Subject: Re: U++ 2017.2 released
Posted by cbpporter on Wed, 10 Jan 2018 09:25:55 GMT
View Forum Message <> Reply to Message

It looks like the problem with VS2015 is that in the SDK folder, some files are not in the x86 subfolder, but in 10.0.16299.0/x86.

Do we need those binaries too? Are the VC ones + includes + libs not enough?

This is the quick personal fix:

bin = df.Get(x64 ? "/windows kits/10/bin/10.0.16299.0/x86" : "windows kits/10/bin/10.0.16299.0/x86", "makecat.exe;accevent.exe");

This fixes both VS 2017 and VS 2015.

A real fix will be needed that looks in all the subfolders, not just 10.0.16299.0.

But first the question is: is it my install or the fact that I added VS2017 on top of 2015. I'll test in the VM that only has 2017 and on one with only VS2015 and get back to you with the results.

But at least I can auto-detect and work.

Almost...

With "Target file override" I still can't debug and get an attempted link every time I try and debug, on both 2015 and 2017, like I described in the EyeTest scenario:

```
Linking...
LINK : c:\z2c\zide.exe not found or not built by the last incremental link; performing full link
   Creating library c:\z2c\zide.lib and object c:\z2c\zide.exp
c:\z2c\zide.exe  ( B) linked in (0:01.64)
```

Quote:
No reason to bother with VS2015

That is a very good choice if you pretty much want to guarantee continual U++ obscurity.