Subject: [feature] Blitz gratiously handling problems
Posted by pvozenilek on Thu, 28 Jun 2007 03:32:23 GMT
View Forum Message <> Reply to Message

If compilation by Blitz fails then the system may try to compile individual files one by one. If this succeeds then the files are not compatible and Blist may use this knowledge.

* it may always compile these files individually
* it may use some heuristic to find out which files do not fit together
* it may notify the user about the problem (this may be quite easily a programmer mistake and pointing to it may be very valuable)
* the compatibility test may be tried again from time to time

E.g Boost.Regex library cannot be compiled in Blitz style. (I tried this manually recently and there's something nasty hidden.)

Subject: Re: [feature] Blitz gratiously handling problems
Posted by mirek on Mon, 02 Jul 2007 06:44:51 GMT
View Forum Message <> Reply to Message

In Package organizer, you can switch BLITZ off for individual packages. There is also #pragma BLITZ_PROHIBIT you can use to exclude individual files. Another possibility is in Output mode - but there the settings are not associated with packages/sources -> they have to be individually activated on every TheIDE installation (machine).

I am afraid that there are no simple heurestics available to identify blitz problems - at least none I can see now. Any ideas how this should work?

BTW2, as you seem to compile Boost with U++, one of old ideas was to provide Boost with U++ installation. Do you think this would be possible / good idea?

Note that I generally have ambivalent feelings about boost. I recognize it as high-profile C++ project, but at the same time I think that the STL and C++ standard library design is dead wrong...

Subject: Re: [feature] Blitz gratiously handling problems
Posted by Novo on Mon, 02 Jul 2007 19:47:08 GMT
View Forum Message <> Reply to Message

luzr wrote on Mon, 02 July 2007 02:44
Note that I generally have ambivalent feelings about boost. I recognize it as high-profile C++ project, but at the same time I think that the STL and C++ standard library design is dead wrong...

For some strange reason the rest of the world is thinking differently

STL is quite simple and sometimes not perfect (who is perfect)
BOOST has not too much in common with STL.

The idea of doing everything by yourself is approving but it leads you to a dead end. Of course, you will improve your skills as a developer, but I do not want upp to finish his life as a project I've been using previously (http://codeworker.free.fr/)

Sorry for the offtopic.