Why does FindFirstFile find short names?

devblogs.microsoft.com/oldnewthing/20050720-16

July 20, 2005

Raymond Chen

The `FindFirstFile` function matches both the short and long names. This can produce somewhat surprising results. For example, if you ask for "*.htm", this also gives you the file "x.html" since its short name is "X~1.HTM".

Why does it bother matching short names? Shouldn't it match only long names? After all, only old 16-bit programs use short names.

But that's the problem: 16-bit programs use short names.

Through a process known as generic thunks, a 16-bit program can load a 32-bit DLL and call into it. Windows 95 and the Windows 16-bit emulation layer in Windows NT rely heavily on generic thunks so that they don't have to write two versions of everything. Instead, the 16-bit version just thunks up to the 32-bit version.

Note, however, that this would mean that 32-bit DLLs would see two different views of the file system, depending on whether they are hosted from a 16-bit process or a 32-bit process.

"Then make the `FindFirstFile` function check to see who its caller is and change its behavior accordingly," doesn't fly because you can't trust the return address.

Even if this problem were solved, you would still have the problem of 16/32 interop across the process boundary.

For example, suppose a 16-bit program calls `WinExec("notepad X~1.HTM")`. The 32-bit Notepad program had better open the file X~1.HTM even though it's a short name. What's more, a common way to get properties of a file such as its last access time is to call `FindFirstFile` with the file name, since the `WIN32_FIND_DATA` structure returns that information as part of the find data. (Note: `GetFileAttributesEx` is a better choice, but that function is comparatively new.) If the `FindFirstFile` function did not work for short file names, then the above trick would fail for short names passed across the 16/32 boundary.

As another example, suppose the DLL saves the file name in a location external to the process, say a configuration file, the registry, or a shared memory block. If a 16-bit program program calls into this DLL, it would pass short names, whereas if a 32-bit program calls into the DLL, it would pass long names. If the file system functions returned only long names for 32-bit programs, then the copy of the DLL running in a 32-bit program would not be able to read the data written by the DLL running in a 16-bit program.

Raymond Chen

Follow