Subject: Re: Building & using U++ without TheIDE
Posted by mirek on Wed, 12 Sep 2007 12:25:10 GMT
View Forum Message <> Reply to Message

sergei wrote on Wed, 12 September 2007 06:31I hoped moving away from MFC would result in getting away from all the macros, but I see they're alive and kicking...
How does it replace anything with line number? TheIDE feature? I've never seen this available in standard C++.

__LINE__ as a pseudomacro defined by C and C++ standard. It returns the current line number. With a little bit of macro hackery magic, you can use this to build identifier with line number in it.

Quote:
If it's non-standard, why not replace all INITBLOCK with INITBLOCK_(X)-s? There are only about 30 of them.

Not using macros is better than using them. Using macros for things that repeat and cannot be done without macros is better than repeating...

Quote:
t.h was included from Core_init.icpp:

#include "Core.h"

#define TFILE <Core/Core.t>
#include <Core/t.h>

The error with the unmodified file was that LngEntry was undefined (quite understandable, since Core_init.icpp doesn't have NAMESPACE_UPP). Less understandable is that the INITBLOCK actually ceases to function if I add NAMESPACE_UPP to t.h.

Ha! So the same issue again, forgotten file. Thank explains it.

Quote:
BLITZ is pretty impressive if it can understand which CPPs are unused (especially for CPPs sucha s Locale.cpp, that don't have H-s). But that's what I like about static libs - everything has to be compiled, so you can be sure you have no code that is simply unreferenced.

Can you please carefully read again what I wrote about what files are to be compiled? Please...

Quote:
I don't yet understand why Core.h gets reincluded all the time (don't include guards prevent this?).

Yes, they do. That is why it can be included so many times.

Quote:
But I think a rather simple batch could be used to create main.cpp with all CPPs/icpps included, and compile just it. BTW, does the whole static lib always get linked, or only the referenced part of it (if only part may get linked, then single compilation unit isn't a good solution)?

Yes, that is why release mode does not use BLITZ SCU so that .libs are build from many .obj and unreferences .objs can be removed. That is also the solely reason for .icpp - these initialization blocks are not referenced from the rest of the code, that is why if they are put into .lib, linker simply excludes them and no initialization happens.. That is why theide build process does not put them into .lib, but links them directly as .obj -> that is the only difference.

Quote:
Now I have a 408MB static debug library
What's the largest debug exe ever built with U++? Does it even come close?

Optimized release mode about 10MB (that is really big app, about one million lines).

Debug mode (with debug info) theide has about 20-30MB with GCC.

Quote:
I am somewhat worried about stuff I removed, some of it might be used for other platforms (I don't have unix/linux/macosx).

So? Your task was to build libs for Win32, so what is the problem?

Thank you for trying. The fact is, for me and other core U++ developers, lib form of U++ is quite redundant, but I understand that it can be quite atractive for many people. Any effort here is highly welcome...

Mirek