The first parameter to VerQueryValue really must be a buffer you obtained from GetFileVersionInfo

devblogs.microsoft.com/oldnewthing/20061226-06

December 26, 2006

Raymond Chen

The documentation for the `VerQueryValue` function states that the first parameter is a "pointer to the buffer containing the version-information resource returned by the `GetFileVersionInfo` function." Some people, however, decide to bypass this step and pass a pointer to data that was obtained some other way, and then wonder why `VerQueryValue` doesn't work. The documentation says that the first parameter to `VerQueryValue` must be a buffer returned by the `GetFileVersionInfo` function for a reason. The buffer returned by `GetFileVersionInfo` is an opaque data block specifically formatted so that `VerQueryValue` will work. You're not supposed to look inside that buffer, and you certainly can't try to "obtain the data some other way". Because if you do, `VerQueryValue` will look for something in a buffer that is not formatted in the manner the function expects. (And it can't even detect that the buffer is improperly formatted because there is no `cbSize` parameter to the `VerQueryValue` function that tells it how big the buffer is. Without that information, `VerQueryValue` can't check that, say, all the internal values are in range, since it doesn't know what "out-of-range" is! Is a value of 4000 out of range? It would be if the buffer were only 3000 bytes long. But it would be okay if the buffer were 5000 bytes long. Since there is no size parameter, the the `VerQueryValue` function is forced to assume that everything is okay.) If it wasn't obvious enough from the documentation that you can't just pass a pointer to a version resource obtained "some other way", it's even more obvious once you see the format of 32-bit version resources. Notice that all strings are stored in Unicode. But if you call the ANSI version `VerQueryValueA` to request a string, the function has to give you a pointer to an ANSI string. There is no ANSI version of the string in the raw version resource, so what can it possibly return? You can't return a pointer to something that doesn't exist. `VerQueryValueA` needs to produce an ANSI string, and it does so from memory that `GetFileVersionInfo` prepared when the resources were extracted.

That there is no `cbSize` parameter to `VerQueryValue` was a failure of the original design of the function, but a slightly more understandable one when you look at the original design of 16-bit version resources: Since 16-bit Windows didn't have Unicode support, there was only one version of `VerQueryValue`, and it operated on ANSI strings. The 16-bit version resource format used ANSI strings, so `VerQueryValue` could just return a pointer into the

raw version resource. No character set conversion was necessary, and therefore no need to reserve additional space as a string conversion buffer, no need for `GetFileVersionInfo` to "prepare" a buffer so that `VerQueryValue` could convert from one character set to another.

Raymond Chen

Follow