Removing the MAX_PATH restriction on paths applies only to paths

devblogs.microsoft.com/oldnewthing/20250611-00/?p=111262

June 11, 2025

A customer was playing around with Windows long paths and found that even though they used the `\\?\` long path prefix, they still couldn't use a long path.

The documentation on the maximum path limitations say that the `\\?\` prefix is always available. Just to make sure, they set the registry key and created the corresponding manifest entry to enable long paths without the `\\?\` prefix, but that didn't help.

They wanted to know what they were doing wrong and included a short program to demonstrate.

The path in their sample program was `C:\looooooooo[ 1000 more o's ]ooong.txt`. This is well under the 32,767 limit,[1] so why doesn't it work?

Although the `\\?\` prefix and the long path setting raise the path limit to 32,767 characters, the length of each individual component of the path is also subject to a length limit. You can query this limit by calling `GetVolumeInformation` and checking the maximum component length. Values you might see include 255 (exFAT, NTFS), 110 (Joliet CD-RW in Unicode mode), and for network volumes, it's determined by the network protocol.

In the customer's case, they were passing a file name that was over 1000 characters long, which probably exceeded the maximum component length.

They can try again by using a path with longer individual components, where each one is only 100 (say) characters long, but which collectively add up in length to something greater than `MAX_PATH` (260).

[1] Note that the 32,767 limit includes any expansion that occurs during internal processing, so the practical limit for applications is a bit less than that.