Subject: 2020.1 rc
Posted by mirek on Tue, 05 May 2020 09:53:18 GMT
View Forum Message <> Reply to Message

Well, I guess it never will be perfect, but current state is good enough....

https://sourceforge.net/projects/upp/files/upp/2020.1rc/

If there are no serious issues, these files will become 2020.1 release in a couple of days...

Subject: Re: 2020.1 rc
Posted by Novo on Tue, 05 May 2020 19:29:20 GMT
View Forum Message <> Reply to Message

This is, probably, not very important, but still ...
I checked compilation on Linux via "umk CLANG -bsu".
Problems:
benchmarks/stdlist - 'gtk/gtk.h' file not found (weird)
benchmarks/Sort - call to 'Sort__' is ambiguous
benchmarks/CoLoop - copying variable of type 'std::atomic<int>' invokes deleted constructor
benchmarks/BenchmarkTess - crashes Clang ...
benchmarks/AllocMT - use of undeclared identifier 'Parse'; did you mean 'Parser'?
examples/SDLSoundDemo - 'begin_code.h' file not found
reference/SDL2Uword - 'begin_code.h' file not found
reference/UWord_FB - use of undeclared identifier 'K_PLUS'

Subject: Re: 2020.1 rc
Posted by Novo on Tue, 05 May 2020 20:11:54 GMT
View Forum Message <> Reply to Message

In addition to that in Release configuration via "umk CLANG -brus".
reference/FilterStream
reference/Tuple

Subject: Re: 2020.1 rc
Posted by Tom1 on Wed, 06 May 2020 11:38:57 GMT
View Forum Message <> Reply to Message

Hi Mirek,

Everything I need appears to work here with 2020.1 rc!

Thanks and best regards,

Tom

Subject: Re: 2020.1 rc
Posted by mirek on Wed, 06 May 2020 11:42:07 GMT
View Forum Message <> Reply to Message

Novo wrote on Tue, 05 May 2020 21:29This is, probably, not very important, but still ...
I checked compilation on Linux via "umk CLANG -bsu".
Problems:
benchmarks/stdlist - 'gtk/gtk.h' file not found (weird)
benchmarks/Sort - call to 'Sort__' is ambiguous
benchmarks/CoLoop - copying variable of type 'std::atomic<int>' invokes deleted constructor
benchmarks/BenchmarkTess - crashes Clang ...
benchmarks/AllocMT - use of undeclared identifier 'Parse'; did you mean 'Parser'?
examples/SDLSoundDemo - 'begin_code.h' file not found
reference/SDL2Uword - 'begin_code.h' file not found
reference/UWord_FB - use of undeclared identifier 'K_PLUS'

Thanks. Might help for the next release.

That said, benchmarks are just development, problems there are OK. SDLSoundDemo / SDL2Uword - it seems like there might be a problem with your SDL2 installation.

Subject: Re: 2020.1 rc
Posted by Novo on Wed, 06 May 2020 14:58:49 GMT
View Forum Message <> Reply to Message

mirek wrote on Wed, 06 May 2020 07:42
SDLSoundDemo / SDL2Uword - it seems like there might be a problem with your SDL2 installation.
Nope. My SDL2 installation installation is fine.
I have libsdl2-dev installed.
This is a bug with SDL2 itself.
Line #include "begin_code.h" in SDL_platform.h is supposed to look like
#include <SDL2/begin_code.h>
For the time being it can be fixed by adding "/usr/include/SDL2" as an include path for POSIX platforms to the Synth package ...

Subject: Re: 2020.1 rc
Posted by mirek on Wed, 06 May 2020 16:45:42 GMT
View Forum Message <> Reply to Message

That is rather looking like compiler bug. #include " is supposed to look into the current directory first.

Really weird. I have got #include "begin_code.h" in SDL_platform.h and everything works just fine, no fixes nor include paths added anywhere....

Subject: Re: 2020.1 rc
Posted by Novo on Wed, 06 May 2020 17:47:36 GMT
View Forum Message <> Reply to Message

mirek wrote on Wed, 06 May 2020 12:45That is rather looking like compiler bug. #include " is supposed to look into the current directory first.

