Subject: U++ 2017.2 released
Posted by mirek on Thu, 14 Dec 2017 13:30:14 GMT
View Forum Message <> Reply to Message

2017.2 (rev 11540) (Dec 2017)

Core
* New AsyncWork class - U++ future-alike tool
* CoWork: Exception propagation from worker threads, job cancellation support.
* Bits class optimized, it is now possible to set multiple bits (count parameter)
* WebSocket refactored for asynchronous operation
* UTF-32 support functions, UnicodeCompose and UnicodeDecompose functions, support for UTF16 surrogate pairs
* DeleteFolderDeep now deletes symlinks
* MakeOne function (alternative to One::Create)
* LoadFromFile, StoreToFile - Event variant
* ReverseRange and SortedRange
* Server Name Indicator support in Socket and HttpRequest
* SSL1.1 API support
* RegisterGlobalSerialize

GUI programming
* ArrayCtrl: Improvements in sorting, new public GetCellRect, GetCellRectM, GetSelKeys, SetRowDisplay, SetColumnDisplay, AcceptRow methods
* Button NakedStyle style.
* WithDropChoice: Set/Remove entry in drop-down list
* TextCtrl::Load refactored, now can break very long (>1MB) lines
* TreeCtrl: GetOpenIds, OpenIds
* IdCtrls, SqlCtrls refactored, improved
* Option: SetColor
* Edit...Spin: RoundFromMin modifier method
* ColorPopup now has button to enter the color as text
* New features in ScatterDraw, ScatterCtrl
* Header / Footer can now be changed using paragraph properties
* QTFDisplay now ignores text color / paper if style is not normal

TheIDE
* Improved debugging of console applications with MinGW
* MacroManager
* Explicit Go to line.. operation (in addition to Navigator)
* BRC now supports LZ4, LZMA and Zstd compression
* File comparison tool improved
* Directory comparison now can show/filter new/modified/removed files
* Dark theme highlighting
* Duplicate package function
* FindAll button in normal Search (finds all occurences in current file)
* HexView now remembers position
* Selection can be interpreted as error list

* now shows the number of selected characters
* tab size now possible on per-file basis
* Repository synchronize refactored, now supports both svn and git
* Layout visgen improvements
* theide now supports Visual Studio 2017 C++ compiler auto setup
* MS build tools autosetup

3rd party modules
* MinGW 64 updated to GCC-7.2.0
* OpenSSL for VC++ updated to 1.0.2l
* plugin/lz4: lz4 updated to 1.7.3
* plugin/bz2: updated to 1.0.6
* plugin/jpg: updated to version 9b
* plugin/pcre: updated to 8.41
* plugin/png: updated to 1.2.57
* plugin/sqlite3: updated to 3.17.9
* plugin/z: Updated to 1.2.11
* plugin/Eigen: Updated to version 3.3.4

Subject: Re: U++ 2017.2 released
Posted by Tom1 on Thu, 14 Dec 2017 14:51:13 GMT
View Forum Message <> Reply to Message

Hi!

Congratulations and thanks again to everybody involved! I really appreciate your work on this excellent platform.

Best regards,

Tom

Subject: Re: U++ 2017.2 released
Posted by koldo on Fri, 15 Dec 2017 21:56:23 GMT
View Forum Message <> Reply to Message

:)

Subject: Re: U++ 2017.2 released
Posted by cbpporter on Mon, 08 Jan 2018 09:55:07 GMT
View Forum Message <> Reply to Message

Great! Thanks!

Been away for a while, time to upgrade...

Subject: Re: U++ 2017.2 released
Posted by cbpporter on Mon, 08 Jan 2018 10:14:49 GMT
View Forum Message <> Reply to Message

cbpporter wrote on Mon, 08 January 2018 11:55Great! Thanks!

Been away for a while, time to upgrade...
Hmm, it works fine for now, but I think there was some sort of first time setup in the past? That step was gone for me.

Plus, clicking on "Automatic build methods setup.." with no VS installed that it recognizes does nothing. A prompt would be welcome.

Otherwise, I'm installing VS 2017 community.

I think I mention this at least once a year, but Microsoft SDK conventions are an ever increasing clusterfuck...

