

---

**Linking a Color Table to a Palette**

Suppose your application displays an image drawn with 64 gray levels in a window and uses a control to adjust the brightness or contrast. If you draw your image with animated palette entries, you can then use **AnimatePalette** to change the particular shades rapidly. But you have to call **PmForeColor** with the index of the upper-left pixel, paint a 1-by-1 rectangle, call **PmForeColor** for the next pixel, paint a rectangle, and so on.

**Color QuickDraw** now supports a method by which a color table can refer to palette indices instead of RGB values. Setting bit 14 (using OR with 0x4000) in the ctFlags field of the color table record causes the fields in a source pixel map's color table to be interpreted as follows:

**Field descriptions**

<u>ctSeed</u>	A unique value.
<u>ctFlags</u>	Flags, with bit 14 set, as described in the previous paragraph.
<u>ctSize</u>	The size of the color table, less 1.
<u>ctTable</u>	An array [0.. <u>ctSize</u> ] of <u>colorSpec</u> values (each an index value and an RGB value).
<u>ctTable[x].value</u>	The palette entry to use when drawing pixels with value x.(Note that this is different from previous uses, where the value field specified which pixel number the entry applied to.)
<u>ctTable[x].rgb</u>	The color used if there is no available palette or if the palette is too small to contain the specified entry.

For example, to draw a 64-level image in animated colors (as described in **Animated Colors**), you put the palette entry number you want for pixel value 0 into the first value field of the source color table's color specification record, the palette entry corresponding to pixel value 1 in the next value field, and so on. If your palette is arranged in the same order as the pixel values, the value fields simply count from 0 to 63. By calling the **AnimatePalette** procedure for each of those entries, you can produce color table animation on all screens that have reserved an animated entry, are crossed by the window, and are able to display such an image.

Since 16-bit and 32-bit direct devices do not have CLUTs, you may wish to post an update to display an image of those areas that intersect a direct device.

In the case of an animated picture adjustment, like a contrast or brightness control, it is better to animate while the user is moving the indicator and update when the user releases it.