** option to configure the BSM.

Note that you may try and restart the program as many times as you want until you have determined the correct restart options. An unaltered image of the program is in Auxiliary memory, so attempting to Restart a program more than once (with or without different options) will not destroy the program stored in Auxiliary memory in any way, providing you do not press the "5" key from the Wait Mode.

What to do if the NMI Button Does Not Interrupt a Program.

Some copy-protection schemes copy their own image of the F8 Monitor ROM into the upper 16k memory bank, and activate the RAM card (which deactives the mother board ROMs and hence deactivates the Senior PROM). This can be problem when trying to NMI a program using this protection, since the NMI will jump through the vectors pointed to by the F8 image in the RAM card, instead of the Senior PROM, thus the NMI doesn't go to the Wait Mode.

The way around this is to boot your protected program with the Senior PROM toggle switch in the deactivated position. When you want to NMI the program, move the Senor PROM toggle switch to the middle (transparent) position and press the CTRL and RESET keys without letting the CTRL and RESET keys up. While you are holding down the CTRL and RESET keys, press the NMI push button switch. Let go of the CTRL and RESET keys, and then let go of the NMI button. You may now continue your NMI procedure as normal (press the "5" key, etc.), as you should be in the Wait Mode.

X: Exchanging Two Programs in Memory.

Using the Senior PROM you may have two different 64k programs active in memory at one time, and you may alternate instantly between the two at will, and as many times as you like. This allows you to have any two 64k programs in memory at once, such as two business applications (i.e. spreadsheet and graphics), or between DOS 3.3 and ProDOS. This essentially offers you two completely different Apple computers available by pressing a button and a couple keys.

To exchange two programs in memory, do the following:

1) Boot your the first program normally with the Senior PROM deactivated.

2) When desired, interrupt the program by moving the toggle switch to the middle position, and pressing the NMI button.

3) Press the **5** key to save Main memory to Auxiliary memory.

4) Press the **ESC** key to reboot the disk drive, and then immediately deactivate the Senior PROM.

5) Insert the next program to boot in the disk drive and allow it to load.

6) When you want to exchange the previous program with the current program, move the toggle switch to the middle position and press the NMI switch (DO NOT press the 5 key this time!).

7) Using the screen select keys (9 thru =), select the appropriate screen to view for the other program (the one not currently running in memory).

8) If the Wait option is needed, press the **W** key.

9) Press the **X** key to **eXchange** the program currently in memory with the program saved in Auxiliary memory. In a approximately four (4) seconds, the first program should reappear and restart. Note if you used the **W** option, your Apple will beep and then you may deactive the Senior PROM and the programs will exchange.

You may exchange the previous program with the current program again and again, as much as desired. Simply repeat steps 6 to 9.

Exchanging Between DOS 3.3 and ProDOS.

It is very easy to exchange between these two operating systems. Just follow the above nine steps using DOS 3.3 and ProDOS. When exchanging between the two, use the **W** (wait) option (No bank selection is necessary for ProDOS).

Note that the eXchange option overwrites memory from $280-294 (20 bytes). If either of the two programs use this memory area, it might cause unpredictable results.

NMI'ing a Program and Creating a Resurrection Disk.

An excellent way to back-up a copy protected disk is to use the NMI option of the Senior PROM. Using the Senior PROM NMI (Non-Maskable Interrupt), you can stop a program currently in memory, examine and change memory, and then restart it from the point it was interrupted. In addition, you can save the program to a normal DOS 3.3 disk, and restart ("resurrect") it from disk at any time. This allows you to circumvent many copy protected disks and maintain a normal DOS version, as you are copying what is in memory, and not the protected disk.

An overview of how to do it: With the Senior PROM <u>deactivated</u>, load a copy protected program into memory. Then move the Senior PROM toggle switch to the <u>middle</u> "transparent" position and push the NMI switch (the push button switch right next to the Senior PROM Toggle switch). This "interrupts" the program, and it will freeze. Now push the "5" key, which will move all 64k of main memory into the Auxiliary 64k memory bank (an Extended 80 Column Card is required in the //e). At this point you may enter the Monitor, change memory, display a disassembly of where the program is running, list the Stack to display how the program got to its current running address, restart the program, or boot the Senior PROM Utilities disk and save the program on a normal DOS disk and later restart it.

