

# I'm sorry, you don't have permission to know where this shortcut file should show up in the Explorer window

 [devblogs.microsoft.com/oldnewthing/20090528-00](http://devblogs.microsoft.com/oldnewthing/20090528-00)

May 28, 2009



Raymond Chen

Commenter Gabe suggested that the shortcut file should contain a bit that indicates whether the target of the shortcut is a folder, so that shortcuts to folders can sort with real folders. “That would allow them to sort properly without doing any additional work beyond that required to see its icon.” (Commenter Josh agreed that “The performance reason doesn't really apply here, since explorer is already looking at the target to get the icon,” rejecting commenter AndyC's argument that performance may have been a concern.)

Well, first of all, shortcuts do remember whether the target is a file or a folder, or at least whether it was a file or folder at the time the shortcut was created:

`IShellLink::GetIDList` + `IShellFolder::GetAttributesOf` tells you this and more about the shortcut target.

But saying that this isn't more work than what's necessary to see the icon misses a few points about file icons: First, you can't sort by icon, so the consequences of getting the wrong icon (or never getting the icon at all) are purely visual. What? Never getting the icon at all? Well, yes, if the file has been archived to tape, Explorer won't bother recalling the file just to get its icon and will instead just use its generic type icon, or if there is no generic type icon, a generic document icon. After all, you don't want viewing a folder in Explorer to recall all the files from tape. That sort of defeats the purpose of archiving the files to tape.

Even if Explorer decides to get the icon, it performs that operation at low priority and in the background. By comparison, you don't want to decide where the item should appear in the list as a low priority background task. If you do, then you either show nothing until all the sort information is ready (in other words, show nothing for a very long time, possibly forever if the file has been archived to tape), or you show everything, but move items around as better information arrives. Neither is a particularly pleasant experience for the user.

What's more, it means that where an item sorts depends on who asks. If you don't have read permission on the shortcut file, then you don't have enough permission to find out whether it's a shortcut to a folder or not, and then Explorer has to decide where “don't know” files go. Do they go with the non-folders by default? Do you create a third “don't know” category? No

matter what you choose, the result is that the location of the file changes depending on who asks. “Hey, where did that shortcut file go? Oh, there it is. What’s it doing over there? Computers are so hard to use.”

Commenter Leo Davidson noted that determining whether the target of a shortcut is a file or folder is cheap if the target is a local hard drive that is not spun down, but that’s an awful lot of *ifs*. Does this mean that you sort shortcuts to local hard drives that are not spun down differently from shortcuts to network drives or shortcuts to drives that are spun down? Won’t users find this confusing? “Hey, where did that shortcut file go? Oh, there it is. What’s it doing over there? Why does this shortcut file show up half of the time in one location and half of the time in that other location? Computers are so hard to use.”

Now, even if it’s a shortcut and the target is on a local hard drive that has not spun down, that still doesn’t mean that you can determine whether or not it is a file or a folder: The target may no longer exist. The act of resolving a broken shortcut is deferred until shortcut is launched, a form of lazy evaluation which avoids having to undertake an expensive search operation until the result is actually needed. If you want to sort shortcuts based on the attributes of the target, you’d have to resolve all the broken shortcuts to see where the target moved to and whether it’s a file or a folder.

Now, you might decide that broken shortcuts are already broken, so who cares what happens? But there are scenarios where *almost all shortcuts are broken* and the user relies on the resolution process to fix them up. But you knew this already. If the Start menu profile roams from one machine to another, the shortcuts are all broken, but when the user decides to launch the shortcut, the resolution process patches them up so they launch the correct program; it just takes a little longer the first time.

I suspect that when Leo Davidson runs the program which sorts shortcuts to folders along with the folders and includes the shortcut target in the column description with “no noticeable effect on performance“, he doesn’t try running it against a server halfway around the world with a 500ms ping time. If you have a folder with 100 shortcuts over a network with 500ms latency, it’ll take you nearly a minute just to sort the items. It’ll be even worse if you have a folder full of files which have been archived to tape. Maybe that scenario isn’t important to him, but it’s important to a lot of people. In fact, one might say that the Explorer folks don’t pay *enough* attention to these scenarios, because at every release of Windows, you can count on those people submitting streams of complaints that the most recent version of Explorer sucks even more on their international corporate network, and then the Explorer team has to do a little redesign work to get things to suck less.

Raymond Chen

**Follow**



