

## Dialog Boxes

This section presents revised guidelines for design and layout of effective dialog boxes. The guidelines rely on the principles of feedback and dialog, forgiveness, and consistency as described in *Human Interface Guidelines: The Apple Desktop Interface*. These guidelines supersede previous guidelines about dialog boxes.

### Modal Dialog Box Behaviors

In system 7.0 the **Dialog Manager** has been updated to provide additional support for feedback mechanisms and menu bar access. When you display a modal dialog box, the **Dialog Manager** disables the **Application** menu, the About Balloon Help command in the **Help** menu, and the About Keyboards command in the Keyboard menu. It then checks to see if you are handling menus during a modal dialog box. These conditions are explained in detail in the **Compatibility Guidelines**.

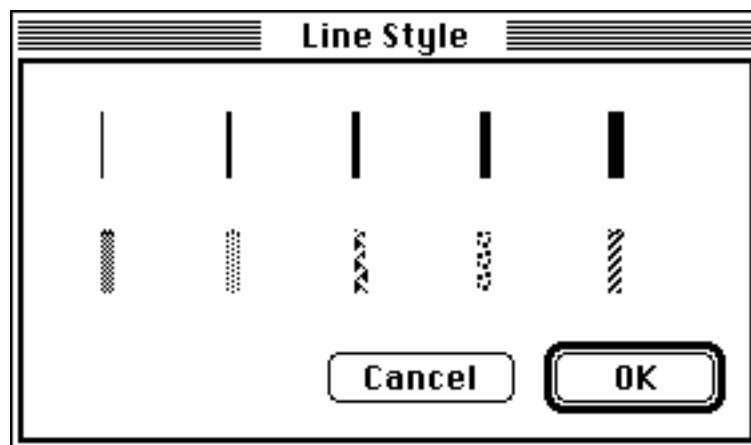
If the **Dialog Manager** determines that you are not handling your own menus, it disables the rest of the menu bar except for the **Help** menu. The **Dialog Manager** then determines whether the dialog box contains an active editable text box and if you have the standard keyboard equivalents for the Cut, Copy, and Paste commands. If both of these conditions are met, then the **Dialog Manager** enables the **Edit** menu and those commands in the **Edit** menu.

If the **Dialog Manager** detects that you are handling menus in your application, it only disables the **Application** menu. You must provide access to the **Help** and **Edit** menus. To support the Cut, Copy, and Paste commands you need to convert the Clipboard before and after you display a modal dialog box. You can also provide menu bar access in your application by enabling menus and commands in those menus that make sense in the context of the current task. See the **Compatibility Guidelines** for information on enabling menus when you display a modal dialog box.

### Movable Modal Dialog Boxes

System 7.0 introduces a new window class, the movable modal dialog box. The user sometimes needs to see document contents that a modal dialog box obscures. To allow the user to move a dialog box in this case, you can use a movable modal dialog box rather than a modal dialog box. The movable modal dialog box has a title bar as part of its standard window so that the user can move the dialog box by dragging the title bar.

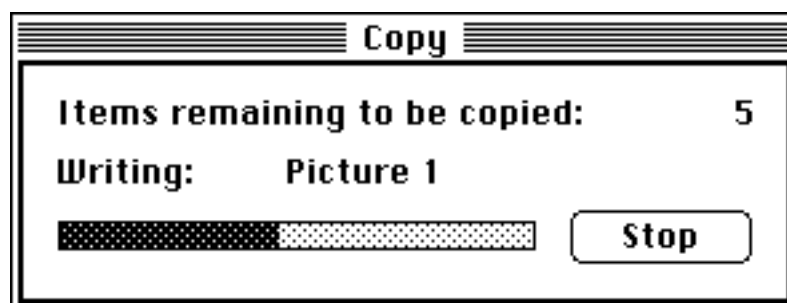
The design of the movable modal dialog box combines the standard modal window with a title bar with racing stripes, but no close box or zoom box. This design gives the user visual feedback that the dialog box is modal, and must be responded to before completing any other action in the active application, but the user can move it. The Figure below shows a movable modal dialog box with attribute options that affect an area a user would want to see, such as the text that a border would surround.



