

The evolution of the ICO file format, part 2: Now in color!

 devblogs.microsoft.com/oldnewthing/20101019-00

October 19, 2010



Raymond Chen

Last time, we looked at [the format of classic monochrome icons](#). But if you want to include color images, too? (Note that it is legal—and for a time it was common—for a single ICO file to offer both monochrome and color icons. After all, a single ICO file can offer both 16-color and high-color images; why not also 2-color images?)

The representation of color images in an ICO file is almost the same as the representation of monochrome images: All that changes is that the *image* bitmap is now in color. (The *mask* remains monochrome.)

In other words, the image format consists of a `BITMAPINFOHEADER` where the `bmWidth` is the width of the image and `bmHeight` is *double* the height of the image, followed by the bitmap color table, followed by the *image* pixels, followed by the *mask* pixels.

Note that the result of this is a bizarre non-standard bitmap. The height is doubled because we have both an image and a mask, but the color format changes halfway through!

Other restrictions: Supported color formats are 4bpp, 8bpp, 16bpp, and `oRGB` 32bpp. Note that 24bpp is not supported; you'll have to convert it to a `oRGB` 32bpp bitmap. Supported values for `biCompression` for color images are `BI_RGB` and (if your bitmap is 16bpp or 32bpp) `BI_BITFIELDS`.

The mechanics of drawing the icon are the same as for a monochrome image: First, the mask is ANDed with the screen, then the image is XORed. In other words,

$$\text{pixel} = (\text{screen AND mask}) \text{ XOR image}$$

On the other hand, XORing color pixels is not really a meaningful operation. It's not like people say "Naturally, fuchsia XOR aqua equals yellow. Any idiot knows that." Or "Naturally, blue XOR eggshell equals apricot on 8bpp displays (because eggshell is palette index 56, blue is palette index 1, and palette index 57 is apricot) but is equal to #FoEA29 on 32bpp displays." The only meaningful color to XOR against is black, in which case you have "black XOR Q = Q for all colors Q". (XORing against white is not generally useful.)

mask	image	result	operation
0	Q	$(\text{screen AND } 0) \text{ XOR } Q = Q$	copy from icon
1	0	$(\text{screen AND } 1) \text{ XOR } 0 = \text{screen}$	nop
1	Q	$(\text{screen AND } 1) \text{ XOR } Q = \text{screen XOR } Q$	dubious

For pixels you want to be transparent, set your mask to white and your image to black. For pixels you want to come from your icon, set your mask to black and your image to the desired color.

We now have enough information to answer a common question people have about icons. After that break, we'll return to the evolution of the ICO file format.

For further reading: [Icons in Win32](#).

[Raymond Chen](#)

Follow

