Why can't I install this DLL via Regsvr32 /i?

devblogs.microsoft.com/oldnewthing/20111117-00

November 17, 2011

Raymond Chen

A customer asked for help installing a particular DLL. They ran the command `regsvr /i SomeDll.dll` but got the error "SomeDll.dll was loaded, but the DllInstall entry point was not found. This file can not be registered." A DLL needs to be specifically written to be used with the `regsvr32` command. You can't just grab some random DLL and expect `regsvr32` to work. As we saw last week, the `regsvr32` program merely loads the specified DLL and calls an entry point established by convention. If the DLL was not written to be used with `regsvr32` then the conventional entry point will not be found, and you get an error message. The `/i` switch to `regsvr32` instructs the program to look for the entry point known as `DllInstall`. By convention, the `DllInstall` function performs installation and setup of a DLL, but since it's just some function exported by a DLL, it could do anything it wants (or nothing at all). You can't just grab a random DLL and expect `regsvr32` to do anything meaningful with it. The DLL has to be designed to operate with `regsvr32`. Handing random DLLs to `regsvr32` is like dialing a random telephone number, sending a tone at 1170Hz and getting upset when you don't get a 2125Hz tone in response. The number one hit for a search on *what does regsvr32 do* is an old Knowledge Base article which explains what `regsvr32` does, and it even contains a sample program which emulates `regsvr32` so you can use it to debug your DLL. (The sample program hasn't been updated to support the `/i` flag, which I leave as an exercise.) One day, I received a piece of email from another employee whom I had never met nor had ever heard of. It didn't even begin with an introduction; it just jumped right in as if we'd been friends for years. "I'm trying to debug a problem where regsvr32 cannot register my DLL. It gives the error 'The specified procedure could not be found.' I saw a blog entry written by you and am trying to understand what our problem is." This blog thing has backfired. The reasons I write these articles is to get people to *stop* asking me questions. (The mechanism for that being to give the answer out in public for everyone to see.) Instead, it turns into "Hi, I found an article you wrote about X, which *ipso facto* makes you not only the world's foremost authority on X, but also the world's leading support technician on X." News flash: Posting a blog entry about something on the Internet should not be taken as evidence that the author is an expert on that subject. (One might argue that it in fact is more likely to be the opposite.) At the time, I wasn't aware of the knowledge base article that explains what `regsvr32` does and how to debug it, so I couldn't point to it. I wrote back, "All `regsvr32` does is `LoadLibrary`, `GetProcAddress`, and

then calls the function. You can write your own test that does the same thing. You do not require any expertise from me." Less than an hour later, I received a reply: "Thanks. I figured it out. There was an older version of the DLL in the path ahead of the one I was trying to register."

And I never did figure out which blog entry I wrote that made them think I was an expert on `regsvr32`. Maybe the person worked in Microsoft Research and used a prototype of their machine that predicts the future, and used it to predict that I was going to write about `regsvr32` two years later.

Raymond Chen

Follow