Subject: Re: U++ 2017.2 released
Posted by cbpporter on Mon, 08 Jan 2018 10:29:39 GMT
View Forum Message <> Reply to Message

cbpporter wrote on Mon, 08 January 2018 12:14
I think I mention this at least once a year, but Microsoft SDK conventions are an ever increasing clusterfuck...

Yup.

VS 2017 Community was not detected...

Subject: Re: U++ 2017.2 released
Posted by cbpporter on Tue, 09 Jan 2018 10:52:54 GMT
View Forum Message <> Reply to Message

I managed to get stuff working with VS2015, by moving over some existing .bm files and editing them to whatever paths the setup decided to use. VS 2015 was not detected.

This time the include and libs are under "10.0.16299.0".

When U++ was upgraded to C++xx whatever, it did not work out of the box for months. Then it got

better. Now there is regression again.

Plus, I have no BLITZ, even though it looks like it is enabled.

----- Core (MSC15 DEBUG DEBUG_FULL MSC WIN32) (1 / 5)

Subject: Re: U++ 2017.2 released
Posted by cbpporter on Tue, 09 Jan 2018 11:02:44 GMT
View Forum Message <> Reply to Message

And weirdly enough, the removal of:

Vector<Ln>   line;

from TextCtrl is affecting me.

That is fixable, I just need to update and merge for inherited controls.

The BM detection needs to be fixed though.

I'm willing to boot up a few VMs and test a few clean setups.

Subject: Re: U++ 2017.2 released
Posted by mirek on Tue, 09 Jan 2018 11:38:46 GMT
View Forum Message <> Reply to Message

cbpporter wrote on Tue, 09 January 2018 12:02And weirdly enough, the removal of:

Vector<Ln>   line;

from TextCtrl is affecting me.

That is fixable, I just need to update and merge for inherited controls.

The BM detection needs to be fixed though.

I'm willing to boot up a few VMs and test a few clean setups.

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

Subject: Re: U++ 2017.2 released
Posted by cbpporter on Tue, 09 Jan 2018 12:06:36 GMT

> 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.

OK, my method ain't that good.

It picks up and incomplete/missing SDK MSC12 as a complete one. Missing "LIBCMT.lib". I guess such a file can be added somewhere at the end of the search to make sure a build is running.

Anyway, Microsoft don't really care about older versions and a lot of information has been purged from the sites, including downloads and any way to make heads or tails from the confusing VS vs. Windows SDK versions.

So I'm preparing a VM right now with a 100% legal Windows 10.

Could you please provide some semblance of what the installation steps should be. There used to be some instructions on the site but right now I can't find them. Probably because auto-detection doesn't work.

Then I will replicate the steps and show auto-detection results.

Two first choices should be VS 2015 and VS 2017 community edition, right?

Subject: Re: U++ 2017.2 released
Posted by cbpporter on Tue, 09 Jan 2018 20:57:48 GMT

How about these settings?

Because this is not auto-detected.

File Attachments
1) vs2017.png, downloaded 714 times

Subject: Re: U++ 2017.2 released
Posted by mirek on Wed, 10 Jan 2018 08:04:28 GMT

cbpporter wrote on Tue, 09 January 2018 20:07So I'm preparing a VM right now with a 100% legal Windows 10.

Could you please provide some semblance of what the installation steps should be. There used to be some instructions on the site but right now I can't find them. Probably because auto-detection doesn't work.

https://www.ultimatepp.org/appideinstall$en-us.html

Quote:
Then I will replicate the steps and show auto-detection results.

Two first choices should be VS 2015 and VS 2017 community edition, right?

No reason to bother with VS2015 (but let us leave the code in the place). First choice should be build tools, secondary choice Community edition, ternary all regular variants of VS. All 2017.

Subject: Re: U++ 2017.2 released
Posted by cbpporter on Wed, 10 Jan 2018 09:25:55 GMT

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.

Subject: Re: U++ 2017.2 released
Posted by mirek on Wed, 10 Jan 2018 10:29:41 GMT

cbpporter wrote on Wed, 10 January 2018 10:25It 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?

Yes. There are some utils required to build.

Quote:

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.

