



Apple II GS

#49: Rebooting (Really)

Revised by: Matt Deatherage
Written by: Matt Deatherage & Jim Merritt

January 1989
November 1988

This Technical Note discusses rebooting the Apple II GS from software.

Changed since November 1988: Corrected two assembly-language instructions in the FROMNATV routine in the example code.

In days gone by, many Apple II applications had a Quit menu option. Unfortunately, a large number of these simply rebooted the machine. Today, this is far from desirable. Even with the advantages of GS/OS-reduced booting time (around 34 seconds with an Apple 3.5 Drive), waiting for the operating system to reload, as well as wiping out any ongoing tasks by desk accessories (such as an alarm clock) makes the standard ProDOS 8 or GS/OS QUIT call much more attractive.

However, there are still instances where an application may wish to require the user to reboot. A common example might be a game. The game might use GS/OS in a completely standard way, but if you QUIT from the program GS/OS booted into, you will be returned to the same program. Since most applications will boot into the Finder, this is not a widespread problem. However, the Finder must also provide the reboot option, and alternate program selector applications may wish to provide this functionality as well.

The Easy Way

GS/OS provides a mechanism for rebooting with the `OSShutdown` call. This call, documented in *GS/OS Reference*, Volume 1, will either reboot the system (after first shutting down all loaded and generated drivers and closing all open sessions) or will shut down everything and present a dialog box which states, "You may now power down your Apple II GS safely." A Restart button is provided which allows the user to reboot without pressing Control-Open Apple-Reset.

Note: When using System Disk 4.0, if the Window Manager is active when you issue the `OSShutdown` call, there must be at least one open window; it need not be visible, but it must be open. This will be fixed in the next revision of GS/OS.

The `OSShutdown` call also provides a way to resize the internal RAM disk (named `/RAM5` by default). Most programs have absolutely no need to use this mechanism, and should avoid it whenever possible. A notable exception would be a third-party RAM disk utility which uses a

battery backup, which may need to make changes which require resizing the RAM disk. Of course, such a utility should ask the user to ensure that erasing the RAM disk content is acceptable. Resizing the RAM disk is only possible when using the `OSShutdown` call; any other method you may be using to accomplish this function from software will break in the future.

If you are using GS/OS, you should **always** use `OSShutdown`. You must not reboot the system in any other fashion. The `OSShutdown` mechanism provides a convenient and supported way to restart or shut down the system. Doing it another way can easily cause a loss of data.

The Hard Way

Programs not using GS/OS have a little more work to do. The supported non-GS/OS method of rebooting is similar to the method used on 8-bit machines: change the value of `POWERUP` (\$00/03F4) and do a long jump to `RESET` (\$FA62). However, there are a few catches:

1. The jump must be made in emulation mode.
2. Interrupts must be disabled.
3. The data bank register must be set to zero.
4. The direct page must be zero.
5. ROM firmware must be visible in the memory map.
6. Internal interrupt sources (such as the ones for AppleTalk) must be shut down.

Simply disabling interrupts without shutting down AppleTalk interrupt sources inside the system will cause the system to hang when the jump to `RESET` is made. Turning off these internal interrupt sources is accomplished by changing softswitch values at `$C039` (`SCCAREG`), `$C041` (`INTEN`), and `$C047` (`CLRVBLINT`).

The following code example demonstrates the correct method:

```

POWERUP      equ    $0003F4          ;the power-up byte in bank zero
STATEREG     equ    $C068            ;ROM/RAM state register
CLRVBLINT    equ    $C047            ;clear VBL interrupt flags register
INTEN        equ    $C041            ;interrupt enable register
SCCAREG      equ    $C039            ;SCC register
RESET        equ    $00FA62         ;ROM reset entry point
;
FROMNATV     anop                    ;enter here from native mode
              sei                    ;disable interrupts
              pea    0
              pea    0                ;push four zero bytes on the stack
              plb                    ;pull data bank register
              plb                    ;(twice to balance the stack)
              pld                    ;pull 16-bit data bank register
              sec
              xce                    ;go into emulation mode
              longa    off
              longi    off
FROMEMUL     anop                    ;enter here from emulation mode
              sei                    ;disable interrupts for people entering here
              dec     POWERUP         ;invalidate the power up byte
              lda     #$0C            ;ROM parameters
              sta     STATEREG        ;swap in the ROM and everything else out
              stz     CLRVBLINT       ;clear VBL interrupts
              stz     INTEN           ;turn off internal interrupt sources
              lda     #$09
              sta     SCCAREG         ;shut down SCC interrupt sources
              lda     #$C0
              sta     SCCAREG
              jml     RESET           ;and off we go into the wild blue yonder

```


These methods of restarting the system are presented for those applications that absolutely must do so. Rebooting is not a suggested way of ending an application and the techniques described in this Note should be used with **extreme** caution.

Further Reference

- *Apple IIgs Firmware Reference*
- *GS/OS Reference*, Volume 1