

The case of the DLL that refuses to load

 devblogs.microsoft.com/oldnewthing/20131127-00

November 27, 2013



Raymond Chen

A customer reported that they had a problem that occurred only on some machines but not others. Their application called

```
HINSTANCE hinst = LoadLibraryEx(strModule, LOAD_WITH_ALTERED_SEARCH_PATH);
```

and the call succeeded on some machines, but failed on others with error

`ERROR_MOD_NOT_FOUND` (“The specified module could not be found”). The path was a fully-qualified path to a file that was confirmed to exist and be readable.

```
strModule = 0x09e875b4 "C:\Users\Bob\Desktop\CopyAndRun\Contoso.dll"
```

If the `sxe ld Contoso.dll` command was used in the debugger to break when the DLL loads, the breakpoint does fire, but a breakpoint on Contoso’s `DllMain` is never hit. “I think this means that the problem is not that Contoso failed to initialize, but what *does* it mean?”

If you get a break from `sxeld` but no breakpoint on `DllMain`, then it means that the DLL was found but couldn’t be loaded. You can use [loader snaps](#) will [tell you what went wrong](#). “My psychic powers tell me that a dependent DLL could not be found or initialized.”

The customer replied, “Ah, of course. We’ll look into that.”

A short while later, they confirmed the diagnosis. “The Contoso DLL was dependent on a version of the C runtime library that was not installed on the machines where it failed to load. But as a follow-up question: I would have expected that the standard *The program can’t start because XYZ is missing from your computer*. dialog to appear in this case. Why isn’t it?”

The reason is there in the error message: The “missing file” error message is shown only when a program cannot *start* due to a missing file. Specifically, it is raised by the loader only during the initial DLL resolution phase that occurs as part of process initialization. These are the DLLs linked implicitly via the module header because you linked against `kernel32.lib`, for example. DLLs loaded explicitly via `LoadLibrary` do not display this error message; instead, the error is returned back to the program, where it is expected to take

appropriate recovery steps. By comparison, if DLL resolution fails during process initialization, there is nowhere to return the failure code. You can't return it to the program since the program isn't running yet. The only place to put the error is on the screen.

Raymond Chen

Follow

