

# Why does Explorer show a + sign even if there are no child folders?

[devblogs.microsoft.com/oldnewthing/20080218-00](http://devblogs.microsoft.com/oldnewthing/20080218-00)

February 18, 2008



Raymond Chen

Mike Williams asked in the Suggestion Box why Explorer shows a + sign next to a folder even though it doesn't have any children. Mind you, this wasn't a suggestion; it was just a random question. I don't enjoy answering random questions about specific details of user interface elements since, as it turns out, I did not write every single user interface element in the system, nor did I attend all of the design meetings. Some people are under the impression that as a senior developer that works with the user interface, I naturally attend every single user interface design meeting, even meetings to discuss things that I am not responsible for. I don't know when these people expect me to get my work done or why I would enjoy going to hundreds of meetings that have nothing to do with my job. These people probably expect a senior electrical engineer at Boeing to attend all meetings on each of the 777's electrical systems, even if her responsibility is just the entertainment system. Another reason I don't enjoy answering these questions is that the person who asks the question is unlikely to be interested in the answer, since the answer is usually boring and technical. Anyway, I don't know the answer, but I'm going to guess. (Actually, that sentence applies to most of what I write. I just guess with confidence.) Recall that Explorer is a view onto the shell namespace, and that each item in the shell namespace has attributes. The attribute that is important here is `SFGAO_HASSUBFOLDER`, which indicates that the item is a folder that *might* have sub-folders. This is a strange backwards-like flag, since it is the absence of the flag that provides definite information. If `SFGAO_HASSUBFOLDER` is not set, then the shell namespace extension is promising that the folder has no sub-folders. If the flag is set, then the folder might have sub-folders, or it might not; no promises. Explorer uses this flag to determine whether to show a plus-sign next to an item in the Explorer folder view. If the flag is set, then a plus sign appears. Yes, this means that the plus sign appears even if there are no sub-folders, but it's better than showing no plus sign when there *are* sub-folders. When you click the plus sign, Explorer goes looking for sub-folders, and if it ultimately finds none, then the plus sign disappears. Why not require that the flag be definite? In other words, why not say that `SFGAO_HASSUBFOLDER` guarantees the presence of a sub-folder, rather than merely hinting at the possibility of a sub-folder? Performance. Determining whether a folder has a sub-folder can be very expensive. For example, if the folder is a directory on a server halfway around the world, hunting for a sub-folder requires enumerating the contents of the directory

looking for anything that's a sub-directory. (Yes, you could try using the `FindExSearchLimitToDirectories` information level, but there is no guarantee that the server halfway around the world supports it.) This is quite an expensive undertaking just to decide whether to show a plus sign, and corporate administrators might not look too fondly on the "unnecessary" network traffic generated by all this probing. What's more, determining whether there is a sub-folder may require additional user interaction. For example, the folder might be on a password-protected FTP server. It would not be a very good design to prompt the user to log in to an FTP server just so Explorer could decide whether or not to show a plus sign next to the server's icon in the folder tree. I believe that Windows Vista sheds a much more skeptical eye on the `SFGAO_HASSUBFOLDER` flag. In part because some shell namespace extensions mis-manage the flag (forgetting to set it when they should) and because some users get upset by the "plus sign that doesn't do anything", Explorer's kicks off a background thread to verify the plus-sign status of each item in the folder tree, or at least verify what it can without causing network administrators grief or disturbing the user. (Users wouldn't like it if you, say, spun up the CD-ROM drive just to check whether or not to show that plus sign.) If you watch carefully, you might see a tiny magnifying glass overlay placed atop the folder icon while this background verification takes place.

The remaining questions in that same comment are even less interesting. Clicking the plus sign will make it go away once Explorer has confirmed that there are no sub-folders after all. Perhaps the namespace extension is slow to enumerate sub-folders? And as for the cancellation of renaming: I don't see this behavior. When a file copy completes in the background, I can still rename whatever I was renaming.

Raymond Chen

**Follow**