A movable modal dialog box

To create a movable modal dialog box, use the window definition ID of the movable modal dialog box in the standard resource type 'WDEF'. As with all movable windows, be sure to save the position of the movable modal dialog box window for the next time it's used.

Movable modal dialog boxes should respond like modal dialog boxes in most ways. When you display a movable modal dialog box, however, there are some additional behaviors you need to support. You must make certain that the dialog box is modal within your application. That is, the user should not be able to switch to another of your application's windows while the dialog box is active. Allow your application to run in the background when you display a movable modal dialog box. For example, system 7.0 uses movable modal dialog boxes to show that an application is busy with a time-consuming operation, yet a user can still switch the application to the background. The Figure below shows a movable modal dialog box displayed by the Finder when it is copying files.



A Finder movable modal dialog box

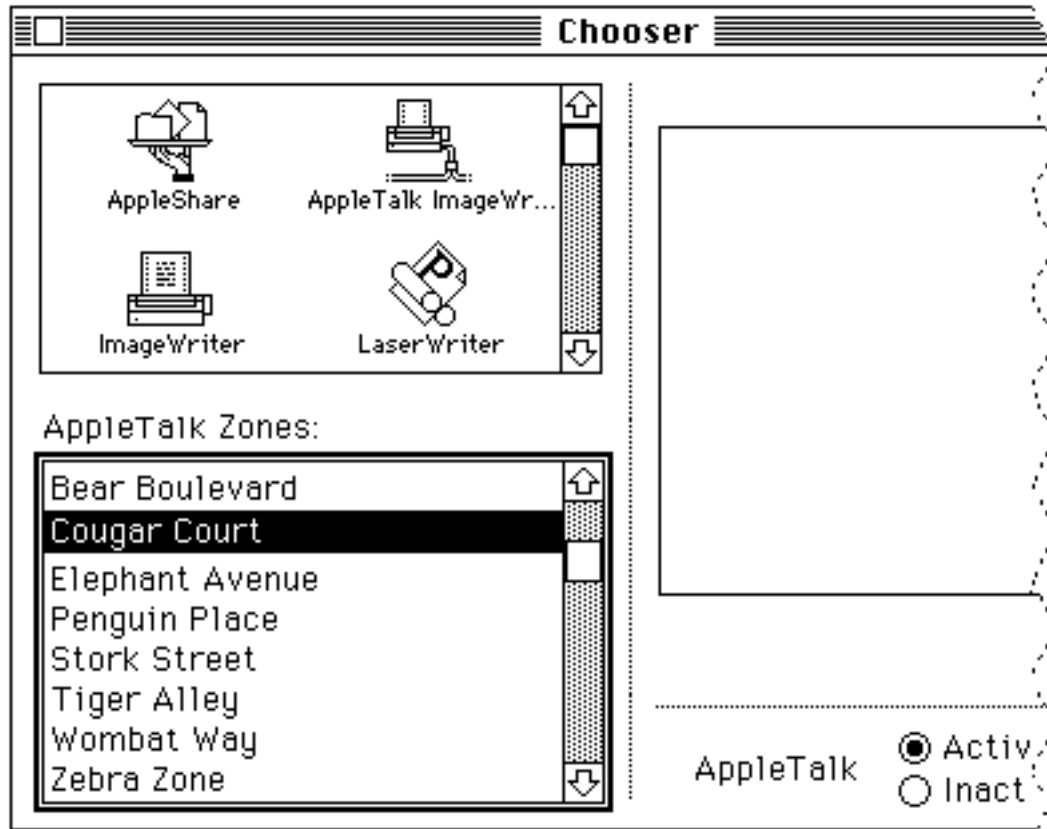
You need to provide access to the menu bar when you display a movable modal dialog box. Provide access to the **Help** menu, the **Edit** menu, the **Keyboard** menu when appropriate, and any context-appropriate commands. Also enable the **Application** menu so the user can switch to another application.

