Subject: Re: TheIDE support for Visual Studio Build Tools 2022?
Posted by Lance on Sun, 19 Dec 2021 14:44:36 GMT
View Forum Message <> Reply to Message

I tried, somehow I could not figure it out.

```
void InstantSetup()
{
 bool dirty = false;
 String default_method;

 String bin = GetExeDirFile("bin");

 if(DirectoryExists(bin + "/clang"))
  for(int x64 = 0; x64 < 2; x64++) {
   String method = x64 ? "CLANGx64" : "CLANG";
  #ifdef INSTANT_TESTING
   method << "Test";
  #endif
   VectorMap<String, String> bm = GetMethodVars(method);

   Vector<String> bins = Split(bm.Get("PATH", ""), ';');
   Vector<String> incs = Split(bm.Get("INCLUDE", ""), ';');
   Vector<String> libs = Split(bm.Get("LIB", ""), ';');
  #ifdef INSTANT_TESTING
   if(CheckDirs(bins, 3) && CheckDirs(incs, 2) && CheckDirs(libs, 2)) {
    if(!x64)
     default_method = Nvl(default_method, method);
    continue;
   }
  #endif

   bmSet(bm, "BUILDER", "CLANG");
   bmSet(bm, "COMPILER", x64 ? "" : "i686-w64-mingw32-c++");
   bmSet(bm, "COMMON_OPTIONS", "");
   bmSet(bm, "COMMON_CPP_OPTIONS", "");
   bmSet(bm, "COMMON_C_OPTIONS", "");
   bmSet(bm, "COMMON_LINK", "");
   bmSet(bm, "COMMON_FLAGS", "");
   bmSet(bm, "DEBUG_INFO", "2");
   bmSet(bm, "DEBUG_BLITZ", "1");
   bmSet(bm, "DEBUG_LINKMODE", "0");
   bmSet(bm, "DEBUG_OPTIONS", "");
   bmSet(bm, "DEBUG_FLAGS", "");
   bmSet(bm, "DEBUG_LINK", "-Wl,--stack,20000000");
   bmSet(bm, "RELEASE_BLITZ", "1");
   bmSet(bm, "RELEASE_LINKMODE", "0");
   bmSet(bm, "RELEASE_OPTIONS", "-O3 ");
```

```
  bmSet(bm, "RELEASE_FLAGS", "");
  bmSet(bm, "RELEASE_LINK", "-Wl,--stack,20000000");
  bmSet(bm, "DEBUGGER", "gdb");
  bmSet(bm, "ALLOW_PRECOMPILED_HEADERS", "1");
  bmSet(bm, "DISABLE_BLITZ", "");

 //  bmSet(bm, "LINKMODE_LOCK", "0");

  String clang = bin + "/clang";

  bins.At(0) = clang + "/bin";
  bins.At(1) = clang + (x64 ? "/x86_64-w64-mingw32/bin" : "/i686-w64-mingw32/bin");
  bins.At(2) = GetExeDirFile(x64 ? "bin/SDL2/lib/x64" : "bin/SDL2/lib/x86");
  bins.At(3) = GetExeDirFile(x64 ? "bin/pgsql/x64/bin" : "bin/pgsql/x86/bin");
  bins.At(4) = GetExeDirFile(x64 ? "bin/mysql/lib64" : "bin/mysql/lib32");

  incs.At(0) = GetExeDirFile("bin/SDL2/include");
  incs.At(1) = GetExeDirFile(x64 ? "bin/pgsql/x64/include" : "bin/pgsql/x86/include");
  incs.At(2) = GetExeDirFile(x64 ? "bin/mysql/include" : "bin/mysql/include");

  libs.At(0) = GetExeDirFile(x64 ? "bin/SDL2/lib/x64" : "bin/SDL2/lib/x86");
  libs.At(1) = GetExeDirFile(x64 ? "bin/pgsql/x64/lib" : "bin/pgsql/x86/lib");
  libs.At(2) = GetExeDirFile(x64 ? "bin/mysql/lib64" : "bin/mysql/lib32");

  bm.GetAdd("PATH") = Join(bins, ";");
  bm.GetAdd("INCLUDE") = Join(incs, ";");
  bm.GetAdd("LIB") = Join(libs, ";");

  SaveVarFile(ConfigFile(method + ".bm"), bm);
  dirty = true;

  if(x64)
   default_method = Nvl(default_method, method);
 }

 enum { VS_2015, VS_2017, BT_2017, VS_2019, VSP_2019, BT_2019, VS_2022 };
 DirFinder df;

 for(int version = /*VS_2012*/VS_2022; version <= VS_2022; version++)
  for(int x64 = 0; x64 < 2; x64++) {
//   String x86method = decode(version, VS_2015, "MSVS15",
//                             VS_2017, "MSVS17", BT_2017, "MSBT17",
//                             VS_2019, "MSVS19", VSP_2019, "MSVC19P", BT_2019, "MSBT19",
//                             "MSBT");
   String x86method = decode(version, VS_2015, "MSVS15",
                             VS_2017, "MSVS17", BT_2017, "MSBT17",
