What is the lpdwHandle parameter in GetFileVersionInfoSize used for?

devblogs.microsoft.com/oldnewthing/20070731-00

July 31, 2007

Raymond Chen

The `GetFileVersionInfoSize` function returns two pieces of information. The return value is the amount of memory needed to record the version information of a file, and the `DWORD` pointed to by the `lpdwHandle` parameter is set to zero. What's the deal with this strange `lpdwHandle` parameter? That parameter used to do something. The documentation for `GetFileVersionInfo` used to read

> *dwHandle*: The value returned by a preceding call to `GetFileVersionInfoSize` in the `lpdwHandle` parameter.

The purpose of that parameter is to allow `GetFileVersionInfoSize` to pass information to `GetFileVersionInfo` about what it found. In 16-bit Windows and Windows 95, 98, and Me, the `GetFileVersionInfoSize` function opened the target file and went searching for the version information. Once it was located, the size of the version was the return value and the file offset of the version information was stored in `lpdwHandle`. The `GetFileVersionInfo` function was very simple: It merely read `dwLen` bytes from the file starting at file offset `dwHandle`. In the Windows NT series, this mechanism was abandoned. The handle is not used any more. Why not? I don't know, but I have some guesses. First, Windows NT supports files larger than 2GB, so a 32-bit value isn't big enough to hold a file offset value. Second, multitasking introduces a race condition in the `GetFileVersionInfoSize` / `GetFileVersionInfo` pattern. Whereas in 16-bit Windows, nobody could modify the file between the two calls due to co-operative multi-tasking, in 32-bit Windows, it's possible that somebody could sneak in and modify the file between the two calls, resulting in the call to `GetFileVersionInfo` returning garbage. (Yes, Windows 95 has this race condition.) Third, the amount of memory required to load the version resource is not the same as the actual size of the version resource. It's not enough just to seek to the specified location and read `dwLen` bytes from it. For example, a program might load the version resources from a 32-bit module, and we've seen earlier that 32-bit version resources are Unicode. But that program might then call `VerQueryValueA` to retrieve the version string in the ANSI code page. The `GetFileVersionInfo` function needs to return a buffer that can hold not only the actual version resource but also enough memory to hold copies of all the strings in the version resource converted to the ANSI character set so that the `VerQueryValueA` function could return them. Whatever the reason, the Windows NT series of operating systems don't use the handle value. When you call `GetFileVersionInfoSize`, the function looks for the version resource and returns the size of the memory block needed to record it. (Which, as we saw above, includes translation space for the ANSI strings.) When you call `GetFileVersionInfo`, the function starts over from scratch and looks for the version resource and copies it into the buffer. The `dwHandle` parameter is now just a vestigial organ. **Prediction**

People will take this as the opportunity to complain about the `GetFileVersionInfo` family of functions. (Because all I have to do is mention a function name, and that makes it open season on all problems related to that function, as if every function I mention is one that I have total responsibility and authority over.)

Raymond Chen

Follow