

How can I force a DLL to register itself if it won't respond to regsvr32?

 devblogs.microsoft.com/oldnewthing/20240528-00

May 28, 2024



Raymond Chen

A customer was having trouble registering a DLL.

We are trying to use the `regsvr32` command to register a DLL that is not already registered, but the command is not working. We get the message “The module ‘contoso.dll’ was loaded but the entry-point `DllRegisterServer` was not found. Make sure that ‘contoso.dll’ is a valid DLL or OCX file and try again.” How can we force contoso.dll to be registered?

Registering a DLL is not a formally-defined operation. It's a general way of saying “Do whatever is required to make it possible to use the DLL in the manner it was intended.” For DLLs that are COM servers, this typically means creating registry keys to advertise the DLL's objects.

There is a convention for DLLs that if they know how to register themselves, they can export a function under the well-known name `DllRegisterServer`, and the function must follow certain rules. Furthermore, a DLL that supports `DllRegisterServer` advertises the fact by adding the string `OLESelfRegister` to their version string file info.

But there is no rule that requires any arbitrary DLL to support this feature.

From the error message, it is apparent that the `contoso.dll` does not support this optional feature. You can't force it to support a feature any more than you can force a program to support the `-?` option or force a book to have a happy ending.

The idea of self-registration comes from the days of ActiveX, when the idea was that users could go to a Web page and install components directly off the Internet. The system would use the `DllRegisterServer` function to tell the DLL, “Okay, you've just been downloaded. Please do whatever is required so that Web pages can start using you.”

ActiveX as a Web technology is dead. Nobody uses it any more. The problem that `DllRegisterServer` was created to solve no longer exists. But the idea of `DllRegisterServer` still carries on. There is nothing inherently Internet-related about `DllRegisterServer`, and some

people still use it to decentralize their setup programs: Instead of putting all of the settings required by their program in one massive file, they let each DLL be responsible for its part of the project.

But the authors of `contoso.dll` didn't do that. The work of setting up the system so that the `contoso.dll` file is ready to use is handled by whatever program installed `contoso.dll`. If you need to register `contoso.dll`, you'll have to run its setup program.