Subject: What next?
Posted by mirek on Thu, 17 Dec 2015 14:54:41 GMT
View Forum Message <> Reply to Message

OK, so U++ is now better than ever, but still largely ignored.

Which is sort of understandable, because nobody knows and most of those that know do not understand (rest are happy U++ users).

As I do not see the point in developing complex framework/ide for tens of users, I would like to concentrate a bit more on PR in the next period.

Things are not that simple, number of public arenas open to "subvertising" of open-source software is now limited. E.g. annoucement post on google groups is now read by about 10 people total. Readmeat does not exist anymore.

So I am trying to figure out what to do next.

- I believe I have to write a document about U++ Core and its performance. I guess this one of early detractors from U++ and it should be properly explained why we need (and everybody should) avoid C++ standard library.

- The only suitable medium for PR I have found is CodeProject, where you can still submit articles about more or less anything. It is not so easy, because articles can be rejected. Anyway, I would start with version of reference/GuiWebCrawler.

- Maybe release some parts of U++ as separate libraries. E.g. memory allocator is a good candidate IMO.

- Maybe create core conventional derivative code editor from theide

Any other ideas or comments?

Subject: Re: What next?
Posted by Tom1 on Thu, 17 Dec 2015 16:25:45 GMT
View Forum Message <> Reply to Message

Hi,

I wish I was better on marketing -- for a number of reasons. Maybe a short survey of what made us (tens of happy U++ users) make the switch to U++ could highlight the key selling points of U++, and then emphasize those points in marketing efforts.

Here's my story: I had worked years with Borland, then Watcom and finally Microsoft C++ tools. I was continuously looking (some heavy googling) for a solution to get rid of my dependence on Microsoft and their policies, (which by the way have not improved over the years, IMO). I wanted

to find a suitably licensed (BSD) cross platform development solution that covered Windows, Linux and maybe *BSD. Finally 2006 I found U++ and slowly started to learn and rewrite my first application on U++. Step by step I became convinced of the superiority of U++ and started to realize how much more it was than what I had even dreamed of.

So, the initial key selling points for me were the BSD license and the cross platform development capability.

Another thing that comes to mind are the programming courses in universities, which could be a great advertising platform. A small project on U++ might give the spark. This would obviously require support from the professors.

Best regards,

Tom

Subject: Re: What next?
Posted by mr_ped on Thu, 17 Dec 2015 16:56:41 GMT
View Forum Message <> Reply to Message

Maybe... (just thinking loud) one possibility may be to pick up some popular unmaintained open source project, and write alternative replacement in U++ from scratch. But I have no idea about what project that could be, maybe some front end cross platform client for some open messaging API, like jabber/gtalk, including video calls, etc? That may have impact over larger user base, but looks like lot of work just to get it going, and without heavy maintenance it would probably die soon after.

And with most ideas it would need serious involvement of some developers, the most probable outcome is that there will not enough people to develop and maintain it.

But in case somebody has some great idea about some SW, which can have serious impact and maybe is eventually willing to do development, write it here.

I personally hardly manage my own projects, lately I don't any programming outside of work, so this is just an idea from me, nothing more. :/

Subject: Re: What next?
Posted by mirek on Thu, 17 Dec 2015 17:16:15 GMT
View Forum Message <> Reply to Message

mr_ped wrote on Thu, 17 December 2015 17:56Maybe... (just thinking loud) one possibility may be to pick up some popular unmaintained open source project, and write alternative replacement in U++ from scratch. But I have no idea about what project that could be, maybe some front end cross platform client for some open messaging API, like jabber/gtalk, including video calls, etc?

That may have impact over larger user base, but looks like lot of work just to get it going, and without heavy maintenance it would probably die soon after.

Well, I would say that would end exactly the same - good but ignored. With PR, it could end better, but if we knew how to do PR then, we can do the same right now, right?

Subject: Re: What next?
Posted by Klugier on Thu, 17 Dec 2015 17:16:33 GMT
View Forum Message <> Reply to Message

Hello,

- First of all, one of the easiest channel to let user know about Ultimate++ is wikipedia. Currently

& Polski). A lot of this article is outdated.or a little bit too small. So my proposition is to create articles for other languages (community support is required) and update/upgrade current.Even small articles especially in Chinless, Indian and Japanese (huge countries with a lot of software potential).

- One of the problem is the heterogeneity of the brand. We have got too many brands that is hard to remember. We have Ultimate++, U++, TheIDE etc.. Especially the last brand (TheIDE) may cause misunderstandings. Maybe it should be something else that shows it is related to Ultimate++ like U++IDE Ultimate++ IDE etc..

- In 2016 Ultimate++ should participate in Google Summer of Code. Conclusions regarding past failures must be drawn. Google Summer of Code is good opportunity to attract new and perspective (young) developers.

