Subject: Re: U++ 2017.2 released
Posted by cbpporter on Tue, 09 Jan 2018 12:06:36 GMT
View Forum Message <> Reply to Message

mirek wrote on Tue, 09 January 2018 13:38

Any help with that is highly appreciated.

There is nothing magical about the detection, check "ide/InstantSetup.cpp".

Unfotunately, it appears to be really fragile. ("it works on my machine"). I actually think that things got broken when I was "forced" by the community to replace dump "search through all directories" approach to using registry entries to "make it fast". (check svn history)

Maybe it is the time to get back to dumb.

Mirek

Well, I did not force anybody, but I was very vocal about it. I don't think that searching 5 minutes is OK when registry auto-detection can work, but disk based can be used as a backup, after a prompt.

But I believe you that it is very fragile. Here is the weird thing: every time I install VS on a new computer, the folders surprise me. I do believe you 100% that you are facing an infinite row of "works for me" scenarios.

But here is the thing. I am working on something with some similarity to U++ and I need auto-detection. I was shamelessly inspired by your old school auto-detection idea and code and use a similar method.

I need to update my method for MSC15, but I need to autodetect MSC10-14, MINGW and Clang, on Win and Linux.

It may be a case of "works for me", but my methods work for me (:)), while the U++ methods fluctuate. My methods have never failed across 5-6 computers in the last few years and I am often forced to maintain valid MSC10-14 installs on random computers, because they are supported, but not really used by anybody.

UT tests do test auto-detection often.

So it can be done.

One key progress in reliability was forgetting about assigning a given SDK a known code, like "10.0.16299.0". I search though all the folders and look for "um"/"ucrt" sub-folders with lib files inside, then order the valid finds. The method is very random, but I found it works though trial and error.

I do this after I have "detected" a SDK, to remove duds:

```
bool BuildMethod::TestLib(bool px86, bool px64) {
 if (Compiler.IsEmpty() || Sdk.IsEmpty())
  return false;

 if (px86) {
  Vector<String> x86um;
  Vector<String> x86ucrt;

  if (FindFile(Sdk + "\\lib\\*.lib"))
   x86um.Add("\\lib\\");
  else if (FindFile(Sdk + "\\lib\\x86\\*.lib"))
   x86um.Add("\\lib\\x86\\");
  else if (FindFile(Sdk + "\\lib\\win8\\um\\x86\\*.lib"))
   x86um.Add("\\lib\\win8\\um\\x86\\");
  else if (FindFile(Sdk + "\\lib\\winv6.3\\um\\x86"))
   x86um.Add("\\lib\\winv6.3\\um\\x86\\");
  else {
   FindFile ff(Sdk + "\\lib\\*");
   while (ff) {

    if (ff.IsDirectory()) {
     String s = ff.GetName();
     if (s != ".." && s != ".") {
      String p = ff.GetPath();
      if (FileExists(p + "\\um\\x86\\User32.Lib")) {
       x86um.Add("\\lib\\" + ff.GetName() + "\\um\\x86\\");
       //break;
      }
      if (FileExists(p + "\\ucrt\\x86\\*.lib"))
       x86ucrt.Add("\\lib\\" + ff.GetName() + "\\ucrt\\x86\\");
     }
    }
    ff.Next();
   }
  }

  if (x86um.GetCount() == 0)
   return false;
  for (int i = x86um.GetCount() - 1; i >= 0; i--) {
   Lib32 << Sdk + x86um[i];
  }
  for (int i = x86ucrt.GetCount() - 1; i >= 0; i--) {
   Lib32 << Sdk + x86ucrt[i];
  }
  if (FileExists(Compiler + "\\VC\\lib\\*.lib"))
   Lib32 << Compiler + "\\VC\\lib\\";
 }
```

I will look over ide/InstantSetup.cpp and see what goes on there. Haven't looked in years and maybe I can find your MSC15 detection and try to fix it. And 15...

I can also add the whole BuildMethod source if you wish.

But first I'm having still problems. It looks like setting a target version override with packages that use the resource compiler cause F5 to repeatedly "link", but never run. Ctrl-F5 works.

You can test this with EyeTests. EyeTests works just fine, but adding "target file oveeride" causes the linking loop.

I get this every F5:

Linking...
LINK : c:\z2c\eye.exe not found or not built by the last incremental link; performing full link
c:\z2c\eye.exe (B) linked in (0:01.23)

All my projects use this override.
