New Technical Notes
Macintosh

Developer Support

Color Cursor Cursing : A Leading Cause
Imaging

Revised by: Alan Mimms October 1989
Written by: Alan Mimms June 1989

Working with color cursors you create from scratch can cause headaches. This Technical Note may help a bit.
Changes since June 1989: Added a warning about purgeable `'clut'` resources.

If you're building an application that creates color cursors, you may encounter some quirks present in Color QuickDraw that manifest themselves in hard-to-understand ways.

If your cursor is, say, 15 pixels tall and 9 pixels wide, you might be tempted to use these values for the `bounds.bottom` and `bounds.right`, respectively, in your cursor's pixel map. **Don't**. The problem is that when the cursor's image needs to be expanded (i.e., when you specify a two bit-per-pixel cursor and the mouse pointer is on an eight-bit screen) the `_SetCCursor` trap rounds the width of the pixel map in such a way that you'll get only the space required for a 15 by 8 pixel map allocated for the expanded cursor data. When the cursor's image is expanded into this too-small expanded cursor data handle as a 15 by 9 pixel map, something in your heap will get munched.

The cure is simple. Make **certain** that you always specify that the `pixmapHandle^^.bounds` be 16 by 16. This will cause `_SetCCursor` to properly allocate the expanded data area, and all will be well in the land. Since the amount of data **drawn** for a cursor is specified by the cursor's pixel values and `'clut'` resource, trying to save a few bytes by making the bounds rectangle smaller than 16 by 16 wouldn't have been very helpful anyway.

Another potential problem is with the color cursor's color table. If you load the color table from a `'clut'` resource using `_GetCTable`, you should make sure that the `'clut'` is marked non-purgeable while the color cursor is in use. If you do not take this precaution, bombs will occur if your `'clut'` gets purged at in inopportune time.