It's important to consider whether you can use a modeless dialog box instead of a movable modal dialog box-to preserve the user's ability to perform any task in any order. See the [\*\*Compatibility Guidelines\*\*](#) for information on implementing movable modal dialog boxes.

### Keyboard Navigation in Dialog Boxes

In previous versions of system software you could select an item in the scrolling list in the standard file dialog box for opening files by using the keyboard. The ability to select an item from a set of items by typing the beginning character or characters of its name is called *type selection*. The user can also use the arrow keys to move the selection by one item in the direction of the arrow. Type selection has been extended to work in other lists, such as the list of files in a Finder window and the list of available devices in the **Chooser**.

Some dialog boxes have several elements, such as text boxes and scrolling lists, that can accept input from the keyboard. It's necessary to visually indicate which element is currently accepting input from the keyboard in order to let users know which of the possible elements is active. Each element has its own distinct indicator. As in the past, a text box displays a blinking insertion point or selected text range to indicate that it is accepting keyboard input. When a scrolling list is the active element in a dialog box, its visual indicator is a rectangular border of two black pixels, which is separated from the list by one pixel of white space. The Figure below shows the AppleTalk Zones list in the **Chooser** as an active scrolling list area.



A selected scrolling list

When a user activates a scrolling list, using the following **QuickDraw** routines outlines the scrolling list in the standard way:

```
PenSize(2,2);  
InsetRect(scrollRect,-3,-3);  
FrameRect(scrollRect);
```

Since all typing goes to the active window, there should be only one active

area and only one indicator at any time. If a dialog box has only one element that can accept keyboard input (and that element is a scrolling list), it's not necessary to outline a scrolling list. In the standard file dialog box the user can use type selection to identify the desired file in the list of files, but, since there's no other list or text box, the selected list does not have a border.

In a dialog box the user can move the active area to any interface element that accepts keyboard input, such as a text box, by clicking the desired element or by pressing the Tab key to cycle through the available elements.

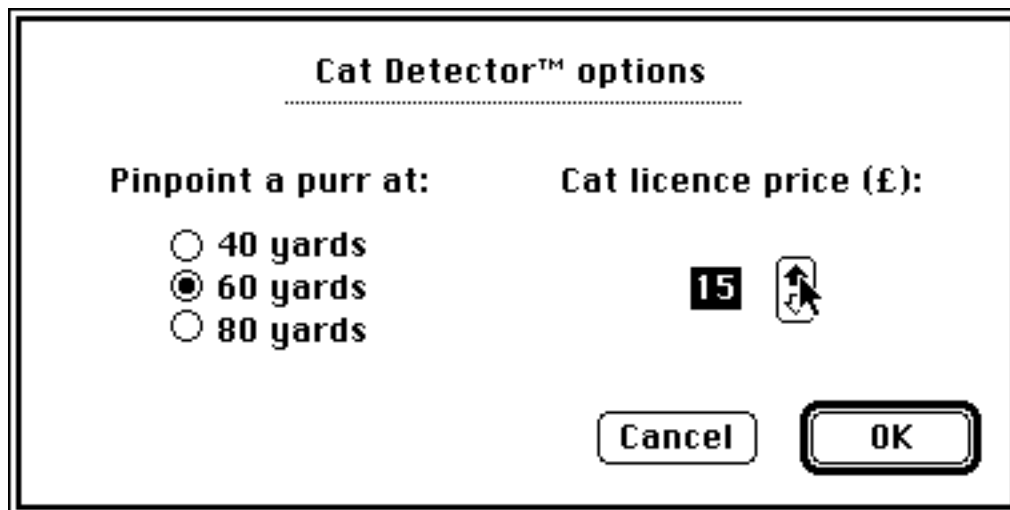
### Button Labels

Whenever possible, label a button with a verb that describes the action that it performs. Use book-title capitalization for button labels. In general, this means that you capitalize one-word titles and, in multiple-word titles, capitalize words of four or more letters. Usually you do not capitalize words like *in*, *an*, or *and*. The specific rules for this type of capitalization appear in detail in the *Apple Publications Style Guide*.

