
Subject: Re: 2020.1 officially released

Posted by [mirek](#) on Sat, 30 May 2020 06:57:48 GMT

[View Forum Message](#) <> [Reply to Message](#)

Novo wrote on Sat, 30 May 2020 07:13: Interesting message when compiling on Windows with CLANGx64:

#0 0x0000000142abaa9c (clang-10+0x2abaa9c)

#1 0x0000000142a52a8c (clang-10+0x2a52a8c)

#2 0x0000000142a6b016 (clang-10+0x2a6b016)

#3 0x0000000142a60146 (clang-10+0x2a60146)

clang-10: error: clang frontend command failed due to signal (use -v to see invocation)

clang version 10.0.0 (<https://github.com/llvm/llvm-project.git>

d32170dbd5b0d54436537b6b75beaf44324e0c28)

Target: x86_64-w64-windows-gnu

Thread model: posix

InstalledDir: c:\local\apps\upp\bin\clang\bin

clang-10: note: diagnostic msg: PLEASE submit a bug report to <https://bugs.llvm.org/> and include the crash backtrace, preprocessed source, and associated run script.

heapdbg.cpp

CharSet.cpp

t.cpp

z.cpp

lz4.c

xxhash.c

clang-10: note: diagnostic msg:

PLEASE ATTACH THE FOLLOWING FILES TO THE BUG REPORT:

Preprocessed source(s) and associated run script(s) are located at:

clang-10: note: diagnostic msg: C:\users\ssg\Temp\wkparser\$blitz-44ef80.cpp

clang-10: note: diagnostic msg: C:\users\ssg\Temp\wkparser\$blitz-44ef80.sh

clang-10: note: diagnostic msg:

plugin/bz2: 2 file(s) built in (0:00.37), 186 msec / file

plugin/bmp: 4 file(s) built in (0:00.30), 76 msec / file

dvlp/ctrl/about: 1 file(s) built in (0:00.33), 332 msec / file

Painter: 28 file(s) built in (0:01.72), 61 msec / file

dvlp/wiki/wkparser: 6 file(s) built in (0:00.47), 79 msec / file

Draw: 35 file(s) built in (0:00.47), 13 msec / file

RichText: 22 file(s) built in (0:00.50), 23 msec / file

CtrlCore: 62 file(s) built in (0:00.87), 14 msec / file

CtrlLib: 58 file(s) built in (0:02.91), 50 msec / file

Core: 68 file(s) built in (0:05.87), 86 msec / file

There were errors. (0:17.72)

program finished with exit code 1

elapsedTime=18.027474

And this is primarily C++98 code with a lot of templates.

I mean my own app.

It would be good to narrow that down to single file / construct and send them report.

For what is worth, for me CLANG was rock stable so far, except maybe some issues with debugger info (but those unfortunately exist with msc as well).

Mirek
