

# What's the difference between CopyIcon and DuplicateIcon?

[devblogs.microsoft.com/oldnewthing/20131004-00](http://devblogs.microsoft.com/oldnewthing/20131004-00)

October 4, 2013



Raymond Chen

There are two functions that can be used to create one icon that is identical to another. One of them is `CopyIcon`. The other is `DuplicateIcon`. What's the difference? There isn't any difference. Both functions clone an icon. In fact, their implementations are basically line-for-line identical. Originally, there was just one function to clone an icon: `CopyIcon`. Windows 3.0 introduced Program Manager, and the developers of Program Manager wrote their own function called `DuplicateIcon`. Why? I have no idea. My guess is that they didn't realize that such a function already existed, so they inadvertently reinvented the wheel. Windows NT 3.1 came along, and the team that ported Program Manager to 32-bit Windows also ported the `DuplicateIcon` function, and they figured, "This function is so useful, we'll export it for anybody to use!" Meanwhile, the original `CopyIcon` function is sitting there saying, "What am I, chopped liver?" Anyway, it's a sad story, but that's how we ended up with two functions that do exactly the same thing. Personally, I would recommend using the `CopyIcon` function. It's in `user32.dll`, which you are almost certainly already linked to if you're doing anything with icons in the first place, so the incremental cost is much lower.

**Update:** Joshua points out that the two functions are not identical. `DuplicateIcon` takes an extra instance handle parameter. Now it makes sense. The shell version is an enhancement to the user version in that it can also transfer icon ownership to another module. (Hence the new first parameter.) This was important in 16-bit Windows because icons were resources which were associated with modules. If you wanted to use an icon after the module was unloaded, you needed to copy it and transfer ownership. But this ownership transfer step is not needed in Win32 because, as we saw yesterday, icons are no longer tied to the underlying resources. So the functions started out different but now they're the same.

[Raymond Chen](#)

**Follow**