Saving the program on a normal DOS has the advantage that you now have a **normal DOS 3.3** version of your protected program, which you can easily copy. When booted, this disk will load memory and "resurrect" the program exactly as it was when you pushed the NMI switch.

The Senior PROM provides you with 3 different disk-based Resurrect options. The reason there are several different options available is to provide flexibility in running the "resurrected" program on different Apple computers, and so you don't save more memory than you absolutely have to, thus saving disk space and loading time. These options are:

- 48k Resurrect (64k Apple required)
- 64k Resurrect (128 Apple //e, //c required
- 128k Resurrect (128 Apple //e, //c required

Each one of these Resurrect programs will be described.

There are, of course, some limitations to using this technique in backing-up copy protected disks. Please see the section "NMI Limitations" at the end of this chapter for a full discussion of these limitations.

Procedure for creating a 64k Resurrect disk.

<u>Step 1:</u> Activate your Senior PROM switch, press CTRL - RESET, and press the "C" key to clear memory and reboot the disk drive. This will write a test pattern into all 48k of Main Memory. Alternatively, you press CTRL - RESET, "2", "7", and then ESC to write a test pattern into all 64k of memory and reboot.

<u>Step 2:</u> Deactivate your Senior PROM switch and insert your protected program in the disk drive.

Step 3: After the program is loaded, and at a time you feel is appropriate, move the Senior PROM switch to the <u>middle position</u> and then press the NMI push button switch. Note you may move the switch to the middle position at any time before you want to interrupt the program. The computer will beep and the program should now freeze. Note on the //c, the video screen will "flash", indicating an NMI has been preformed. You may press the 9 thru = keys to change between the viewing screens. This may help you determine which screen would be best for the resurrect disk.

Step 4: Press the "5" key. A moment later, an inverse "S" will appear at the lower right of your screen, indicating all of Main memory has been saved to Auxiliary memory. Now put your Senior PROM Utilities disk in drive 1 and press the ESC key to boot the disk drive. As the disk drive recalibrates and boots, deactivate the Senior PROM.

Step 5: After the Utilities disk is booted, choose the option **Save Auxiliary 64k RAM Memory.**

Step 6: Put a blank disk in either drive, and when prompted by the program **Aux Saver,** enter the desired disk drive number (1 or 2). Your blank disk will be initialized and all of the Auxiliary 64k will be saved to this disk.

Step 7: After the files are saved, put the Senior PROM Utilities disk back in the drive and type "RUN HELLO" (or re-boot the Senior PROM Utilities disk).

Step 8: Choose the option **Create a Self-booting Resurrect Disk.** Next you will be prompted for the Resurrect program option you want. Choose the option **64k Resurrect (128 Apple //e, //c Required).**

Step 9: Select the screen options you would like when you boot the resurrect disk. For most games, choose **hi-res, page 1, full page.** For a program that text was on the screen, try choosing **text, page 1.** If these are not correct when you boot the disk, you may repeat steps 7 to 9 and specify different viewing screens.

Step 10:   When prompted, put the disk in the drive that was just created by the **Aux Saver** program (the disk with the memory files on it).

You now have a self-booting 64k resurrect disk. To use this disk, insert the disk in the drive and boot it in the normal fashion. The program will be re-loaded and resurrected from the point you interrupted it.

Resurrect Disk Operating Description (Referring to Previous 10 Steps):

The first step preformed (CTRL - RESET, "C") will copy a "test pattern" into all 48k of main RAM (or all 64k if you press "7", "2", "ESC" after CTRL - RESET). This test pattern is a single byte copied repeatedly to each page of memory (i.e. page $1F gets filled with $1F's, page $20 gets filled with $20's, etc.). This step is not absolutely necessary, but if you do any "memory snooping" through a program you will appreciate this step. It allows you to easily tell what memory sections are not used (loaded into) by the program, by identifying these test patterns.

