

# New Technical Notes

Macintosh



®

---

Developer Support

## Power Manager Q&As Devices

Revised by: Developer Support Center

June 1993

Written by: Developer Support Center

October 1990

This Technical Note contains a collection of Q&As relating to a specific topic—questions you've sent the Developer Support Center (DSC) along with answers from the DSC engineers. While DSC engineers have checked the Q&A content for accuracy, the Q&A Technical Notes don't have the editing and organization of other Technical Notes. The Q&A function is to get new technical information and updates to you quickly, saving the polish for when the information migrates into reference manuals.

Q&As are now included with Technical Notes to make access to technical updates easier for you. If you have comments or suggestions about Q&A content or distribution, please let us know by sending an AppleLink to DEVFEEDBACK. Apple Partners may send technical questions about Q&A content to DEVSUPPORT for resolution.

New Q&As in this Technical Note:

Think C SetWUTime glue problem and workaround

Conditions for networked PowerBook not going to sleep

---

### Think C SetWUTime glue problem and workaround

Date Written: 2/1/93

Last reviewed: 4/1/93

Does the wakeup timer work on PowerBooks? At first accRun time in my driver, as a test, I do the following using Think C:

```
{
unsigned long      currentSecs;
OSErr              err;

GetDateTime(&currentSecs);
err = SetWUTime(currentSecs + 60);
}
```

I then manually sleep my PowerBook and wait for it to awake after a minute. It doesn't. Am I doing something wrong?

---

There's an error in the Think glue for this routine. It works as documented in MPW, but fails when you use it with Think C. After a little digging we found that the Think C glue is

expecting the value to be passed by reference. So the following code fragment should be correct for Think C.

```
{
unsigned long      currentSecs;
OSErr             err;

GetDateTime(&currentSecs);
currentSecs = currentSecs +60L;
err = SetWUTime((long)&currentSecs);
}
```

## **Conditions for networked PowerBook not going to sleep**

Date Written: 1/20/93

Last reviewed: 4/1/93

When my PowerBook is connected to a network, it never goes to sleep. Why?

---

There are two situations in which the PowerBook won't go to sleep, as contrasted to simply dimming the screen. For reference, screen dimming is based on user activity, and sleep is based on system activity. One reason for the PowerBook to remain awake is that AppleTalk is active *and* the PowerBook has the recharger cord plugged in. This was a design decision made on the assumption that if the recharger is plugged in and AppleTalk is active, then you're at your office and it's likely you'd want the PowerBook to be as responsive as possible. Note that the PowerBook only detects whether the recharger cord is plugged in, not whether there's actual current flowing through the recharger.

The second reason for the PowerBook not going to sleep is that AppleTalk is active and the user has activated some network service such as AppleShare or some e-mail service. The Power Manager detects that there is periodic activity at the AppleTalk port and will not allow the unit to go to sleep. Simply having AppleTalk active and plugged into a LocalTalk network will not keep the unit awake when the power cable is not connected.

## **SetWUTime documentation fix**

Date Written: 5/18/92

Last reviewed: 7/13/92

I'm trying to use SetWUTime to wake the PowerBook in one minute from the current time, as follows:

```
long time;
GetDateTime( &time);
SetWUTime( time + 60);
```

I run this code and then sleep the PowerBook but it never wakes up as it should. This seems very simple. Am I missing something?

---

Earlier documentation for SetWUTime is in error in that the time parameter takes a pointer to the long time, not the long itself, as shown below:

```
long time;
GetDateTime( &time);
time = time + 60;
SetWUTime( &time);
```

## **Power Manager BatteryStatus returns voltage & charger details**

Date Written: 10/15/91

Last reviewed: 8/1/92

How can we check if a Macintosh Portable is charging?

—

The Power Manager chapter of *Inside Macintosh* Volume VI (pages 31-23 and 31-24) describes the BatteryStatus call, which provides you with the following information:

- Whether the charger is connected (status bit 0)
- Whether the charger is in High or Normal charge
- If the battery voltage is below the warning threshold
- A power value from which you can calculate the battery's current voltage

To calculate a battery's current voltage, you'll need to average the voltage over an extended time period (tens of seconds to several minutes for good accuracy). The power load within the Portable is a dynamic environment, so the current draw of the various subsystems on the battery will affect the voltage that you read.

## **How to notify a device driver when Macintosh Portable sleeps**

Date Written: 10/30/90

Last reviewed: 8/1/92

Is there any way for a device driver to be notified when the Macintosh Portable goes to sleep or shuts down the disk?

—

Yes, there is. You can install a queue element in the Sleep Queue, and that routine will be called during a sleep or wakeup request or demand. This is described in the Power Manager chapter of *Inside Macintosh* Volume VI. The routines you'll want are

```
struct SleepQRec {
    struct SleepQRec *sleepQLink;
    short sleepQType;           /*type = 16*/
    ProcPtr sleepQProc;        /*Pointer to sleep routine*/
    short sleepQFlags;
};

typedef struct SleepQRec SleepQRec;
typedef SleepQRec *SleepQRecPtr;

void SleepQInstall(SleepQRecPtr qRecPtr)
    = {0x205F,0xA28A};
void SleepQRemove(SleepQRecPtr qRecPtr)
```

```
= {0x205F, 0xA48A};
```

The type of request (sleep request [1] sleep demand [2] wake [3] sleep request cancelled [4]) is passed in D0, and a pointer to your queue entry is passed in A0, if you need them. Most of the time you won't. You can deny a sleep request, but you *cannot* deny a sleep demand, since this could be caused by a seriously low battery, and your denying it could damage the user's hardware. As a general rule, you should allow both requests and demands. You accept the request by clearing D0 to 0, and deny it by storing a non-zero number in D0.

That's a brief overview. You should look at the beta Inside Macintosh chapter to get the complete story.