

# How can I get WRL to link my object into its activation factory?

[devblogs.microsoft.com/oldnewthing/20220922-00](https://devblogs.microsoft.com/oldnewthing/20220922-00)

September 22, 2022



Raymond Chen

Recall that under the classical model for linking, symbols in libraries are not included unless they or some other symbol in the same object file is referenced by an explicitly-included object file, or indirectly via a chain of object files.

Component libraries like ATL and WRL use a linker trick to create self-registering objects. ATL has a set of object map macros for doing this, and WRL has its own macros for activatable objects and COM-creatable objects.

But if your object is in a static library, you need to do something to pull in those object files.

For ATL, Larry Osterman used the trick of calling two dummy functions in the object file from dead code.

WRL codifies the mechanism with a set of macros that create explicit references to the magic objects:

```
#define WrlCreatorMapIncludePragma(className)
#define WrlCreatorMapIncludePragmaex(className, serverName)
#define CoCreatableClassWrlCreatorMapInclude(className)
#define CoCreatableClassWrlCreatorMapIncludeEx(className, serverName)
```

Invoke those macros somewhere in the main project (next to your `DllMain` is a fairly obvious place), and they will pull in the object registration information.

Raymond Chen

**Follow**

