

Whenever a window is displayed on the screen, the application must make a decision about its size and location. If a user moved the window in an earlier session, then it should be restored to its previous position; otherwise, the application must choose an appropriate default position. This document gives details about making these decisions.

Saving and restoring window positions

Users change the locations and sizes of windows for a reason. They might want to view two documents side by side, or they might want to display a window on a larger monitor so that more of it can be seen at once. They might want to display only the interesting area of the window, which may be quite small, or they might want to position the window in such a way that certain icons on the desktop are still visible. In any case, one of the most important principles of a good interface is that the user is in control, so applications must respect the user's reasoning by reopening each window in the same location and with the same size that the user left it.

Here is a simple, but effective, procedure for saving and restoring window positions:

1. When opening a new window, put it in the default position (see the next section of this document for details about determining the default position).
2. Before closing a movable window, check to see if its location or size have changed. If so, save the new location and size. If the window can be zoomed, save the user state and also save whether or not the window is in the zoomed (standard) state. Note that if the window corresponds to a Finder document and there were no other changes to the document, the new location and size should be saved without changing the modification date of the document.

Note: If the location and size of a movable window have not changed, do not save them, because the default location and size may be different the next time the window is opened (e.g., if the window is reopened on a different Macintosh with a different screen size).

3. When reopening a movable window, check its saved position. If the window is in a position to which the user could have dragged it, then leave it there. If the window can be zoomed and was in the zoomed state when it was last closed, put it in the zoomed state again. (Note that the current and previous zoomed states are not necessarily the same, since the window may be reopened on a different monitor.) If the window is not in a position to which the user could have dragged it, then it must be relocated, so use the default location. However, do not automatically use the default size when using the default location; if the entire window would be visible using the default location and the stored size, then use the stored size.

Remember that checking to see if the saved position is reasonable before reopening the window is a necessary part of this procedure, not an option. When an application opens windows outside of the visible space, users tend to switch to competitors' products.

Choosing a default window position

The appropriate default position of a window may depend upon several factors, including whether the window is a document window or an alert, the locations of other open windows, the user's center of attention, and whether or not the window contains information that is closely related to other open windows. The rest of this document includes a series of default window position examples for several common cases. Developers should consider how their particular situations relate to these common ones to determine the best default positioning. In any case, the default position of any window must never overlap multiple screens, as this can look and feel very strange with monitors of different depth and resolution.

Independent document windows, single screen

On a single screen the first document window should be positioned in the upper-left region of the gray area of the screen. Each additional window should be staggered slightly below and to the right of its predecessor, if no windows are moved or closed. When a window is moved or closed, its original position becomes available again. The next window opened should use this position. Similarly, if a window is moved onto a previously-available position, that position becomes unavailable again.

Figure 1 illustrates independent document windows on a single screen.

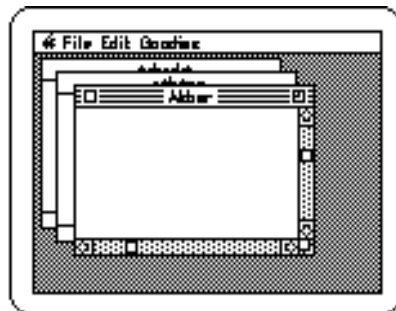


Figure 1—Independent document window positions

Independent document windows, multiple screens

The first document window should be positioned in the upper-left region of the gray area of the main screen (the screen with the menu bar). Each additional independent window should be staggered from the upper-left of the screen that contains the largest portion of the frontmost window. Thus, if the user starts the application, creates a single window, drags that window over to a secondary monitor, and then creates a second window, the second window and subsequent windows should appear on the secondary monitor. Figure 2 illustrates independent document windows on multiple screens.

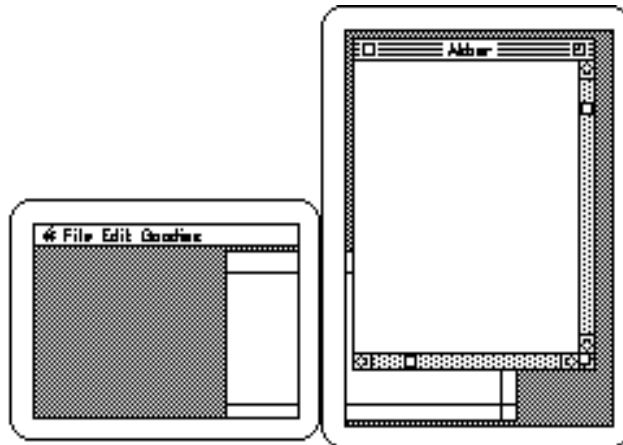


Figure 2—Independent document window positions on multiple screens

Child windows

A child window is a window that contains more detail about part of another window. For instance, in ResEdit a window showing all string resources of a given file is a child of the window showing all resource types for that file. Child windows of visible parent windows should be created just below and to the right of the parent window. Figure 3 illustrates child windows.

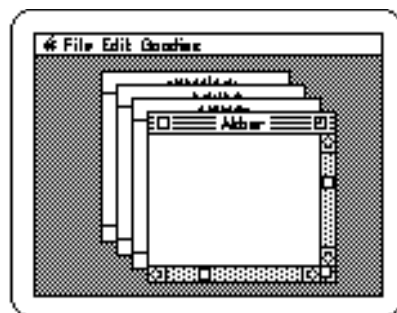


Figure 3—Child Windows

Alert or dialog box, single screen

Alerts or dialog boxes should be centered horizontally and positioned vertically such that one-third of the remaining vertical gray screen space is above the window and the other two-thirds are below.

Figure 4 shows a typical alert.



Figure 4—Alert

Alert or dialog box, multiple screens

This case is similar to the previous one, except that the alert or dialog box should be drawn on the screen closest to the user's center of attention. Always putting an alert or dialog box on the screen containing the cursor is a good rule of thumb. An even better rule is to use the screen on which the last user action took place. For instance, if the user is typing into a word processing document and presses Command-O, put the standard file dialog box on the same screen as the word processing document. When an alert or dialog box appears in response to the user selecting a menu item with the mouse, put the alert or dialog box on the screen with the menu bar. Figure 5 shows an alert in a two-screen configuration.

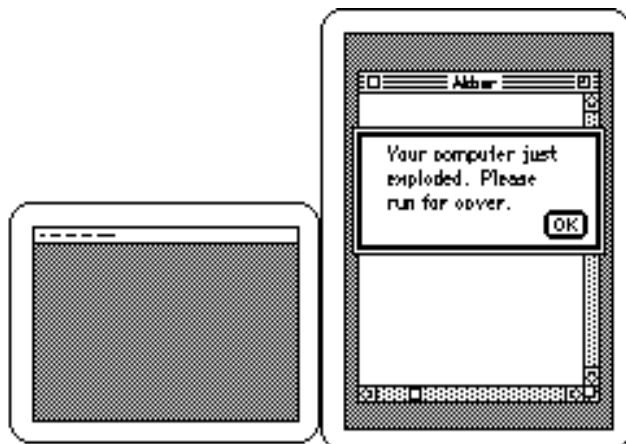


Figure 5—Alert on multiple screens