Subject: shared libraries for debug, static for release ?
Posted by pvozenilek on Mon, 25 Jun 2007 20:51:03 GMT
View Forum Message <> Reply to Message

(This is question from a newbie who still struggles (a lot) to understand how TheIDE works.)

I have a large non-UPP project and try to use TheIDE instead of wxDevC++ (WinXP & Linux, MSVC 7.1 & MingW 4.1.2).

My wish is to build selected packages(?) as shared libraries in debug mode and as static libraries in release mode. This would allow fast compilation and especially linking during the development (I use all the other tricks to reduce these times already and know about the upp linker) and the release application would be single executable.

Is something like this possible within TheIDE so that switching between debug and release modes would automatically do the right thing?

As an aside: Digital Mars C++ compiler is the fastest I know about - it may not compile templates in Core but perhaps it could it be supported as yet another compiler set for non-UPP applications. This would, for me, allow to shrink write-compile-test cycle enough to eliminate a need for embedded scripting language.

/Pavel

Subject: Re: shared libraries for debug, static for release ?
Posted by mirek on Mon, 25 Jun 2007 21:54:14 GMT
View Forum Message <> Reply to Message

pvozenilek wrote on Mon, 25 June 2007 16:51(This is question from a newbie who still struggles (a lot) to understand how TheIDE works.)

I have a large non-UPP project and try to use TheIDE instead of wxDevC++ (WinXP & Linux, MSVC 7.1 & MingW 4.1.2).

My wish is to build selected packages(?) as shared libraries in debug mode and as static libraries in release mode. This would allow fast compilation and especially linking during the development (I use all the other tricks to reduce these times already and know about the upp linker) and the release application would be single executable.

Is something like this possible within TheIDE so that switching between debug and release modes would automatically do the right thing?

No, sorry, you cannot built just "selected" packages as shared - it is either all or nothing. In the same time, there is some - but only a little downside about using shared in Win32 (other than DLL hell) unlike traditional environments, e.g. TheIDE makes all public C symbols for this purpose "dll

public" too...

OTOH, you mention MSVC 7.1., which is to our knowledge the optimal compiler/linker for most of C++ stuff. With incremental linker it posses, there is a little need (except the extremely large applications, more than 2 milions of lines of C++) to use shared libraries just to speedup the build process.

Quote:
As an aside: Digital Mars C++ compiler is the fastest I know about - it may not compile templates in Core but perhaps it could it be supported as yet another compiler set for non-UPP applications. This would, for me, allow to shrink write-compile-test cycle enough to eliminate a need for embedded scripting language.

If I remember well Digital Mars has MSVC compatibility mode via some switch. My interest about this compiler dropped at the moment when I have found it has not even achieved C preprocessor compatibility (unable to interpret some of U++ macros) but I remember trying to compile U++ with it (Note: I have high respect to Walter and his D efforts, even if I think he is wrong

Mirek

Subject: Re: shared libraries for debug, static for release ?
Posted by pvozenilek on Mon, 25 Jun 2007 22:19:23 GMT
View Forum Message <> Reply to Message

"OTOH, you mention MSVC 7.1., which is to our knowledge the optimal compiler/linker for most of C++ stuff. With incremental linker it posses, there is a little need (except the extremely large applications, more than 2 milions of lines of C++) to use shared libraries just to speedup the build process."

Boost is the largest hog - a 1 MLOC translation unit is nothing uncommon. I do include *.cpp files into single TU (which I think Blitz does) to combat with this.

"If I remember well Digital Mars has MSVC compatibility mode via some switch. My interest about this compiler dropped at the moment when I have found it has not even achieved C preprocessor compatibility (unable to interpret some of U++ macros)."

DMC has a wrapper named cl.exe which translates MSVC options and calls dmc.exe. DMC preprocessor was fixed some half an year ago and is able to deal with Boost.Preprocessor, which is more than most other preprocessors. DMC is still not able to deal with the most tricky templates in Boost but for less ambitious projects it should be (the 8.50 beta) OK as is.

Subject: Re: shared libraries for debug, static for release ?
Posted by mirek on Tue, 26 Jun 2007 07:47:49 GMT
View Forum Message <> Reply to Message

pvozenilek wrote on Mon, 25 June 2007 18:19"OTOH, you mention MSVC 7.1., which is to our knowledge the optimal compiler/linker for most of C++ stuff. With incremental linker it posses, there is a little need (except the extremely large applications, more than 2 milions of lines of C++) to use shared libraries just to speedup the build process."

Boost is the largest hog - a 1 MLOC translation unit is nothing uncommon. I do include *.cpp files into single TU (which I think Blitz does) to combat with this.

Yes, BLITZ does this ".cpp merging" automatically.

Quote:
DMC has a wrapper named cl.exe which translates MSVC options and calls dmc.exe. DMC preprocessor was fixed some half an year ago and is able to deal with Boost.Preprocessor, which is more than most other preprocessors. DMC is still not able to deal with the most tricky templates in Boost but for less ambitious projects it should be (the 8.50 beta) OK as is.

Thank you for the info, perhaps it is the time to check again...

Mirek