Tech Info Library

Don't Rely On Network Events To Signify Completed Operations

Revised: 3/31/88
Security: Everyone

Don't Rely On "Network Events" To Signify Completed Operations

==

This article last reviewed: 22 March 1988

You should not rely on "network events" to signify when an operation is completed. Network events may be purged from the event queue if it overflows, thus preventing the application from ever receiving them.

In general, you should avoid using anything other than disk-inserted or device driver events: although disk-inserted and device driver events don't have any higher priority than network events, they are much less likely to get removed from the queue because of a queue overflow. (Network events have this purging problem because multiple ATP requests may complete simultaneously, generating lots of events.)

NOTE: Another major reason not to use network events is because Apple can't guarantee to support them in the future.

It is not necessary to find alternatives to disk or driver events.

Most applications do not need to respond to disk-inserted events. By the time a disk-inserted event reaches the application, the system has already attempted to mount the volume. Most applications use Standard File, which does the response to disk-inserted events for you.

NOTE: If your application DOES need to react to disk-inserted events, it should receive this event in the regular fashion, as described in the "Event Manager" chapter of "Inside Macintosh" (page I-241, Apple Part #030-1277-A).

Also, applications usually don't have to respond to device driver events, either; in most cases, you will use higher level managers. For example, most applications depend on the File Manager, rather than the Disk Driver. Another example: the Sound Driver allows you to make a call to SoundDone to find out if an asynchronous sound is finished.

In the case of device drivers, when an asynchronous call is made you can poll the ioResult field of the parameter block to see if the call has completed. (This field is set to 1 when the call is made.) Control is returned to the application immediately, even though the call hasn't necessarily completed. The ioResult field receives the actual result code on completion of the call, so you can determine if the call has completed by periodically checking to see if ioResult is still 1.

NOTE: See the Device Manager chapter of the above-mentioned "Inside Macintosh" (page II-173) for more details.

Tech Info Library Article Number:2755