Provide a **Cancel** button whenever you can, and always map Command-period and the Esc (Escape) key to the **Cancel** button. Map the Return key and the Enter key to the default button, which is usually the button with the safest result or the most likely response. Do not display a default border around any button if you use the Return key in editable text boxes. Having two behaviors for one key confuses users and makes the interface less predictable.

In all dialog boxes, any buttons that are activated by key sequences must invert to give visual feedback that indicates which item has been chosen. A good rule of thumb is to invert the button for 8 ticks of the clock, which is long enough to be visible, but short enough that it's not annoying. All alert boxes and modal dialog boxes that use the **ModalDialog** procedure exhibit this behavior. If you implement your own dialog boxes or alert boxes, be sure to include this behavior. See the **Compatibility Guidelines** for more information on the **ModalDialog** procedure.

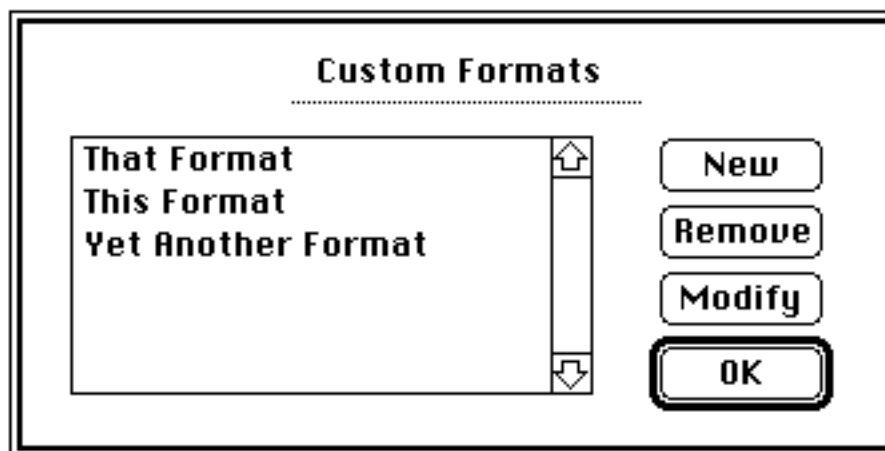
A user typically reads the text in a dialog box until it becomes familiar and then relies on visual cues, such as button names or positions, to respond. Names such as **Save**, **Quit**, or **Erase Disk** allow users to identify and click the correct button quickly. These words are often more clear and precise than words like **OK**, **Yes**, and **No**. If the action can't be condensed into a word or two, **OK** and **Cancel** or **Yes** and **No** may serve the purpose. If you use these generic words, be sure to phrase the wording in the dialog box so that the action the button initiates is clear. The Figure below shows a dialog box with appropriate **OK** and **Cancel** buttons.



A dialog box with **OK** and **Cancel** buttons

Use **Cancel** for the button that closes the alert or dialog box and returns the computer to the state it was in before the alert or dialog box appeared. **Cancel** means "dismiss this operation, with no side effects." It does not mean "I've read this dialog box" or "stop what you're doing regardless."

When it is impossible to return to the state that existed before an operation began, do not use the word **Cancel**. You can use **OK** or **Stop**, which are useful in different situations. Use **OK** for the name of a button that closes the alert or dialog box and accepts any changes made while the dialog box was displayed. The Figure below shows a dialog box that illustrates this guideline.

