
Subject: Re: Upp package binding for LLVM/Clang library (libclang)

Posted by [Sender Ghost](#) on Sun, 20 Apr 2014 20:19:53 GMT

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

Hello Koldo.

koldo wrote on Sun, 20 April 2014 15:54 I have tried to install clang on TheIDE in windows unsuccessfully. I have used the precompiled files.

You didn't say about which precompiled files you used, but if they are official v3.4 "Pre-built Binaries: Clang for Windows" from llvm.org download page, then they didn't work here also (perhaps, because of 64-bit compiler build, which don't support 32-bit Windows operating system). koldo wrote on Sun, 20 April 2014 15:54 I have added the include, lib and path directories, and added clang++ as "Compiler Name". However the compiler does not find `#include <typeinfo>` in `Core.h`.

The plain "include" directory from MinGW is not enough to configure Clang compiler on Windows, in case of unsupported search directories in the source code. To solve this with TheIDE, you could add full include directories, as explained in 40030 message.

koldo wrote on Sun, 20 April 2014 15:54 Just removing clang++ as "Compiler Name" and the files are compiled without problem (with MinGW).

Yes, because of "c++" compiler name by default for GCC builder.

In conclusion, I recommend to download/install necessary software and build LLVM/Clang by yourself, e.g. as explained in 30322 message, to try it out. But you could try to find different precompiled LLVM/Clang (e.g. from <http://www.drangon.org/mingw/>) or use following temporary links (by clicking "Direct download link"):

LLVM/Clang v3.3 release (32-bit, 86.5 Mb).

LLVM/Clang v3.4 release (32-bit, 128 Mb).

The last working version for GCC builder of TheIDE is 3.3. The 3.4 version uses error messages instead of warning messages for unsupported compiler/linker flags (e.g. "-finline-limit", "-mwindows", "-mconsole", etc.) and will not build (e.g., without excluding/changing some flags for GCC builder). But this is not true, if you just need to use libclang library, compiled by MinGW GCC, with GCC compiler.
