

# Fixups are not the same as rewriting code, they're just fixups

 [devblogs.microsoft.com/oldnewthing/20090305-00](http://devblogs.microsoft.com/oldnewthing/20090305-00)

March 5, 2009



Raymond Chen

Classically speaking, a linker cannot rewrite the code the compiler generated; its job is merely to resolve symbols. Of course, resolving symbols means that references to those symbols in the code generated by the compiler turn from “I don’t know” to “Here it is.” Somebody named George appeared to be confused by this, believing that all changes by the linker counted as “rewriting code.”

Obviously if the linker weren’t allowed to change anything at all, there wouldn’t be much point to having a linker. For example, you wouldn’t be able to access functions or variables in a separate compilation unit (for example, from a library) because the compiler doesn’t know what the final address is; that’s what the linker is supposed to figure out. (Indeed, if function-level linking is enabled, then even calls within a compilation unit are left as *insert address here*.) The linker resolves symbols and patches up code references that used to say *insert address here* so that they contain the actual address.

I think George took my statement a bit too literally. The linker can’t change a direct call to an indirect call; that’s not something that the compiler told the linker how to do. All the linker can do is patch up *insert address here* fixups. It can’t say “Oh, I’m going to take your two `mov` instructions and reverse the destination registers, and then flip the arguments to the subsequent `sub` instruction.” (I’m talking about classical linking; the introduction of link-time code generation further blurs the line between the compiler and the linker.)

George also made the common mistake not only of calling this Web site a “document from MSDN” but also applying the same description to a *Microsoft Systems Journal* article from 1997. The back issues of MSJ are included in MSDN as a courtesy. Neither the back issues nor the contents of this Web site are under MSDN editorial control.

Raymond Chen

**Follow**