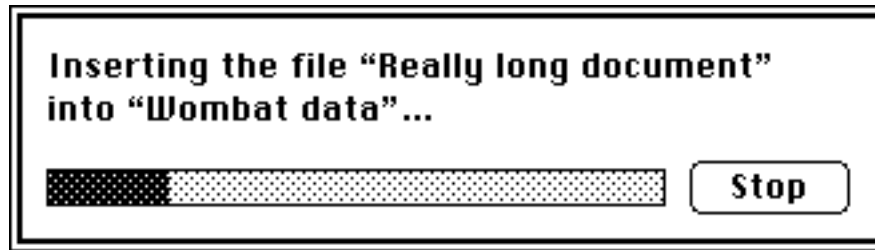


A dialog box with **OK** instead of a **Cancel** button

This dialog box uses **OK** because clicking the button maintains any changes that were made subsequent to the display of the dialog box. If the button were named **Cancel**, clicking it should remove any formats created, removed, or changed since the dialog box appeared, and it should return the computer to the state it was in before the dialog box appeared.

Use **Stop** for a button that halts an operation midstream while accepting the possible side effects. **Stop** may leave the results of a partially complete task intact, whereas **Cancel** always returns the computer to its previous state. It's appropriate to change the button name in the middle of the operation from

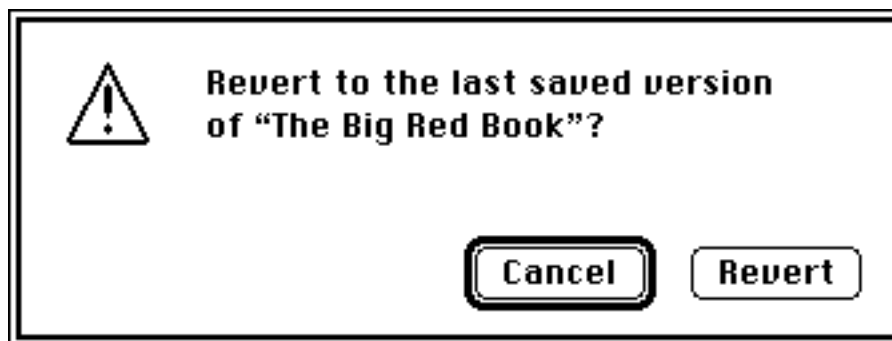
**Cancel** to **Stop** if you can determine when it's no longer possible to cancel. The Figure below shows a dialog box that illustrates this guideline.



A progress indicator that uses a **Stop** button

The dialog box in the Figure above uses **Stop** because clicking the button maintains the text that is already inserted while preventing completion of the insert operation.

In an alert box that requires confirmation, use a word that describes the result of accepting the message in the dialog box. For example, if a dialog box says "Revert to the last saved version of this document," label the button **Revert** rather than **OK**. The Figure below shows a dialog box with appropriately labeled buttons.



A confirmation alert box

If there is a most likely action, use a default button. This button usually completes the action that the user initiated to bring up the dialog box. The default button is outlined with an additional border of three black pixels, separated by a border of one white pixel, and its action is performed when the user clicks the button or presses the Return or Enter key.

Do not use a default button if the most likely action is dangerous—for example, if it causes a loss of user data. When there is no default button, pressing Return or Enter has no effect; the user must explicitly click a button. This guideline protects users from accidentally damaging their work by pressing Return or Enter out of habit. You can consider using a safe default button, such as **Cancel**.

