What is so special about the instance handle 0x10000000?

devblogs.microsoft.com/oldnewthing/20121227-00

December 27, 2012

Raymond Chen

A customer wanted to know what it means when the `LoadLibrary` function returns the special value `0x10000000`. Um, it means that the library was loaded at `0x10000000`? Okay, here's some more information: "We're trying to debug an application which loads DLLs and attempts to hook their registry accesses when they call `DllRegisterServer`. It looks like when the special handle is returned from `LoadLibrary`, the registry writes go through and bypass the hook. On the other hand, when a normal value is returned by `LoadLibrary`, the hook works." There is nothing special about the value `0x10000000`. It's an address like any other address. At this point, your psychic powers might start tingling. Everybody who does Win32 programming should recognize that `0x10000000` is the default DLL base address assigned by the linker. If you don't specify a custom base address, the linker will base you at `0x10000000`. Now things are starting to make sense. The DLL being monitored was probably built with the default base address. The value `0x10000000` is special not because of its numeric value, but because it matches the DLL's preferred address, which means that no rebasing has occurred. And this in turn suggests that there's a bug in the registry hooks if the DLL is loaded at its preferred address. The code in question was copied from a book, so now they get to debug code copied from a book. Wait, we're not finished yet. You may have answered the customer's question, but you haven't *solved their problem*.

Hooking and patching DLLs like this is not supported. But what *is* supported is the `RegOverridePredefKey` function. In fact, the `RegOverridePredefKey` was designed *specifically to solve this very problem*:

> The **RegOverridePredefKey** function is intended for software installation programs. It allows them to remap a predefined key, load a DLL component that will be installed on the system, call an entry point in the DLL, and examine the changes to the registry that the component attempted to make.

The documentation continues, explaining how such an installation program might use the `RegOverridePredefKey` function to accomplish the desired task.

Raymond Chen

Follow