

Re: does VS C++ 2005 actually work????

Re: does VS C++ 2005 actually work????

Source: <http://www.tech-archive.net/Archive/VC/microsoft.public.vc.mfc/2006-06/msg01155.html>

- *From:* "Jonathan Wood" <jwood@xxxxxxxxxxxxxxxxxxx>
 - *Date:* Wed, 14 Jun 2006 16:54:54 -0600
-

BTW, I really hate the `DWORD_PTR`, `INT_PTR` naming convention. Although they may be the same size as a pointer, they have nothing logically to do with pointers and I think the name is misleading.

--

Jonathan Wood
SoftCircuits Programming
<http://www.softcircuits.com>

"Joseph M. Newcomer" <newcomer@xxxxxxxxxxxxx> wrote in message
<news:cn2192hfpkhjim33g9uvb85hl6o4iqnn08@xxxxxxxxxxx>

I have no problem with warning messages, particularly when the consequences can be serious. But confusing extern "C" with actual C interfaces demonstrates that the concept is overloaded; perhaps it should be extern "nomangle" when you want a C++ method that isn't name-mangled. I tend to compile at high levels of optimization for product code, and it is VERY annoying that Microsoft never understood signed-vs-unsigned (exactly WHAT does it mean to have a negative index of a 0-based array?), and now we have a problem of `sizeof()` and `size_t` generating warnings when used in the context of API calls that only accept a `DWORD` length (and some of them should have been modified to accept `DWORD_PTR` values!). So my product code often has a lot of gratuitous casts to suppress these warnings, and that often scares me because it opens the opportunity for actual bugs to sneak through. But I believe that every build should be "warning-free" at W4 level. I also heavily use lint; usually when I run it, I find a dozen or two bugs in the code I'm sent (I never even start a new project where someone sends me the code if I can't compile it W4 and get a reasonably low count of warnings from lint).

Re: does VS C++ 2005 actually work????

Re: does VS C++ 2005 actually work????

I absolutely despise the assumption that a function can be called without a prototype and the assumption that int is a default return type. Fortunately, I believe both of these are not only deprecated, but will actually be removed from the next C standard.
joe

On Wed, 14 Jun 2006 09:55:51 -0600, "Jonathan Wood"
<jwood@xxxxxxxxxxxxxxxx> wrote:

Of course, this is a different issue from those raised by the OP.

But I've also dealt with these tons of new warnings. On the one hand, it's irritating to have to deal with tons of warnings in working code. But on the other hand, I think the goal is worthwhile. Microsoft gets a lot of justified criticism regarding the reliability of Windows. Most of these problems are due to faulty software running on Windows, most written by folks who would claim they know how to program just fine.

Early versions of C compilers did not warn when unprototyped functions were called. The compiler just assumed undefined functions returned an int. Over the years, compilers have got more and more picky in an effort to impose additional type safety. I think it is reasonable to expect this trend to continue.

Joseph M. Newcomer [MVP]
email: newcomer@xxxxxxxxxxxxx
Web: <http://www.flounder.com>
MVP Tips: http://www.flounder.com/mvp_tips.htm