Next the protected program is booted with the Senior PROM deactivated. After the program is booted, the Senior PROM switch is moved to the middle position and the NMI push button switch is pushed. Timing may be critical when pushing the NMI button to "catch" the program at the desired spot. You may have to reboot the program a few times to "catch" it right where you feel appropriate. Note that timing is unimportant when moving the switch to the middle position, just so long as its before you push the NMI button.

After pressing the NMI switch, the computer will beep and the program should freeze, and the Senior PROM has taken control of your computer. It is now waiting for your next action.

Right after the program freezes, press the **5** key to copy all of Main memory to Auxiliary memory. To see what viewing pages you would like to run the program under (after resurrection), press the 9, 0, -, or = keys. These are toggle switches and will switch the viewing screen between page 1 and page 2, text or graphics, and lo-res or hi-res graphics, respectively. They have no effect on any memory. You should make a note of the correct viewing page as you will be asked this information when creating a self-booting Resurrect disk.

Now you may press the ESC key to reboot the disk drive. Deactivate the Senior PROM just after pressing the ESC key, as the Utilities disk will not completely boot without it decativated.

Now the Senior PROM is deactivated and the Utilities disk is booted. The Utilities disk has not disturbed your program in the Auxiliary 64k RAM. Picking the menu option that runs the program "Aux 64k Saver" will save all the contents of the Auxiliary 64k RAM to a normal DOS 3.3 disk in 3 files. The file names of these files reflect what memory regions they contain:

- $00-85FF: Contains Main Memory from $00 to $85FF.
- $8600-BFFF: Contains Main Memory from $8600 to BFFF.
- $D000-FFFF: Contains memory from $D000-FFFF of Main banked switched memory bank 1, and memory from $D000-DFFF of Main banked switched memory bank 2.

Finally, using the **Create a Self-Booting Resurrect Disk** option of the Senior PROM Utilities disk will create a "Resurrect" program and save it to the disk with all the above memory file. You may now reboot this memory file disk and it re-run the interrupted program from the exact point you stopped it.

How Resurrect works: The file "Resurrect" on your self booting resurrect disk BLOADs the appropriate memory files, and moves them to their appropriate places. After all memory sections are put back in their correct places, the registers are reloaded, and the correct viewing page is turned on. Then a "Return from Interrupt" (RTI) is executed. This brings the program back to life (resurrects it), and it continues to run from exactly where it was originally interrupted.

For a more detailed description of the interworking of each of the Resurrect programs and the program **Aux Saver**, the "Merlin" source code files have been provided on the Senior PROM Utilities disk (the files are suffixed with a ".S").

Three Different Resurrect Programs for Flexibility.

1) **The 48k Resurrect:** The procedure for making a 48k Resurrect is exactly the same as previously described for the 64k Resurrect, except you use the 48k Resurrect program instead of the 64k Resurrect program. The 48k Resurrect program will only replace the lower 48k (memory from $00-BFFF) of the original program. Because of this it is recommended that you try the 64k Resurrect before trying the 48k Resurrect.

The difference is that only the files $00-85FF and $8600-BFFF are required for the 48k Resurrect, and that it can be run on any Apple with a minimum of 64k. This will allow you to run the Resurrect disk on a variety of Apples, take up less disk space, and have a faster loading time.

After you use the program "Aux Saver" to save all 64k of memory, you may delete the file $D000-FFFF from the Resurrect disk. Then use the option from the Senior PROM Utilities disk to "Create a self-booting Resurrect disk", and select the 48k Resurrect menu pick. You will be prompted when to insert the disks created by **Aux Saver** in the drive, and the 48k Resurrect program will be placed on this disk.

It is advisable to make sure the 48k resurrect works properly first, before deleting the file $D000-FFFF from your disk. If the program does not work correctly, try using the 64k Resurrect program. You may do this by using the Senior PROM Utilities disk option "Creating a Self-Booting Ressurect Disk", and then select the 64k menu pick.

2) **The 64k Resurrect:** This will be your most often used Resurrect, and requires the least effort to create. The steps outline in **NMI'ing a Program and Creating a 64k Resurrect Disk** describes exactly how to make a 64k Self booting Resurrect disk.

