

How do I print non-error messages during compilation?

 devblogs.microsoft.com/oldnewthing/20120113-00

January 13, 2012



Raymond Chen

Commenter Worf remarked, “My one wish is that #warning would be supported.” I always find it interesting when people say “I wish that Microsoft would stop following standards,” since the `#warning` directive is nonstandard. The Microsoft C/C++ compiler implements the feature in a method compatible with the standard, namely via a `#pragma` directive. `#pragma message("You really shouldn't be doing that.")` If you want to warn people away from deprecated functionality, you can use the `#pragma deprecated()` directive or the even more convenient (but more standards-troublesome) `__declspec(deprecated)` declaration specifier. The declaration specifier is much more convenient than the preprocessor directive because you can use it in a macro, and you can attach it to specific overloads of a function. (It’s also more standards-troublesome because, while it is still permitted by the standard because it begins with a double-underscore, it is also not required to be ignored by compilers which do not understand it.)

In my experience, however, printing messages during compilation is of little consequence. Print all the messages during compilation as you want; nobody will read them. The only thing that gets attention is an actual warning or error. (And in many cases, only the error will get any attention at all.)

Raymond Chen

Follow