Then it is simple. The support for versions is there, used for include / lib. See lines bellow. So it is just that I have not anticipated bin being in versioned folder (only).

Quote:
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.

I guess it is MS constantly changing everything...

Subject: Re: U++ 2017.2 released
Posted by cbpporter on Wed, 10 Jan 2018 21:38:49 GMT

mirek wrote on Wed, 10 January 2018 12:29
I guess it is MS constantly changing everything...

I guess MS were as short sighted as usual and did not anticipate several versions of the same SDK (all 10). This time they didn't go with 7 vs 7a or 8 and 8.1 scheme.

Anyway, 2017 does indeed use versioning. Tested on the clean VM. I'll try to fix it as with the libs and test it for a while and provide the fix.

Sorry this is taking a while, but my clean VM PC is the opposite corner of the city from my work PCs where I have every single VS version in existence.

Subject: Re: U++ 2017.2 released
Posted by mirek on Wed, 10 Jan 2018 22:11:56 GMT
View Forum Message <> Reply to Message

cbpporter wrote on Wed, 10 January 2018 22:38mirek wrote on Wed, 10 January 2018 12:29
I guess it is MS constantly changing everything...

I guess MS were as short sighted as usual and did not anticipate several versions of the same SDK (all 10). This time they didn't go with 7 vs 7a or 8 and 8.1 scheme.

Anyway, 2017 does indeed use versioning. Tested on the clean VM. I'll try to fix it as with the libs and test it for a while and provide the fix.

Sorry this is taking a while, but my clean VM PC is the opposite corner of the city from my work PCs where I have every single VS version in existence.

I have added versioning of bin, hopefully fixing your setup. Please check (now in trunk, tomorrow nightly).

Subject: Re: U++ 2017.2 released
Posted by cbpporter on Thu, 11 Jan 2018 08:46:52 GMT
View Forum Message <> Reply to Message

So fast!

I'll check!

Do you have idea on the other problem I mentioned? I can't debug even command line apps with "target file override".

Subject: Re: U++ 2017.2 released
Posted by mirek on Thu, 11 Jan 2018 09:04:39 GMT
View Forum Message <> Reply to Message

cbpporter wrote on Thu, 11 January 2018 09:46So fast!

I'll check!

Do you have idea on the other problem I mentioned? I can't debug even command line apps with "target file override".

Unable to reproduce (so far). "works on my machine" :(

Any further input on this is appreciated.

Subject: Re: U++ 2017.2 released
Posted by cbpporter on Thu, 11 Jan 2018 09:16:45 GMT
View Forum Message <> Reply to Message

mirek wrote on Thu, 11 January 2018 11:04cbpporter wrote on Thu, 11 January 2018 09:46So fast!

I'll check!

Do you have idea on the other problem I mentioned? I can't debug even command line apps with "target file override".

Unable to reproduce (so far). "works on my machine" :(

Any further input on this is appreciated.
Yeah, it is tricky. My theory was that it is related to the resource compiler, but today I can't debug compile line either.

But I managed to reproduce it with EyeCare. AdressBook does not reproduce.

Set a target override for EyeCare, press F5, wait to link, it should finish, but not launch debugger. Pres F5 again, it will attempt to link again.

Subject: Re: U++ 2017.2 released
Posted by cbpporter on Fri, 12 Jan 2018 09:52:28 GMT
View Forum Message <> Reply to Message

For now I had to revert versions.

I tried 2017.1 and it works, but it is way too ancient. It misses several fixes I need, 2 of them done by me. So I can't use it.

I am now on the 2017.2 RC, it has all the fixes I need and it works I think. Need to test more.

So from RC to release the debugging + target override got broken.

Narrowing it down further...

Subject: Re: U++ 2017.2 released

Posted by [mirek](#) on Fri, 12 Jan 2018 10:10:19 GMT
[View Forum Message](#) <> [Reply to Message](#)

cbpporter wrote on Fri, 12 January 2018 10:52For now I had to revert versions.

I tried 2017.1 and it works, but it is way too ancient. It misses several fixes I need, 2 of them done by me. So I can't use it.

I am now on the 2017.2 RC, it has all the fixes I need and it works I think. Need to test more.

