Subject: Re: Building & using U++ without TheIDE
Posted by sergei on Mon, 10 Sep 2007 15:57:54 GMT
View Forum Message <> Reply to Message

Oh, so that's what this is about. Thanks for the info. Quite similar to the approach with stdafx.h and precompiled headers in MSVC. Advantage is rather obvious, but so is the disadvantage - you take on yourself the job of defining compile order, and I'm not sure if that's easier than defining dependencies. Not sure if it affects build time (there shouldn't be any difference IMHO).

So I'll try to guess what the compiler does:
1) Pick a random CPP.
2) See that it needs Core.h, go to Core.h
3) Go from first to last include
4) For every include, define stuff in H and compile stuff in CPP
5) When done, Core.h and most CPPs will be compiled, and rest could be finished
That's the way it should work? If so, in a static lib there's no need for dummy functions in icpp, since nothing is thrown out, right?

Ok, let's see what's in Core.h:

#include <algorithm>
#include <string>
I thought U++ replaced STL with own containers...

I see that all includes are ordered, so it should've worked...
I'll try again at home, but I've found that XML.h, that is included just before Lang.h and i18n.h that I have trouble with, has #include <Core/Core.h>. XML.cpp includes "Core.h" too.
Why not add #pragma once to all H, just in case?

There are also some minor includes inside includes (like AString.hpp and t_.h), but these probably don't affect the big idea.

P.S. what's the difference between #ifdef and #if defined()? Both are used in Core.h.

P.S.2 I've found the tpp help files, is there any external viewer?