3) **The 128k Resurrect:** This Resurrect should only be used when the other Resurrects have failed, or if you know that the Auxiliary 64k RAM is being used by the program. You may often determine the memory requirements for a program by reading the program's packaging or instruction material. The procedure for creating a 128k Resurrect disk starts out the same as for a 64k Resurrect. This means if you are not sure if the program requires 128k, try the 64k Resurrect first.

Note that in 128k Resurrects only Auxiliary memory from $00-2FF and $400-BFFF are restored (the lower 48k). Auxiliary Bank Switched Memory from $D000-FFFF and Page $03 are used by the 128k Resurrect program. All 64k of Main memory is restored.

Here are the necessary steps to produce a 128k Resurrect:

1) Put aside the disk created for the 64k Resurrect.

2) Boot the original protected disk with the Senior PROM deactivated.

3) At approximately the same time as you NMI'ed the program previously, activate the Senior PROM and press CTRL - RESET.

4) Now press the DELETE key to enter the monitor.

5) Deactivate the Senior PROM and type **6 CTRL P RETURN** and boot your Senior PROM Utilities disk.

6) Select the option to run **Auxiliary 48k Memory Saver.**

7) When prompted put the Resurrect disk you created with the 64k memory files in the drive, and specify the drive it is in. The Auxiliary 48k memory will be saved into 2 files, **$00-85FF.AUX** and **$8600-BFFF.AUX**.

8) Now put the Senior PROM Utilities disk back in the drive and type "RUN HELLO" (or reboot the Senior PROM Utilities disk.

9) Select the option for Creating a Self-Booting Resurrect Disk, and then choose the `128k Resurrect (Apple //e, //c required)`.

10) Enter the screen choices you want, and when prompted put the Resurrect disk (with all the memory files on it) in the drive.

You have now created a 128k Self Booting Resurrect disk.

NMI Limitations and Resurrect Notes:

Althought the NMI is very easy to use, it does have some limitations that should be observed. Here is a list of programs that the NMI **will not** work on blindly, or without considerable thought:

- Some programs requiring the Auxiliary 64k RAM (Extended 80 Column Card on the //e).
- Programs that use multiple disk access.
- Programs using the 80 column text mode.

Programs Using the Auxiliary 64k RAM (128K): It is possible to NMI and resurrect most programs using the Auxiliary 64k RAM, depending how the extra 64k is being used and when the program is interrupted with the Senior PROM NMI.

The Senior PROM should be active and the NMI button pushed when it is known (or assumed) that the program is running in Main memory, instead of Auxiliary memory. If the program is using the Auxiliary memory just for work space (many spreadsheet programs do this), you will have no problems NMI'ing the program and using the 128k Resurrect program.

Also many Double Hi-res programs can be easily NMI'ed. Interrupt the program when its assumed to be in main memory, not when the program is drawing a double hi-res page. It would be better to interrupt it before it starts accessing the double hi-res graphics, or while a text screen is present.

If the program uses the Auxiliary banked switched memory, the Senior PROM will not be able to resurrect it. The Auxiliary BSM is used by the 128k Resurrect program, and therefore can not be restored.

Programs using Mulitple disk access: As with all the copy products that copy memory, saving a multiple disk access program is a problem. This is because the **whole** program is not in memory at one time, only pieces are loaded (remember you are copying memory, not the disk).

For example, many games load in from disk for each level you progress. You may interrupt the game at the particular level you are playing at, but when you advance to the next level the program will try and load in the next level from disk. Of course your disk will not contain the next level, and the program will bomb.

One way around this is to use your resurrect disk to save and load the current level you are playing, and then insert your original disk when it is required. This will save some wear from your original disk.

You may also refer to the documentation **Deprotection Methods and Techniques Using the Senior PROM** to try and remove the copy protection from the disk all together. This is the optimal method, but it requires more energy than simply using the NMI method.

Programs Using the 80 Column Text Screen: Due to hardware limitation imposed by the Senior PROM, a program can not be restarted in the 80 column text screen mode.

The Resurrect Disk Program Notes.

"Merlin" source code is available on the Senior PROM Utilities disk. All source files are suffixed by ".S".

You may use a file transfer program like FID from the DOS 3.3 System Master to transfer the Resurrect files from the Senior PROM Utilities disk to your Resurrect disk. But there will be a noticable different between this Resurrect program and the one created by the **Create a Self-Booting Resurrect Disk** from the Utilities disk.

