

Linking an Executable to a DLL

Visual Studio 2015

An executable file links to (or loads) a DLL in one of two ways:

- [Implicit linking](#)
- [Explicit linking](#)

Implicit linking is sometimes referred to as static load or load-time dynamic linking. Explicit linking is sometimes referred to as dynamic load or run-time dynamic linking.

With implicit linking, the executable using the DLL links to an import library (.lib file) provided by the maker of the DLL. The operating system loads the DLL when the executable using it is loaded. The client executable calls the DLL's exported functions just as if the functions were contained within the executable.

With explicit linking, the executable using the DLL must make function calls to explicitly load and unload the DLL and to access the DLL's exported functions. The client executable must call the exported functions through a function pointer.

An executable can use the same DLL with either linking method. Furthermore, these mechanisms are not mutually exclusive, as one executable can implicitly link to a DLL and another can attach to it explicitly.

What do you want to know more about?

- [Working with Import Libraries and Export Files](#)
- [Determining which linking method to use](#)
- [The search path used by Windows to locate a DLL](#)

See Also

[DLLs in Visual C++](#)

© 2015 Microsoft