Subject: Re: 2022.3rc5

Posted by Oblivion on Wed, 08 Feb 2023 16:55:24 GMT

View Forum Message <> Reply to Message

Hello Mirek,

I've stripped the code, and wrote the below test:

```
struct Clang {

 CXIndex index = nullptr;
 CXTranslationUnit tu = nullptr;

 Clang()
 {
  MemoryIgnoreLeaksBlock __;
  index = clang_createIndex(0, 0);
 }

 void Parse()
 {
  Vector<const char*> args = {
   "-std=c++14",
   "-xc++",
   "-I/usr/include/llvm",
   "-I/usr/include/c++",
   "-I/usr/include/c++/12.2.1",
   "-I/usr/include/c++/12.2.1/x86_64-pc-linux-gnu",
   "-I/usr/include/c++/12.2.1/backward",
   "-I/usr/lib/clang/15.0.7/include",
   "-I/usr/include"
   "-I/usr/local/include",
  };

  tu = clang_parseTranslationUnit(
      index,
      "/home/user/test.cpp",
                        args,
                        args.GetCount(),
                        nullptr,
                        0,
                        CXTranslationUnit_None
                 );
 }

 ~Clang()
 {
  MemoryIgnoreLeaksBlock __;
  if(tu) clang_disposeTranslationUnit(tu);
  clang_disposeIndex(index);
 }

CONSOLE_APP_MAIN
{
 StdLogSetup(LOG_COUT);
 Clang().Parse();
}
```

This works. clang_parseTranslationUnit() returns a translation unit handle every single time. (And the handle can be successfully used to traverse the AST.

However, the same code applied to TheIDE's Clang::Parse() method, with hard coded paths as is above,
clang_parseTranslationUnit() still fails to return a translation unit every single time.

I've tried both dynamic loading and static linking (LCLANG), tested both the above code & TheIDe on Linux 6.1.9/Clang 14 & 15

At this point I am almost sure that this is not a user-side problem, because I have installed vanilla ArchLinux on both a real hardware and on a VM (on windows machine), yet I get the same failure... Could this be a process env issue? Any ideas?

Best regards,
Oblivion