Subject: [Solved] Default compiler flags
Posted by keltor on Fri, 21 Jun 2013 07:57:40 GMT
View Forum Message <> Reply to Message

Hello all,

I have been working on a program to make some numerical simulations. The other day, just for kicks, I decided to port a part of it into our beloved U++, simply because I like TheIde. I have witnessed, much to my dismay, that the program, compiled in U++, is slower than the exact same program compiled with Code::Blocks, almost by a factor of two!

Edit: Actually, I made a blunder: I was comparing 64-bit code to 32-bit. So it's not a factor of two, but it's still noticeable.

The reason, it seems, is simply that U++ adds default flags depending on the type of compilation (Optimal/Size/Speed) that interfere with the optimisations that I have selected myself. For instance, in Speed mode, U++ chooses -Gy (function-level optimisation), which seems to collide with my choice of -GA (whole-program optimisation). And sadly, it adds the default options at the end of the compilation command, thus overriding the ones I specify.

Does anyone know of a way to tame U++ into accepting my optimisations?

Thanks,

Kel

Subject: Re: Default compiler flags
Posted by dolik.rce on Sun, 23 Jun 2013 17:02:30 GMT
View Forum Message <> Reply to Message

keltor wrote on Fri, 21 June 2013 09:57And sadly, it adds the default options at the end of the compilation command, thus overriding the ones I specify.

Does anyone know of a way to tame U++ into accepting my optimisations?

Thanks,

Kel

Hi Kel,

Judging from a quick glance into the code in MscBuilder.icpp, I believe that the simplest way to force your optimizations would be to set them on package level. The compiler options set for package in Package organizer should be appended at the end of the command-line, thus overriding the previous arguments. I did not test it though, as I don't have windows

It might be actually wise to set such options only for the package(s) that contain computation

heavy code, otherwise your whole binary might get way too big in size and/or take longer to compile.

There are other options too, but I guess that modifying TheIDE just to make it work is not something you'd really prefer

Best regards,
Honza

Subject: Re: Default compiler flags
Posted by keltor on Mon, 24 Jun 2013 06:20:35 GMT
View Forum Message <> Reply to Message

That's a good idea, I didn't realise that the package options took over general options. Probably, as you say, even better than to set global options since it can tweak only those packages that need tweaking. Better indeed than to change TheIDE

Thank you,

Kel

Subject: Re: Default compiler flags
Posted by keltor on Fri, 30 Aug 2013 10:04:30 GMT
View Forum Message <> Reply to Message

Hi again,

I have updated my U++ IDE to version 6254 and it seems that this issue is back. Now, even when I specify compiler flags, some of them get overwritten by the default TheIDE options. For instance, if I set /Ox (maximum optimisation) for Visual Studio 11 x64, I get

cl : Command line warning D9025 : overriding '/Ox' with '/O1'

I have switched back to an older version and it works fine. It seems to me that the new Visual Studio 11 support adds this, to me undesirable, behaviour.

Any thoughts on how to fix?

Keltor

Subject: Re: Default compiler flags

Posted by keltor on Thu, 26 Sep 2013 07:49:42 GMT
View Forum Message <> Reply to Message

I found a workaround, not a solution, for this. One can specify the desired flags under the "Optimize for speed" category, then build the Speed version. Rather annoying if you ask me, but at least it works.

Subject: Re: Default compiler flags
Posted by keltor on Tue, 01 Oct 2013 08:32:28 GMT
View Forum Message <> Reply to Message

Edit: Nevermind, solved it now. My build file got a read-only attribute somehow. Removed it, edited it, now it compiles correctly. Sorry about the trouble.