The System 7.0 Environment

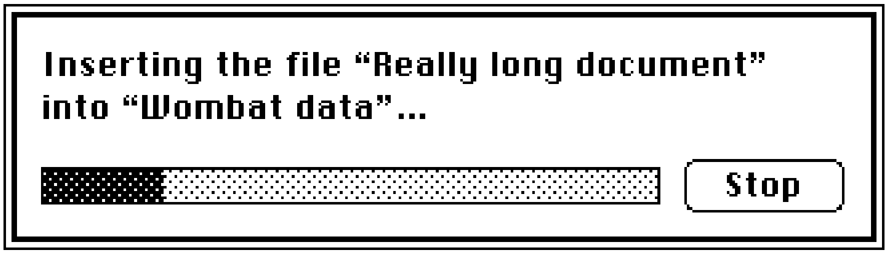
This section briefly describes the general user interface recommendations that affect your application when it runs in the system 7.0 environment. The changes to system software and the operating environment are described in the Introduction to the System Software Version 7.0 Environment in *Inside Macintosh, Volume VI,* and in the
Compatibility Guidelines .

In previous versions of system software, a cooperative multitasking environment was available to users with MultiFinder. Users could turn on MultiFinder so that they could open multiple applications at one time; however, most people did not use MultiFinder regularly. In system 7.0 the cooperative multitasking environment is standard. Now all users can open as many applications and desk accessories as their computer's memory can support. The Macintosh computer manages applications in much the same way that each application handles its own windows.

As in previous versions of system software, only one application can be active at a time. The frontmost application, the one interacting with the user, is the active application. Its small icon represents the **Application** menu in the menu bar and appears next to the application's name in the **Application** menu. Your application should update the controls in the frontmost window whenever the user switches to your application. If you previously did not update your application to be compatible with MultiFinder, you now need to modify your application's event loop to accommodate a cooperative environment. For more information on the operating environment in system 7.0 , see the **Compatibility Guidelines**.

User Feedback

When your application is the active application, you need to provide feedback to the user to indicate what's happening. A user learns to predict how long certain operations last. In the system 7.0 environment, multitasking, virtual memory, and network connectivity cause task length to become more variable. A user will not always be able to predict the length of time per task. Therefore it becomes more essential to display feedback about what is taking place. If you do not, the user may think that the Operating System stopped running and may attempt to correct a perceived error condition, perhaps by manually restarting the machine. At least use the spinning beach ball or animated watch cursor to indicate an operation in process. If you can approximately determine the amount of time a task will last, it's even better to use a progress indicator so that a user knows that the Operating System is still running and that an operation is occurring. The Figure below shows an example of a progress indicator.

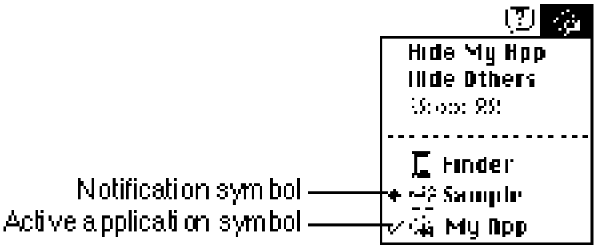


Inserting the file “Really long document” into “Wombat data”…

A progress indicator

Background Notification

When your application runs in the background, you may need to get the user's attention to respond to a task completion or a request for input. The **Notification Manager** provides several ways for your application to alert the user. When a background task is running and you need to notify the user, use the **Notification Manager** to alternate an icon in the menu bar with the icon for the **Application** menu or Apple menu as appropriate. In general, you should display an icon that corresponds to your application or system extension, so that the user gets a visual clue about which application is requesting attention. In addition, you should display a diamond-shaped mark next to your application's name in the **Application** menu. You can also play a sound. The Figure below shows an example of a notification symbol.



The **Application** menu with a notification symbol

Nothing more should happen until the user chooses to activate your application, at which time you can display a modal dialog box. Your dialog box or message must inform the user about what needs attention, why attention is needed, and what to do. For example, a dialog box might say "Transmission of the file My Phone List to 415-555-1212 could not be completed because the phone line went dead," and it might present the user with two buttons, **Try Again** and **Cancel**.

A background application should not take control from the user by placing an alert box on the screen when the user has not activated the application. If an immediate response is crucial and the user does not respond to the notification request, your application needs to handle the situation gracefully.

See the **Notification Manager** for information about implementing these techniques.