About The Notification Manager and System 7.0

This section describes how you can use the **Notification Manager** to inform users of significant occurrences in applications that are running in the background or in software that is largely invisible to the user. This software includes device drivers, vertical blanking (VBL) tasks, Time Manager tasks, completion routines, and desk accessories that operate behind the scenes. It also includes code that executes during the system startup sequence, such as code contained in 'INIT' resources.

The **Notification Manager** is available in system software versions 6.0 and later. You can use the Gestalt function to determine whether the **Notification Manager** is present. See the **Compatibility Guidelines** for complete details on using Gestalt.

The information in this section supersedes the information that was previously published in Appendix D of the *Programmer's Guide to MultiFinder*. You need to read this section if your application, desk accessory, or device driver might need to notify the user of some occurrence while it is running in the background or is otherwise invisible to the user. You also need to read this section if you want to write 'INIT' resources that might need to inform the user of important occurrences during their execution at system startup time.

The **Notification Manager** provides an asynchronous notification service. It allows software running in the background (or otherwise unseen by the user) to communicate information to the user. For example, applications that manage lengthy background tasks (such as printing many documents or transferring large amounts of data to other machines) might need to inform the user that the operation is complete. These applications cannot use the standard methods of communicating with the user, such as alert or dialog boxes, because such windows might easily be obscured by the windows of other applications. Moreover, even if those windows are visible, the background application cannot be certain that the user is aware of the change. So some more reliable method must be used to manage the communication between a background application and the user, who might be awaiting the completion of the background task while running other applications in the foreground.

In the same way, relatively invisible operations such as Time Manager tasks, VBL tasks, or device drivers might need to inform the user that some previously started routine is complete or perhaps that some error has rendered further execution undesirable or impossible.

In all these cases, the communication generally needs to occur in one direction only, from the background application (or task, or driver) to the user. The **Notification Manager**, included in system software versions 6.0 and later, allows you to post to the user a notification, which is an audible or visible indication that your application (or other piece of software) requires the user's attention. You post a notification by issuing a notification request to the **Notification Manager**, which places your request into a queue. When your request reaches the top of the queue, the **Notification Manager** posts a notification to the user.

You can request three types of notification:

- **Polite notification.** A small icon appears to flash by periodically alternating with the Apple menu icon (which is the Apple symbol) or

the Application menu icon in the menu bar.

- **Audible notification.** The **Sound Manager** plays the system alert sound or a sound contained in an 'snd ' resource.

- **Alert notification.** An alert box containing a short message appears on the screen. The user must dismiss the alert box (by clicking the **OK** button) before foreground processing can continue.

These types of notification are not mutually exclusive; for example, an application can request both audible and alert notifications. Moreover, if the requesting software is listed in the Application menu (and hence represents a process that is loaded into memory), you can instruct the **Notification Manager** to place a diamond-shaped mark next to the name of the requesting process. The mark is usually intended to prompt the user to switch the marked application into the foreground. Finally, you can request that the **Notification Manager** execute a notification response procedure, which is a procedure that is executed as the final step in a notification.

In short, a notification consists of one or more of five possible actions. If you request more than one action, they occur in the following order:

1. A diamond-shaped mark appears next to the name of your application in the Application menu, as illustrated in the Figure below. Note that the diamond is present only when your application is in the background (because the diamond is replaced by a checkmark if your application is the active application). In the Figure below, the **Other App** application is the active application.

2. A small icon alternates with the Apple menu icon or the Application menu icon in the menu bar. Typically, the small icon is your application's small icon. Note that several applications might post notifications, so there might be a series of small icons alternating in the menu bar. The location of each flashing icon follows that of the posting application's mark (if any). If your application is marked with a diamond (or a checkmark) in the Application menu, the icon flashes above the Application menu; otherwise, the icon flashes above the Apple menu.

3. The **Sound Manager** plays a sound. Your application can supply its own sound (by passing the **Notification Manager** a handle to an 'snd ' resource loaded into memory) or request that the **Sound Manager** use the user's system alert sound.

A notification in the Application menu

4. An alert box appears, and the user dismisses it. Your application specifies the text in the alert box.

5. A response procedure executes. You can use the response procedure to remove the notification request from the queue or perform other processing.

The mark in the Application menu and the alternating small icon remain until the requesting application removes the notification request from the queue. However, the sound and the alert box are presented only once, if at all.

Any applications, desk accessories, tasks, routines, or drivers can use the **Notification Manager**, whether they are running in the background or not. It is especially useful for background applications, such as the PrintMonitor application. (The system alarm, which is called by the Alarm Clock desk accessory, also uses the **Notification Manager**.) Foreground applications can, however, use the services of the **Notification Manager** to achieve effects (such as the alternating small icon) that are otherwise more difficult to create. For the same reasons, the **Notification Manager** can be useful even to applications that might be executing in a Finder-only environment under system software version 6.0.

The **Notification Manager** provides applications with a standard user interface for notifying the user of significant events. It is suggested that your application adopt the following three-level notification strategy for communicating with the user:

1. Display a diamond next to the name of the application in the Application menu.

2. Insert a small icon into the list of icons that alternate with the Apple menu icon or the Application menu icon in the menu bar, and display a diamond next to the name of your application in the Application menu.

3. Display a diamond, insert a small icon, and put up an alert box to notify the user that something needs to be done.

Ideally, the user should be allowed to set the desired level of notification. The suggested default level of notification is level 2. In levels 2 and 3, you might also play a sound, but the user should have the ability to turn the sound off. In addition, a user should have the ability to turn off background notification altogether, except in cases where damage might occur or data would be lost.

> **Note:** This suggested notification strategy may not be appropriate for your application. Notifications posted by system software do not follow these guidelines.

Each application, desk accessory, and device driver can issue any number of notification requests. Each requested notification is presented separately to the user. For this reason, you should try to avoid posting multiple notification requests for the same occurrence. Depending on the method of notification you specify, multiple requests might result in an annoying number of notification sounds or a large number of alert boxes that the user must dismiss before continuing.

Note that the **Notification Manager** provides a one-way communications path from an application to the user. There is no provision for carrying

information back from the user to the requesting application, although it is possible for the requesting application to determine if the notification was received. If you require this secondary communications link, do not use the **Notification Manager**. Instead, you should wait until the user switches your application into the foreground and then use standard means (for example, a dialog box) to obtain the required information.