The Resurrect program transfered to your Reserrect disk by the Senior PROM Utilities has the viewing screen options hard-coded into it. This means everytime you boot the Resurrect disk, the screens you choose when creating the self-booting Resurrect disk will always be displayed, without prompting.

If you transfer the Resurrect files directly from the Utilities using FID, you will be prompted each time the disk is booted for the desired viewing screens. This will enable you, if unsure, to easily determine which viewing screen you feel is best for the program. Then you may run the **Create a Self-Booting Resurrect Disk** option using the Senior PROM Utilities disk and this will automatically turn on the viewing screens you specified each time the Resurrect disk is booted.

A final note on using FID to transfer the Resurrect programs to your disks. There are 3 Resurrect programs on the Utilities disk, **RESURRECT 48K**, **RESURRECT 64K** and **RESURRECT 128K** (note the source files are suffixed by a .S). Transfer the Resurrect program needed, then use the DOS RENAME command to rename the file to **RESURRECT**. This is the filename looked for by DOS when booting the self-booting Resurrect disk.

Chapter 10: Other Senior PROM Features.

Covered in this chapter are the Senior PROM Mini-Assembler, Step and Trace functions, On-the-Fly Interrupts, and using DOS commands from within the Senior PROM Environment.

The Mini Assembler:

The Mini-Assembler is a program within the Senior PROM monitor which allows you to type in Assembly language programs in the same format as the Monitor's List command displays. This program was originally in the early Apple II monitor ROM, and has been unavailable to many Apple users till now. It is a "mini" Assembler because it cannot understand symbolic labels, something that a full-blown assembler does (such as Merlin). Regardless, it is useful to have the Mini-Assembler available for entering small programs or editting ones currently in memory.

To enter the Mini-Assembler, activate the Senior PROM and enter the monitor. Then type:

*F666G

You will then receive the "!" prompt, which signifies you are in the Mini-Assembler. You may enter any Monitor command by preceding it with a dollar sign ($). Aside from that the Mini-Assembler has an instruction set and syntax all its own.

Before you start to enter a program, you must set the Program Counter to point to the location where you want your program to go. Do this by typing the address followed by a colon. Follow this with the mnemonic for the first instruction in your program, followed by a space. Now type the operand of the instruction. Finally press RETURN.

The Mini-Assembler converts the line you typed into hexadecimal, stores it in memory beginning at the location of the Program Counter, and then disassembles it again and displays the disassembled line on top of your input line. It then prompts for the next line. To tell it you want the next instruction to follow the first, press only the SPACEBAR, followed by the next instruction's mnemonic and operand. Press RETURN and it will assemble that line and prompt for the next.

If the line you typed has an error in it, the Mini-Assembler will beep and display a "caret" under or near the offending character in the input line. You will also get an error if you forget to press the SPACEBAR before entering the mnemonic. A common error that will be detected is a Branch that is out of range (more than 127 locations distant from the address of the instruction).

To exit the Mini-Assembler, either use CTRL - RESET or type the monitor command:

!$FF69G

Your assembly language program will stay in memory and you may list it using the Monitor's List command.

Mini Assembler Instruction Formats.

The Mini–Assembler recognizes 56 mnemonics and 13 addressing formats used in 6502 Assembly language programming. The mnemonics are standard, as used in any 6502 Assembly language manual, but the addressing formats may be different. Here are the Apple standard address mode formats:

Mode	Format
Accumulator	None
Immediate	#$value
Absolute	$address
Zero Page	$address
Indexed Zero Page	$address,X
	$address,Y
Indexed Absolute	$address,X
	$address,Y
Implied	None
Relative	$address
Indexed Indirect	($address,X)
Indirect Indexed	($address),Y
Absolute Indirect	($address)

An "address" consists of one or more hexadecimal digits. The Mini–Assembler interprets addresses in the same manner that the Monitor does: if an address has fewer than four digits, it adds leading zeros. If an address has more than four digits, then is uses on the last four.

All dollar signs ($), signifying that the addresses are in hexadecimal notation, are ignored by the Mini–Assembler and may be omitted.

