

Interrupts are effective only if they are enabled most of the time. Interrupts that occur while interrupts are disabled will not be serviced.

Pascal, DOS 3.3, and ProDOS turn off interrupts while performing disk operations because of the critical timing of disk read and write operations. Some peripheral cards used in the Apple IIe disable interrupts while reading and writing.

---

**Original IIe**

Although the enhanced Apple IIe firmware never disables interrupts during screen handling, the original Apple IIe periodically turns interrupts off while doing 80-column screen operations. The effect is most noticeable while the screen is scrolling.

---

**Important**

Don't use PR#6 to restart your Apple IIe while running ProDOS with interrupts enabled since PR#6 doesn't disable interrupts. If you try it, ProDOS will fail as it starts up since its interrupt handlers aren't yet set up. If you have to restart, use Control-Reset or make sure that your program disables interrupts before it ends.

---

---

## Rules of the interrupt handler

Unlike the Apple IIc, the enhanced Apple IIe's interrupt-handling firmware is not always switched in. Here are the reasons why this is so and the implications that necessarily follow.

There is *no* part of memory in the Apple IIe that is always switched in. Thus, there is no location for an interrupt handler that works for all memory configurations. However, the \$C3 page of firmware is present on all systems that have 80-column text cards in their auxiliary slots, so it was selected as the starting location of the built-in interrupt-handling routine.

There are two factors that determine if the \$C3 firmware is switched in and therefore whether or not interrupts will be usable:

- ☐ Is there an 80-column text card in the auxiliary slot?
- ☐ If not, is there a peripheral card in slot 3 with built-in ROM with bytes \$C305 = \$38 and \$C307 = \$18?