

How do I preserve the user's notification icon preferences for my program after I update it?

devblogs.microsoft.com/oldnewthing/20171027-00

October 27, 2017



Raymond Chen

When a program creates a notification icon, the user can set preferences for that icon: Show icon and notifications, Hide icon and notifications, and Only show notifications. When you update the program, what do you have to do to get those preferences to carry over to the new program?

Give your notification icon a GUID.

Specifically, in the `NOTIFYICONDATA` structure, set the `NIF_GUID` flag in the `uFlags` and put a GUID in the `guidItem` member. If the old version of the program and the updated version use the same GUID, then the system will recognize that the new icon should pick up the settings from the old icon.

More generally, the GUID is how the system decides which user preferences apply to which icons. When the user selects preferences for an icon, the system associates those preferences with the GUID, and the next time an icon with the same GUID appears, the system uses those preferences.

But what happens if you don't specify a GUID?

In that case, the system starts applying heuristics. It checks whether the full path to the executable is mostly the same, where "mostly the same" means that some directories are considered equivalent; for example, a file in `Program Files` is considered to match a corresponding file in `Program Files (x86)`. If the executable is mostly the same, then the system applies additional fuzzy logic based on trying to match some combination of the `uID`, the tooltip provided in the `szTip` member, and the pixels in the icon provided in the `hIcon`.¹ If the fuzzy matcher says "Yeah, they're probably the same," then the system will consider the two icons as equivalent and share their settings.

Note that this heuristic is a fallback, so you shouldn't rely on it in your programs. You should just set a GUID on your notification icon so that you get consistent behavior.

Bonus chatter: If you generate notification icons dynamically, you would want to specify a GUID that is deterministically generated from whatever identity you use to generate your notification icon. For example, if you create a separate notification icon for each server that you discover, then you can hash the server name in order to produce a type 3 or type 5 UUID. That way, when you encounter the same server in the future, you will regenerate the same GUID.

¹ Note that this means that programs that animate their icon are less likely to have a successful match because the pixels of the icon that the preferences were saved under may not match the pixels of the icon that is currently being displayed.

Raymond Chen

Follow