There is no syntactical distinction between the Absolute and Zero Page addressing modes. If you give an instruction to the Mini–Assembler which can be used in both Absolute and Zero-Page mode, then the Mini–Assembler will assemble that instruction in Absolute mode if the operand is greater than $FF. It will assemble that instruction in Zero Page mode if the operand is less than $0100.

Instructions with the Accumulator and Implied addressing modes need no operand.

Branch instructions which use the Relative addressing mode require the **target address** of the branch. The Mini–Assembler will automatically figure out the relative distance to use in the instruction. IF the target address is more than 127 locations distant from the instruction, then the Mini–Assembler will sound a "beep" and place a caret under the target address, and ignore the line.

If you give the Mini–Assembler the mnemonic for an instruction and an operand, and the addressing mode of the operand cannot be used with the instruction you entered, the Mini–Assembler will not accept the line.

Mini–Assembler 65C02 Instructions:

With versions 1.02 and later, your Senior PROM now contains advanced features found in very few other debugging and programming development aids. In addition to all the features described earlier, the new 65C02 instructions are assembled correctly by the mini-assembler and are disassembled correctly by the monitor when the Senior PROM is activated.

Note that if you **do not** have a 65C02 microprocessor installed in your //e (you already have one if you are a //c owner), these new instructions will not perform properely and may do some very strange things!

The new instructions include:

    BRA = BRanch Always (no more SEC, BCS).
    STZ = STore Zero in memory.
    TRB = Test and Reset Bits in memory.
    TSB = Test and Set Bits (both use accumulator as mask).
    PHX = PusH X-reg on stack.
    PHY = PusH Y-reg on stack.
    PLX = PulL X-reg from stack.
    PLY = PulL Y-reg from stack.
    DEC = DECrement accumulator (without operand).
    INC = INCrement accumulator (without operand).

And new addressing modes for:

    BIT: zero page, X; absolute, X; immediate.
    JMP: Indexed indirect. i.e. JMP (34,X)

All the following have zero page, X and indirect modes. i.e. CMP $6F,X; EOR ($6E44):

    ASC
    AND
    CMP
    EOR
    LDA
    ORA
    SBC
    STA

Step and Trace.

Another program within the Senior PROM is the Step and Trace facility. The **Step** command decodes, displays, and executes one instruction at a time. The **Trace** command steps quickly through a program, stopping when a BRK instruction is executed.

Each **Step** command causes the Monitor to execute the instruction in memory pointed to by the Program Counter. The instruction is displayed in its disassembled form, then executed. The contents of the 6502's internal registers are displayed after the instruction is executed. After execution, the Program Counter is bumped up to point to the next instruction in the program.

For example, if you wanted to step through a program starting at $300, here is what would happen:

*300S

0300-   A2 02   LDX   #$02
 A=0A   X=02 Y=D8   P=30   S=F8

*S

```
0302-     B5 00          LDA    $00,X
 A=0C  X=02  Y=D8   P=30   S=F8
```

The Monitor keeps the Program Counter and the last opened address separate from one another, so that you can examine or change the contents of memory while you are stepping through your program.

Also **300SS** could have been typed to do multiple steps. One step will be performed for each **S**.

The **Trace** command is just an infinite Step command. It will stop tracing the execution of a program only when you press CTRL - RESET or it encounters a BRK instruction in the program. If the Trace encounters the end of a program which returns to the Monitor via a RTS instruction, the Trace can only be terminated by pressing CTRL - RESET. For example:

*300T

```
0300-     A2 02          LDX    #$02
 A=0A  X=02  Y=D8   P=30   S=F8
0302-     B5 00          LDA    $00,X
 A=0C  X=02  Y=D8   P=30   S=F8
```

Until you press CTRL - RESET or a BRK instruction is encountered.

On-the-Fly Interrupts when Program requires Standard Apple ROMs.

Some programs may bomb into the Monitor when the Senior PROM is activated because they require the code in the standard Apple ROMs. If your intent is to merely break out of the program, flipping the switch to the active position causes no problems, as the program will either freeze (press Reset to interrupt), or bomb into the Monitor.