So from RC to release the debugging + target override got broken.

Narrowing it down further...

What about nightly build or trunk?

One possible explanation: 32-bit ide cannot debug 64-bit application. Any chance you are running ide32 and building 64-bit?

Mirek

Subject: Re: U++ 2017.2 released
Posted by [cbpporter](#) on Mon, 15 Jan 2018 09:37:55 GMT
[View Forum Message](#) <> [Reply to Message](#)

My tests were with a self-compiled TheIDE, but I was using a 64bit to debug 32, like I usually do. Updating to nightly didn't help.

But I figured that my upp folder was compromised and indeed, after a fresh install, it works. As an added bonus BLITZ works again. It didn't on my broken setup. Autodection works too, at least for 2017.

So everything is a go. Thank you!

Which reminds me: it is probably time to stop using 32 bit for good.

How do you manage both 32 and 64 bit binary releases? Not for U++, but in general. Two packages? One package with both 32 bit and 64 binaries, with different names? Is there some lightweight .bat or .exe or something to autodetect and launch the proper version?

Subject: Re: U++ 2017.2 released
Posted by [mirek](#) on Mon, 15 Jan 2018 10:24:54 GMT
[View Forum Message](#) <> [Reply to Message](#)

cbpporter wrote on Mon, 15 January 2018 10:37My tests were with a self-compiled TheIDE, but I was using a 64bit to debug 32, like I usually do. Updating to nightly didn't help.

But I figured that my upp folder was compromised and indeed, after a fresh install, it works. As an added bonus BLITZ works again. It didn't on my broken setup. Autodection works too, at least for 2017.

So everything is a go. Thank you!

Which reminds me: it is probably time to stop using 32 bit for good.

How do you manage both 32 and 64 bit binary releases? Not for U++, but in general. Two packages? One package with both 32 bit and 64 binaries, with different names? Is there some lightweight .bat or .exe or something to autodetect and launch the proper version?

Uhm, I have tree cases I can comment on:

- U++ itself. I provide theide32.exe as backup option.
- My commercial app1: intended for many users with sometimes low-spec HW, app is not performance limited: We provide 32-bit only .exe
- My commercial app1: intended for many users with usually hi-spec HW, app is very performance limited: We provide 64-bit as standard, 32-bit on demand (which is very low).

Subject: Re: U++ 2017.2 released
Posted by mirek on Mon, 15 Jan 2018 10:26:28 GMT
View Forum Message <> Reply to Message

cbpporter wrote on Mon, 15 January 2018 10:37
But I figured that my upp folder was compromised and indeed, after a fresh install, it works. As an added bonus BLITZ works again. It didn't on my broken setup. Autodection works too, at least for 2017.

OK, so nightly build install now works out of box? (Asking because thinking about doing the next release soon).

Mirek

Subject: Re: U++ 2017.2 released
Posted by Zbych on Mon, 15 Jan 2018 10:50:44 GMT
View Forum Message <> Reply to Message

mirek wrote on Mon, 15 January 2018 11:26(Asking because thinking about doing the next release soon)

Do you have time to look at this ide crash before next release?

Subject: Re: U++ 2017.2 released
Posted by cbpporter on Mon, 15 Jan 2018 15:15:19 GMT

View Forum Message <> Reply to Message

mirek wrote on Mon, 15 January 2018 12:26cbpporter wrote on Mon, 15 January 2018 10:37
But I figured that my upp folder was compromised and indeed, after a fresh install, it works. As an added bonus BLITZ works again. It didn't on my broken setup. Autodection works too, at least for 2017.

OK, so nightly build install now works out of box? (Asking because thinking about doing the next release soon).

Mirek

So far so good.

Stable and auto-detection works for me.

I've been forking CodeEditor for years now and I see some convergent parallel evolution. I tried in the past to merge my efforts, but it was a total failure every time. This time I'd like to do a super small merge and would like to see that incorporated. Mostly bar placements bugs and binary literals, a dozen or so line changed.

Plus, there is a small minor bug with a DropDown still triggering and tool-tip from an underlying control.

Plus I need to get around to saving the size of the run options dialog.

But otherwise a new release will be welcome and I am not aware of critical bugs.