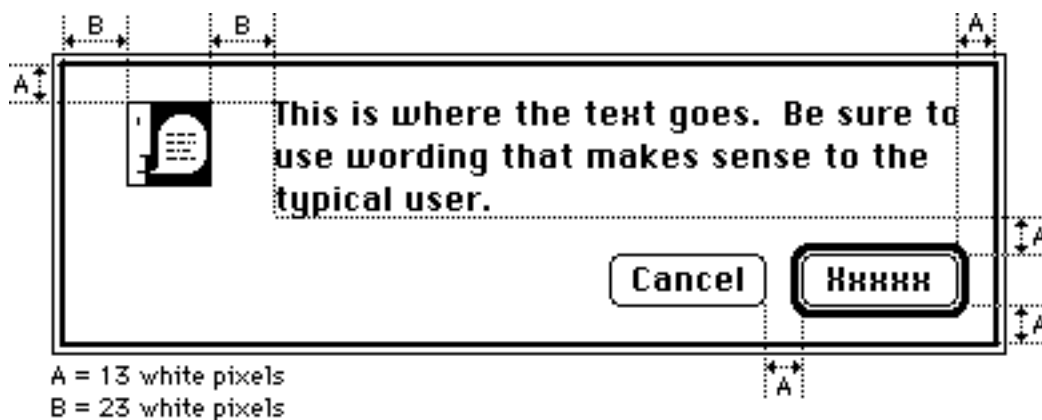
A modal dialog box usually cuts the user off from the task. That is, he or she cannot see the area of the document that changes when choices are made in the dialog box until dismissing the dialog box. Once the area becomes visible by dismissing the dialog box, the user sees whether the changes are the desired ones. If the changes are not appropriate, then the user has to repeat the entire operation. To provide better feedback to the user, you need to provide a way for the user to see what the changes will be. Therefore, any selection made in a

modal dialog box should immediately update the document contents, or you should provide a sample area in the dialog box that reflects the changes that the user's choices will make. In the case of immediate document updating, the **OK** button means "accept this change" and the **Cancel** button means "undo all changes done by this dialog box."

Some applications use an **Apply** button to approximate this behavior. This method confuses the meaning of **OK** and **Cancel** and is not recommended. If you must implement modal dialog boxes with an **Apply** button, you need to include a **Cancel** button and a **Revert** button in the dialog box. Otherwise the **Cancel** button becomes confusing to the user. When there is an **Apply** button, the **Cancel** button undoes the results of the **Apply** operation and dismisses the dialog box. The **OK** button dismisses the dialog box. The **Revert** button returns the document to the state it was in before the dialog box was displayed. The user must always be able to undo any actions caused by the dialog box.

### Dialog Box Layout

In most simple dialog boxes, such as alert boxes, you should place buttons in functional and consistent locations, both within your application and across all applications that you develop. Place the action button in the lower-right corner with the **Cancel** button to its left. The Figure below shows the recommended location for buttons and text. The default button can be any button; its assignment is secondary to the consistent placement of buttons. This rule keeps the action button and the **Cancel** button consistently placed. Otherwise, the buttons would keep changing location depending on the default choice for the dialog box.



The recommended spacing of buttons and text in a dialog box

Use a consistent amount of white space between the border of the dialog box and its elements. This creates a balanced appearance in the dialog box. Otherwise the user might perceive a lopsidedness or other visual imbalance in your dialog box.

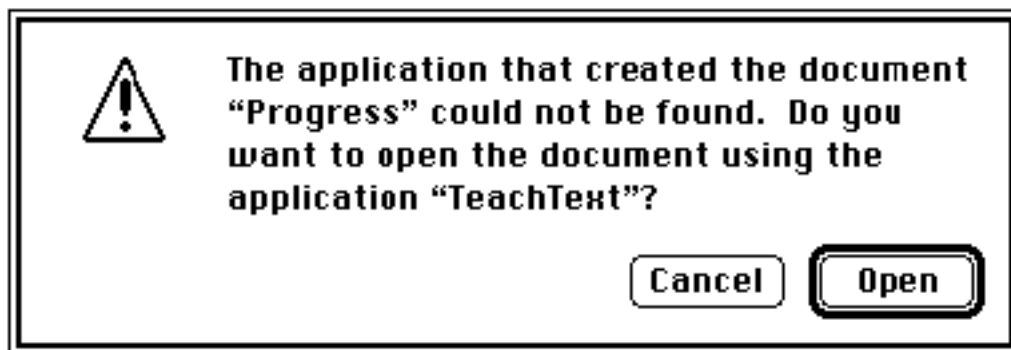
The Western reader's eye tends to move from the upper-left area of the dialog box to the lower-right area. Put the initial impression that you want to convey in the upper-left area (like the alert icon that appears in alert boxes), and place the buttons that a user clicks in the lower-right area. Following this guideline makes it easier for users to identify what's important in a dialog box.

