

## **Chapter 2     Software Developer Guidelines**

This chapter provides guidelines that will assist you in making sure that the applications you are developing will run on the Macintosh Portable computer when it is introduced. Included is a list of things you should and shouldn't do when writing your application programs. You are encouraged to pay close attention to these guidelines. If you have any questions about the list of Do's and Don'ts, immediately contact Apple Developer Technical Support (AppleLink address MACDTS) and mention the Macintosh Portable product name.   ▪

## 2-2 Developer Notes

These guidelines will help to ensure that the software you are developing is compatible with the Macintosh Portable computer. The following chapters describe the hardware and software of the Macintosh Portable and will make clearer some of the terms used in this chapter which may be unfamiliar. The following list stresses things you should be extremely careful about. The Macintosh Portable has some significant differences from the Macintosh SE, and even though the software you are developing may work on the Macintosh SE, it may not work properly on the Macintosh Portable unless you follow these rules.

1.
  - **Don't** go directly to the Power Manager trap if your software needs to put the Macintosh Portable in the sleep state.
  - **Do** use the new **Sleep** trap.

A new trap has been added to the Macintosh Portable Operating System. If you are developing a new application and it includes software that can put the Macintosh Portable in the sleep state, such as a smart alarm, you must use the Sleep trap rather than directly accessing the Power Manager trap (see Chapter 6, “The Power Manager”).

2.
  - **Don't** go directly to the Power Manager trap to set and get time.
  - **Do** use the **SetDateTime** and **GetDateTime** traps.

The clock in the Macintosh Portable differs slightly from the clock in the Macintosh SE. Use the traps to set the clock and read the time. If you try to access the Power Manager trap directly for these functions, your program will not work correctly.

3.
  - **Do** use the PmgrOp trap to access the **wake-up timer** commands. This is a new feature.

The Power Manager Op trap is currently the only way to enable the wake-up feature. If you want the Macintosh Portable to return to the operating state from the sleep state at a predetermined time, you can use this trap to access the wake-up timer command. This command enables the wake-up feature (implemented as a timer), and when the real-time clock reaches the preset value, the Macintosh Portable returns to the

operating state. If you need to use the auto-wake-up feature, contact Apple Developer Technical Support.

4.
  - **Don't** access the ADB (Apple Desktop Bus) hardware directly.
  - **Do** use the ADB traps.

Access to the ADB hardware is completely different on the Macintosh Portable than the other Macintosh computers.

5.
  - **Don't** talk directly to the SCC (Serial Communication Controller) chip.
  - **Do** make normal communications calls to the Serial Driver.

You will be taking a large risk if your program attempts to talk directly to the SCC hardware. The serial chip in the Macintosh Portable is turned off whenever it's not in use. The serial driver knows how to turn the serial chip back on if your program makes normal serial communications calls. However, software that attempts to go directly to the serial chip will wind up talking to the chip when it is turned off, resulting in a loss of communications. If you need to directly access the SCC, you should contact Apple Developer Technical Support.

6.
  - **Don't** access any hardware directly (by hardware address).
  - **Do** use the appropriate trap to invoke a manager.

7.
  - **Don't** ignore error checking in your program.
  - **Do** continue to check for errors throughout your code.

Always check for errors. You might be in the middle of a transaction and get errors that you have not previously experienced on the Macintosh SE. For example, if the Macintosh Portable goes into the sleep state in the middle of an AppleTalk<sup>®</sup> transaction, the session may have timed out when the execution of your code continues. This example should only be taken as a general warning. In the future, decisions could be made that would prevent the Macintosh Portable from going into the sleep state while programs are running.

Remember! The user can put the machine into the sleep state at any time. The sleep feature is going to cause some errors that you may not expect, and they can occur any time in the middle of any code.

8.
  - **Don't** assume any screen size.
  - **Do** make sure your program checks for screen size.

As an example, lots of older code, like AppleLink<sup>®</sup>, assumed that if it wasn't running on a Macintosh Plus it must be running on a Lisa<sup>®</sup> (Macintosh XL) and you ended up with a window that filled the Lisa screen. If you later ran this code on a Macintosh SE, you ended up with a window that was off the screen.

Do not assume that if the screen size is 640 pixels wide, it's going to be 480 pixels high like the Macintosh II. In fact, don't assume its going to be any particular size; have your code read the size from the appropriate data structure. The Macintosh Portable screen size is actually 640 pixels wide by 400 pixels high.

9.
  - **Don't** assume only one size disk.
  - **Do** be prepared to handle unusual disk sizes.

In addition to the 1.4 MB floppy disk drive, the Macintosh Portable has E-disks (electronic disks, also known as RAM disks) in a user-settable range of sizes to 2 MB. Make sure that your program is prepared to operate with unusual disk sizes.

10.                   ■ **Don't** attempt to talk to the IWM chip.

Remember! The Macintosh Portable doesn't have an IWM chip but instead uses a SWIM chip to control the functions of the 1.4 MB floppy disk drive. This is of particular importance to people doing copy protection schemes, and it will completely change the way in which copy protection works.

11.                   ■ **Don't** confuse "time" and "ticks"

On the current Macintosh computers, one "tick" is equal to 1/60 of a second in time; therefore, you can assume that if 60 ticks go by, one second in "time" will pass by. However, because of its sleep feature, this does not hold true for the Macintosh Portable. Two different timers run in the Macintosh Portable. If the Macintosh Portable is in the sleep state, "time" keeps running because of the real-time clock, but "ticks" stop until the Macintosh Portable returns to the operating state. If you attempt to use these two constants interchangeably, and the Macintosh Portable goes into the sleep state, your program may become completely confused.

12.                   ■ **Do**  
understand how SysEnvirons handles machine type.

Beginning with System 6.0.2, *all versions* of SysEnvirons may return values larger than those currently defined for machine type. Your program should treat machine types of Zero (= machine unknown), and all values larger than the current largest value, as an unknown machine. *Do Not* check just a specific set of values. For more information, see the October '88 revision of Tech Note 129.

## **2-8**    Developer Notes