Subject: Problem when compiling multiple threads
Posted by mubeta on Thu, 15 Jul 2010 06:39:10 GMT

View Forum Message <> Reply to Message

Hi,

when I compile an GUI MT application, the MINGW compiler say:

C:\upp_2467\uppsrc\Draw\DrawLock.cpp:11: error: thread-local storage not supported for this target
C:\upp_2467\uppsrc\Draw\Font.cpp:273: error: thread-local storage not supported for this target
C:\upp_2467\uppsrc\Draw\Font.cpp:335: error: thread-local storage not supported for this target
C:\upp_2467\uppsrc\Draw\Font.cpp:362: error: thread-local storage not supported for this target
C:\upp_2467\uppsrc\Draw\Font.cpp:363: error: thread-local storage not supported for this target
C:\upp_2467\uppsrc\Draw\Font.cpp:364: error: thread-local storage not supported for this target

??

Subject: Re: Problem when compiling multiple threads
Posted by zsolt on Thu, 15 Jul 2010 07:47:45 GMT

View Forum Message <> Reply to Message

AFAIK, thread local storage is not supported in MINGW.
Switch to MS compiler. It is freely available and much faster than MINGW.

Subject: Re: Problem when compiling multiple threads
Posted by koldo on Thu, 15 Jul 2010 10:47:04 GMT

View Forum Message <> Reply to Message

zsolt wrote on Thu, 15 July 2010 09:47AFAIK, thread local storage is not supported in MINGW.
Switch to MS compiler. It is freely available and much faster than MINGW.
I agree.

Anyway you can use the mingw compiler here:
 http://sourceforge.net/projects/upp/files/upp/1517/mingw_4_4 _0.7z/download

It will work.

Subject: Re: Problem when compiling multiple threads
Posted by mubeta on Thu, 15 Jul 2010 12:06:08 GMT

View Forum Message <> Reply to Message

With MSC9 I only get more troubles:

C:\upp_2467\uppsrc\Core/Core.h(144) : fatal error C1083: Impossibile aprire il file inclusione 'windef.h': No such file or directory.

I found MSC9 in the Microsoft web site, inclded in: Microsoft Visual Studio 10.0. The donwload was without problems.

I used TheIde Automatic setup, lot of time. In the include directory of MSC9 I don't found any "windef.h".

Maybe MINGW will be better.

Subject: Re: Problem when compiling multiple threads
Posted by zsolt on Thu, 15 Jul 2010 12:56:03 GMT
View Forum Message <> Reply to Message

Try this.

Subject: Re: Problem when compiling multiple threads
Posted by mubeta on Sat, 17 Jul 2010 20:54:57 GMT
View Forum Message <> Reply to Message

Ok for MSC9 and SDK32 ... (lot of troubles for donwload all).

Now, compiling the application I get thi s two complains:

C:\upp_2467\uppsrc\Core\Debug.cpp(233) : warning C4390: ';' : empty controlled statement found; is this the intent?

C:\upp_2467\uppsrc\plugin\z\lib\gzio.c(131) : warning C4996: 'strcpy': This function or variable may be unsafe. Consider using strcpy_s instead. To disable d
eprecation, use _CRT_SECURE_NO_WARNINGS. See online help for details.

With google I undertand that only second complain is serius point, but without a standard solution for different compiler, and both can be hide using #pragma ...

Where can do it, without touch U++ code??

Thanks.

Subject: Re: Problem when compiling multiple threads
Posted by mirek on Fri, 23 Jul 2010 09:19:15 GMT

---

mubeta wrote on Sat, 17 July 2010 16:54Ok for MSC9 and SDK32 ... (lot of troubles for donwload all).

Now, compiling the application I get thi s two complains:

C:\upp_2467\uppsrc\Core\Debug.cpp(233) : warning C4390: ';' : empty controlled statement found; is this the intent?

C:\upp_2467\uppsrc\plugin\z\lib\gzio.c(131) : warning C4996: 'strcpy': This function or variable may be unsafe. Consider using strcpy_s instead. To disable d
 eprecation, use _CRT_SECURE_NO_WARNINGS. See online help for details.

With google I undertand that only second complain is serius point, but without a standard solution for different compiler, and both can be hide using #pragma ...

It is not a problem at all. It is just M$ playing smart ass, invalidating most of standard C library for "security reasons".

Happily ignore the warning. In U++ code you will not get these, but plugin/z is opensource library created according to existing standards.

Mirek