When dialog boxes are localized for worldwide versions of system software, the text in the dialog box may become longer or shorter. The alignment of the

items in the dialog box may vary with localization. Arabic and Hebrew are written right to left, so alignment of the items in an Arabic or Hebrew dialog box should be right to left. The **Control Manager**, **Menu Manager**, and **TextEdit** routines handle the alignment of dialog box components. For more information, see the sections that describe those managers. Be sure to create dialog items of the same size, so that they align properly when a user has a script that reads from right to left. This guideline is discussed earlier in **Worldwide Software Development**.

### Dialog Box Messages

Write messages in dialog boxes and alert boxes that make sense to the user. Use simple, nontechnical language and do not provide system-oriented information that the user cannot respond to. When possible, give the user information that helps explain how to correct the problem. The Figure below shows an example of a well-written dialog box message that replaces the message users used to see, "The application is busy or missing."



A well-written dialog box message

Use the name of the document or application in a dialog box when the text refers to it. For example, a dialog box that appears when a user chooses Shut Down after working on the company's annual report using the TeachText application should say "Save changes to the TeachText document "Annual Report" before quitting?" rather than simply "Save changes before quitting?" This kind of labeling helps users who are working with several documents or applications at once to make decisions about each one individually.

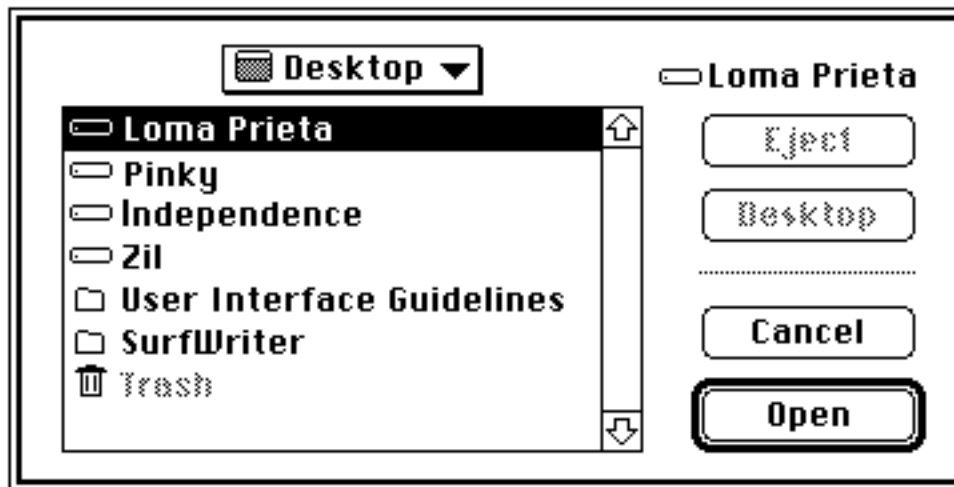
### Standard File Dialog Boxes

The system 7.0 standard file dialog boxes present some new information to the user. They show a file's position in relation to the disk it's stored on. Instead of showing the root level of a hard disk as the highest level of the directory structure, the desktop now appears as the top level of the Hierarchical File System. The **Drive** button has been replaced with the **Desktop** button. A user can view and select disk drives from the standard file dialog box and can see other desktop entities such as the Trash folder. The dialog box that appears when the user chooses **Save As** includes a **New Folder** button that allows the user to create a folder in which to store the document. The pop-up menu in this dialog box now includes the downward-pointing triangle for additional visual feedback.

