Why does SetWindowsHookEx take an HINSTANCE parameter?

devblogs.microsoft.com/oldnewthing/20060811-20

August 11, 2006

Raymond Chen

An anonymous commenter asked why the `SetWindowsHookEx` function takes an `HINSTANCE` parameter if the first thing it’s going to do is convert the instance handle into a file name. Because that’s not how it worked in 16-bit Windows. In 16-bit Windows there was no such thing as hook injection. All 16-bit Windows applications ran in the same address space, so there was no need to inject anything anywhere. Consequently, there was no need to convert the instance handle into a file name in order to inject it. Instead, the instance handle was used to increment the reference count on the module so that the hook procedure wouldn’t get unloaded while the hook was active. When the hook was uninstalled, the module reference count was decremented.

Even in 32-bit Windows, the window manager needs the instance handle in order to convert the function pointer back to an RVA so the function can be located when the module is loaded into another process. If you passed a `LPCWSTR` with a path to the module, the window manager would have to do a `GetModuleHandle` anyway to recover the instance handle. Since most programs have their instance handle readily available, that was the more natural choice. (Not to mention that it would preserve source compatibility with 16-bit Windows, which was an important criterion when trying to get people interested in porting their code to Win32.)

Raymond Chen

Follow