



Tech Info Library

MPW: PixMap Color Issue

This article last reviewed: 9 June 1989

A programmer has a PixMap with a Raster image that has 8-bit pixel values ranging from 0 to 127 and is using CopyBits with the SrcCopy mode. The foreground and background colors are correctly set. The color window has 128 animating colors reserved.

The question is what color table should be assigned to the PixMap? Does he need to use the color table generated by the PaletetoCTab call with the source palette being the palette of the color window?

A NIL handle results in a black and white image. (Note: With tolerant colors used instead of animating colors, the picture appears OK, but you can't animate the colors.)

You can create a similar problem without involving a PixMap. When you use animating entries in the palette, you always get a black and white picture. When you use tolerant entries, we get color. This is a similar problem.

A solution is to use calls to AnimateEntry as a replacement for SetEntryColor. When you use SetEntryColor with animating color entries, you always get black and white. When you use AnimateEntry to set an entry's color, you get color. Specifically:

```
SetPalette(mywindow, mypalette, false);
mycolor.red := 65535;
mycolor.green := 0;
mycolor.blue := 0;
AnimateEntry(mywindow, 2, mycolor);
ActivatePalette(mywindow);
PMForeColor(2);
PaintRect(myrect);
```

yields a red rectangle. If you replace the AnimateEntry call with a SetEntryColor call, you get a black rectangle. The code segment above assumes "mywindow" is a window pointer to a color window. "mycolor" is a variable of type rgbcolor. "mypalette" is a variable of type palettehandle. "myrect" is a variable of type rect.

Copyright 1989 Apple Computer, Inc.

Keywords: <None>

=====

This information is from the Apple Technical Information Library.

19960215 11:05:19.00

Tech Info Library Article Number: 4076