Changing the Color Environment

The colors available in any particular hardware environment depend on the color or black-and-white settings and the pixel depths that the attached cards and screens can support, and on what they are set to by the system or by the user.

Since your palette may define more colors than the available hardware can display, the **Palette Manager** allocates your requests according to their priority. To make the best use of the **Palette Manager**, you should understand the allocation process. Prioritization is important only when the **ActivatePalette** procedure is called, which occurs automatically when your window becomes the frontmost window. (You may also call **ActivatePalette** yourself after changing one or more of the palette's colors or usage categories.)

The **Palette Manager** first allocates animated colors that are also marked for explicit usage. Colors that are marked as tolerant and explicit are allocated next.

The **Palette Manager** allocates animated colors next. Starting with the first entry in your window's palette (entry 0), the **Palette Manager** checks to see if it is an animated entry. It checks each animated entry to see that the entry has a reserved index for each appropriate device, selecting and reserving an index if it does not. This process continues until all animated colors have been satisfied or until the available indices are exhausted.

The **Palette Manager** handles tolerant colors next. It assigns each tolerant color an index until all tolerant colors have been satisfied. The **Palette Manager** then calculates for each entry the difference between the desired color and the color associated with the selected index. If the difference exceeds the tolerance you have specified, the **Palette Manager** marks the selected device entry to be changed to the desired color.

Since explicit colors designate index values, not the colors at those index locations, and since courteous colors are amenable to being assigned any RGB value, neither is considered during prioritization.

When the **Palette Manager** has matched as many animated and tolerant entries as possible, it checks to see if the current CLUT is adequate. If modifications are needed, the **Palette Manager** overrides any calls made to the **Color Manager** outside the **Palette Manager** and then calls the **Color Manager** to change the device's color environment accordingly (with the **SetEntries** procedure).

Finally, if the color environment on a given device has changed, the **Palette Manager** checks to see if this change has affected any other window in the system. If another window is affected, the **Palette Manager** checks that window to see if it specifies an update in the case of such changes. Applications can use the **SetPalette**, **NSetPalette**, or **SetPaletteUpdates** procedure to specify whether a window should be updated when its environment has been changed because of actions by another window. (If so, the **InvalRect** procedure, described in the **Window Manager**, updates the window, using the boundary rectangle of the device that has been changed.)