If you interact with the file system directly and use a dialog box similar to the standard file dialog boxes, you should replicate the organization and appearance of the standard file dialog boxes. The Figure below shows an example of the new



standard file dialog box for opening files. For more information, see the [Standard File Package](#).



The new standard file dialog box for opening files

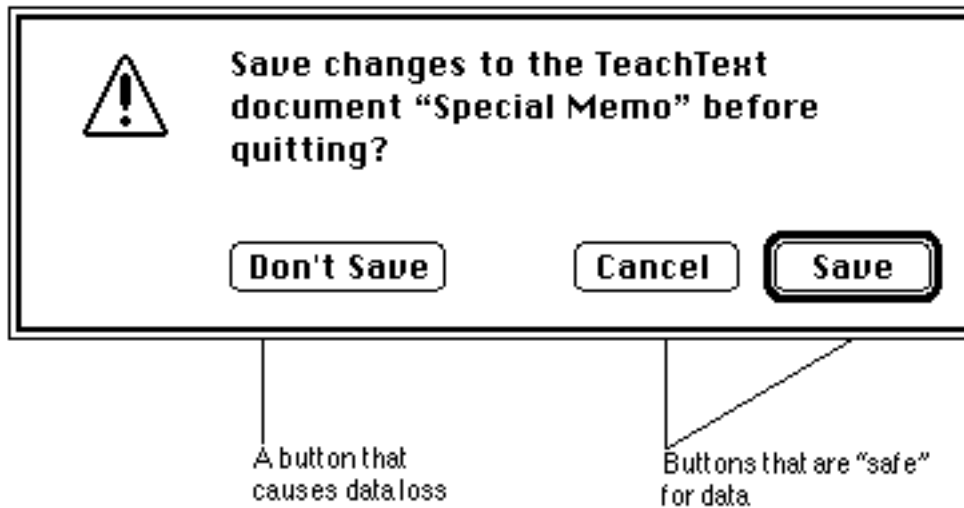
### Save Changes Dialog Box

This section describes the new standard dialog box for saving all changes to a document before a user quits an application. The design presented in Volume IV of *Inside Macintosh* created some situations in which users, especially inexperienced users, could experience a loss of data. The new design addresses those concerns and standardizes the appearance of the dialog box so that users can quickly identify potentially dangerous actions.

Place the standard warning icon in the upper-left corner of the dialog box. This icon indicates to users that they need to carefully consider the dialog box message before clicking the default button or the Return key. The warning icon should always be in the same, predictable location so that users easily recognize it as a warning and respect its meaning.

Previously the buttons in the save changes dialog box were labeled Yes, No, and **Cancel**. The save changes dialog box changes the names of the buttons to correlate to the action users perform by pressing the button. The buttons should now read **Save**, **Do not Save**, and **Cancel**. Using these verbs reinforces the identity of each possible action to the user so that the experience is more intuitive. In other words, the **Do not Save** label provides much more context for the user than the word No does.

The new design provides a safeguard for the user by standardizing the location of buttons in a safe configuration. In order to prevent accidental clicks of the wrong button, you should always keep safe buttons apart from buttons that could cause data loss. Place the **Save** button in the lower-right corner with the **Cancel** button to its left. Place the **Do not Save** button on the left and left-aligned with the message text. This way, the user must explicitly move the pointer and click the button that could cause irretrievable loss of data. The Figure below shows an example of a standard save changes dialog box.



The save changes dialog box

Include the name of your application and the name of the document in the dialog box message, as shown in the Figure above. When a user shuts down the computer, several save changes dialog boxes may appear if there are several open documents on the desktop. This addition of information to the standard message helps the user by identifying to which application and document the message refers.