                             VS_2019, "MSVS19", VSP_2019, "MSVC19P", BT_2019, "MSBT19",
                             VS_2022, "MSVS22", "MSVS22");
```

```
   String x64s = x64 ? "x64" : "";
   String method = x86method + x64s;
//   String builder = decode(version, VS_2015, "MSC15",
//                           VS_2017, "MSC17", BT_2017, "MSC17",
//                           VS_2019, "MSC19", VSP_2019, "MSC19", BT_2019, "MSC19",
//                           "MSC19"
   String builder = decode(version, VS_2015, "MSC15",
                           VS_2017, "MSC17", BT_2017, "MSC17",
                           VS_2019, "MSC19", VSP_2019, "MSC19", BT_2019, "MSC19",
                           VS_2022, "MSC22", "MSC22"
                ) + ToUpper(x64s);

  #ifdef INSTANT_TESTING
   method << "Test";
  #endif

   String vc, bin, inc, lib, kit81;

   VectorMap<String, String> bm = GetMethodVars(method);
   Vector<String> bins = Split(bm.Get("PATH", ""), ';');
   Vector<String> incs = Split(bm.Get("INCLUDE", ""), ';');
   Vector<String> libs = Split(bm.Get("LIB", ""), ';');
  #ifdef INSTANT_TESTING
   if(CheckDirs(bins, 2) && CheckDirs(incs, 4) && CheckDirs(libs, 3)) {
    if(x64)
     default_method = Nvl(default_method, x86method);

    continue;
   }
  #endif

   if(version == VS_2015)
    vc = df.Get("/microsoft visual studio 14.0/vc",
"bin/cl.exe;bin/lib.exe;bin/link.exe;bin/mspdb140.dll");
   else
//    vc = df.Get(decode(version, BT_2017, "/microsoft visual studio/2017/buildtools/vc/tools/msvc",
//                               VS_2017, "/microsoft visual studio/2017/community/vc/tools/msvc",
//                               BT_2019, "/microsoft visual studio/2019/buildtools/vc/tools/msvc",
//                               VS_2019, "/microsoft visual studio/2019/community/vc/tools/msvc",
//                               VSP_2019, "/microsoft visual studio/2019/professional/vc/tools/msvc",
//                               ""),
    vc = df.Get(decode(version, BT_2017, "/microsoft visual studio/2017/buildtools/vc/tools/msvc",
                               VS_2017, "/microsoft visual studio/2017/community/vc/tools/msvc",
                               BT_2019, "/microsoft visual studio/2019/buildtools/vc/tools/msvc",
                               VS_2019, "/microsoft visual studio/2019/community/vc/tools/msvc",
                               VSP_2019, "/microsoft visual studio/2019/professional/vc/tools/msvc",
                               VS_2022, "/Microsoft Visual Studio/2022/Community/VC/Tools/MSVC",
                               //VS_2022, "/microsoft visual studio/2022/community/vc/tools/msvc",
```

```
                ""),
            x64 ? "bin/hostx64/x64/cl.exe;bin/hostx64/x64/mspdb140.dll"
                : "bin/hostx86/x86/cl.exe;bin/hostx86/x86/mspdb140.dll");

    bin = df.Get("/windows kits/10/bin", "x86/makecat.exe;x86/accevent.exe");
    inc = df.Get("/windows kits/10", "um/adhoc.h");
    lib = df.Get("/windows kits/10", "um/x86/kernel32.lib");
```

for unknown reason, vc is empty even when I am sure I have vs community 2022 installed in default location.

Also, can we make it smarter? the directory structure is quite stable across versions: they seems to differ by the year number. Ideally we can assume future version will not break the tradition until it does. That way we don't need to modify source code every time VS has a new version.