If your program bombs when activating the Senior PROM, the NMI is more desireable for interrupting the program than Reset. Boot the program with the Senior PROM deactivated. Before you want to interrupt the program, move the toggle switch to the middle position. The Senior PROM will remain deactivated until you press the NMI button, at which time the Senior PROM becomes immediately activated, and the standard NMI/Reset key functions are available from the Wait Mode.

On-the-fly Resets are available, if desired. Run the program with the Senior PROM deactiveated (or in the middle position). When you want to interrupt the program on-the-fly with a Reset, press CTRL - RESET without letting up on either key. Now flip the toggle switch to the activated position, and let go of the RESET and CTRL keys. You will now be in the Wait Mode with all the Senior PROM Reset functions available.

Using DOS Commands From Within the Senior PROM Environment.

Although you can not work within the BASIC programming environment with the Senior PROM active, you certainly can use many of the standard DOS commands, assuming DOS and its associated hooks are present in memory.

To use DOS from within the Senior PROM environment (assuming you have previously booted DOS), it will probably be necessary reconnect it before attempting any DOS commands. To do this type:

*3EAG

This will reconnect DOS and enable you to use any of the DOS commands (such as CATALOG, BLOAD, BSAVE, etc.). Note you will be unable to use LOAD or SAVE in the Senior PROM environment as Basic is disconnected.

Chapter 11: Senior PROM //e and //c Compatibility.

When the Senior PROM is deactivated, your Apple will act just as you never installed the Senior PROM. This is because the Senior PROM has an exact image of the standard Apple Monitor ROMs.

When the Senior PROM switch is activated, you have a whole new set of utilities instantly available to you. These utilities do not cuase you to give up any standard // features, and APPLESOFT Basic has been disabled ONLY when the Senior PROM switch is activated.

This doesn't mean that if you are programming in Basic and need to use a Senior PROM utility, you will lose your Basic program. You only temporarily lose APPLESOFT, not your program. The reason is your Basic program lives in lower RAM, starting at $801. When you activate the Senior PROM switch, it has no affect on lower RAM: it merely switches off the standard Apple Monitor and activates the Senior PROM Monitor. When you are done with the Senior PROM, just flip the switch back and everything is back to normal. Your Basic program is intact as though nothing has happened.

Because the Senior PROM can be so transparent, you can enjoy all those functions and maintain the compatibility of a standard Apple //e or //c. But at the flip of a switch, your whole Apple //e or //c takes on a new meaning, one which you are sure to enjoy using!

The Senior PROM NMI functions have been tested for compatibility with other "copy cards" presently installed in your //e. If you own a "Wildcard" or a "Replay" card, and it is presently installed in your //e, the Senior PROM NMI function may interfer with its operation. The Senior PROM NMI push button may activate the installed copy card, instead of the Senior PROM NMI functions. If you wish to use the Senior PROM NMI functions you may have to remove the copy card from your //e. All other Senior PROM functions (except the NMI) will work properely with the copy card installed.

It is important to note that if you use the Senior PROM Utility Copy a Disk, memory from $800-95FF is used for as a buffer for the copying process. Likewise, the default memory buffer for the Sector Editor is $800-8FF. Therefore if you use either of these utilities any programs in the above mentioned memory spaces will be destroyed.

Apple //c Compatibility Notes:

The Senior PROM is compatible with all //c memory expansion cards and co-processors (such as a CP/M card). Unfortunately due to space limitations within the //c, the Senior PROM will not fit inside the Apple //c case along with any other interally mounted //c cards. To solve this problem, Cutting Edge Enterprises offers an External Senior PROM //c Adapter for $25.00 which mounts the Senior PROM outside of the //c case. This is the only alternative when the Senior PROM is used in a //c with other user installed expansion cards.

Starting in March 1987, the Senior PROM //c comes with the new 32k Apple Unidisk ROM with Apple Talk compatability (Monitor ROM part number 042-0033-A). This is the newest //c Monitor ROM available from Apple.

If you have an older //c (before November 1986) with the UniDisk Senior PROM //c installed, it is not necessary to have your local dealer upgrade your //c for use with the UniDisk 3.5 inch drives. The UniDisk Senior PROM //c has already done this for you.