Really weird. I have got #include "begin_code.h" in SDL_platform.h and everything works just fine, no fixes nor include paths added anywhere....
Below is what I have on my machine (Ubuntu 20.04):

#include <SDL2/_real_SDL_config.h> (SDL_config.h)
_real_SDL_config.h is located in /usr/include/x86_64-linux-gnu/SDL2/
It contains #include "SDL_platform.h", which is supposed to be located in /usr/include/x86_64-linux-gnu/SDL2/
And it can be found there. It is just a link: SDL_platform.h -> ../../SDL2/SDL_platform.h
So, compiler is looking for "begin_code.h" in /usr/include/x86_64-linux-gnu/SDL2/, but it is supposed to be looking in /usr/include/SDL2/ ...
I doubt this is a bug with Clang ...

Subject: Re: 2020.1 rc
Posted by Novo on Wed, 06 May 2020 17:54:36 GMT
View Forum Message <> Reply to Message

Well, I checked the same project with GCC, and it is fine.
Looks like a bug with either Clang or GCC ...

Subject: Re: 2020.1 rc
Posted by Novo on Wed, 06 May 2020 18:12:06 GMT
View Forum Message <> Reply to Message

No matter whose fault is that, SDL2 cannot be used with Clang on Linux without adding of a system-wide include path ...

Subject: Re: 2020.1 rc
Posted by mirek on Wed, 06 May 2020 22:05:19 GMT
View Forum Message <> Reply to Message

Novo wrote on Wed, 06 May 2020 20:12No matter whose fault is that, SDL2 cannot be used with Clang on Linux without adding of a system-wide include path ...

I thing you generalize too much. It works out of box in ubuntu and rapsbian.. Either way, this is not our problem to fix.

Mirek

Subject: Re: 2020.1 rc
Posted by Novo on Thu, 07 May 2020 03:56:44 GMT
View Forum Message <> Reply to Message

mirek wrote on Wed, 06 May 2020 18:05Novo wrote on Wed, 06 May 2020 20:12No matter whose fault is that, SDL2 cannot be used with Clang on Linux without adding of a system-wide include path ...

I thing you generalize too much. It works out of box in ubuntu and rapsbian.. Either way, this is not our problem to fix.

Mirek
This is entirely up to you. :roll:
I'm just a messenger. :roll:
I just tested U++ and found several problems.

Subject: Re: 2020.1 rc
Posted by Novo on Thu, 07 May 2020 04:02:14 GMT
View Forum Message <> Reply to Message

Another problem. I tried to build a shared object on Linux using flag DLL and got a bunch of "recompile with -fPIC" messages.
Code wasn't compiled with -fPIC.
Sorry for reporting problems ... :blush:

Subject: Re: 2020.1 rc
Posted by mirek on Thu, 07 May 2020 07:14:28 GMT
View Forum Message <> Reply to Message

Try to add "SO" too.

Subject: Re: 2020.1 rc
Posted by Novo on Thu, 07 May 2020 16:00:44 GMT
View Forum Message <> Reply to Message

mirek wrote on Thu, 07 May 2020 03:14Try to add "SO" too.
Thanks.
"SO" has a different purpose. It turns each module into a shared object.
"SO" allows me to compile my project, but instead of one shared object I get a whole bunch of them.
Another small issue. My DLL is named "project.dll" on Linux ... :roll:

Subject: Re: 2020.1 rc
Posted by mirek on Thu, 07 May 2020 16:20:28 GMT
View Forum Message <> Reply to Message

Novo wrote on Thu, 07 May 2020 18:00mirek wrote on Thu, 07 May 2020 03:14Try to add "SO" too.
Thanks.
"SO" has a different purpose. It turns each module into a shared object.
"SO" allows me to compile my project, but instead of one shared object I get a whole bunch of them.
Another small issue. My DLL is named "project.dll" on Linux ... :roll:

I am afraid that what you need is not yet supported.

We can do that for the next release.

Mirek