- Backing to university, Android Builder (introduced in 2015.2 version) was my master thesis (So it is definitely a possibility for the future - if you want to write thesis within Ultimate++ you can do this)

Sincerely,
Klugier

Subject: Re: What next?
Posted by mirek on Thu, 17 Dec 2015 17:53:33 GMT
View Forum Message <> Reply to Message

mr_ped wrote on Thu, 17 December 2015 17:56Maybe... (just thinking loud) one possibility may be to pick up some popular unmaintained open source project, and write alternative replacement in U++ from scratch. But I have no idea about what project that could be, maybe some front end

cross platform client for some open messaging API, like jabber/gtalk, including video calls, etc? That may have impact over larger user base, but looks like lot of work just to get it going, and without heavy maintenance it would probably die soon after.

Funny, just stumbled upon this:

https://sourceforge.net/p/keepass/discussion/329220/thread/1 bd4b047/?limit=25

(look at weekly downloads...)

Subject: Re: What next?
Posted by mirek on Thu, 17 Dec 2015 18:07:29 GMT
View Forum Message <> Reply to Message

Klugier wrote on Thu, 17 December 2015 18:16Hello,

- First of all, one of the easiest channel to let user know about Ultimate++ is wikipedia. Currently

& Polski). A lot of this article is outdated.or a little bit too small. So my proposition is to create articles for other languages (community support is required) and update/upgrade current.Even small articles especially in Chinless, Indian and Japanese (huge countries with a lot of software potential).

I welcome anybody doing that. However, it is considered "bad" to use Wikipedia to promote own ideas, so I should not meddle with the article.

BTW, a couple of days ago, I have stumbled upon a very nice proposal for wikipedia article but some "wiki intern" or something. What a shame it was not published... (unfortunately, I fail to find it again).

Quote:
- One of the problem is the heterogeneity of the brand. We have got too many brands that is hard to remember. We have Ultimate++, U++, TheIDE etc.. Especially the last brand (TheIDE) may cause misunderstandings. Maybe it should be something else that shows it is related to Ultimate++ like U++IDE Ultimate++ IDE etc..

Actually, the name was likely a one of mistakes. It sounds like ranting. That is why later I have rather tried to promote U++.

In fact, maybe we should change the name.

Quote:
- In 2016 Ultimate++ should participate in Google Summer of Code. Conclusions regarding past failures must be drawn. Google Summer of Code is good opportunity to attract new and

perspective (young) developers.

Maybe. But it would be just pointless PR stunt with little impact (who reads all these applications outside Google?). And Google will not choose us, because we are unknown.

Mirek

Subject: Re: What next?
Posted by dolik.rce on Thu, 17 Dec 2015 20:08:55 GMT
View Forum Message <> Reply to Message

mirek wrote on Thu, 17 December 2015 15:54- Maybe release some parts of U++ as separate libraries. E.g. memory allocator is a good candidate IMO.
That is actually very good idea. I had a similar one few months ago, but never got time to actually try it. There is actually quite a lot of projects that use jemalloc or similar preloadable libraries to squeeze out some more performance out of their code. I believe "umalloc" could offer similar performance boost and it might help spread the word about the rest of U++ too.

Additionally, I think it might be worth to spend some time making U++ less dependent on TheIDE. Even though it is a great editor, many programmers are not willing to switch from their favorite IDE/editor (which they spent years learning and customizing) just to try a new framework. And using U++ without TheIDE is currently non trivial.

I remember some experiments with converting upp files to Visual Studio projects, waf based builds and of course the universal makefile, but frankly, all of those are quite difficult to use without previous knowledge of U++. Making a library that could be just linked against would help a lot, but I know this has already been tried several times before, with mixed results.

Best regards,
Honza

Subject: Re: What next?
Posted by Novo on Fri, 18 Dec 2015 03:53:08 GMT
View Forum Message <> Reply to Message

IMHO, it would be a good idea to present about U++ at Meeting C++, CppCon, or CppNow. These events are very popular and a lot of people watch videos from these events and read slides. If you are presenting you do not need to pay fee. If Mirek cannot go there himself maybe somebody else can make a presentation about U++. meetingcpp.com has a list of C++ groups in different countries.

Subject: Re: What next?

Posted by Novo on Fri, 18 Dec 2015 04:19:09 GMT
View Forum Message <> Reply to Message

Two main competitors of U++ are Qt and wxWidgets. Both support Mac and can be built with all popular build systems.

Another thing. It is almost impossible to find any reference to U++ anywhere on Internet. It is just not listed among GUI and C++ frameworks.

Subject: Re: What next?
Posted by mr_ped on Fri, 18 Dec 2015 08:49:05 GMT
View Forum Message <> Reply to Message

mirek wrote on Thu, 17 December 2015 18:53Funny, just stumbled upon this:

https://sourceforge.net/p/keepass/discussion/329220/thread/1 bd4b047/?limit=25

Interesting... now I recall I was recently (1m back?) googling a bit different terms around U++ (out of curiosity, to see the impact), already almost forgot about it.

The main line is, that it is extremely difficult to find anything... the term "ultimate" is so common, that if I would be first time TheIDE user, somehow ignorant of official web presence (like maybe just trying it out from repository, and not paying attention to "Help/..."), I would be unable to find anything.

Anyway, digging deeper into results I found few articles even outside this website, one of them is skype plugin, which went from ultimate++ in v0.20 (and using U++ only for linux version probably) to wxWidgets in v1.00:
http://clownfish-translator.com/linux.html

--
The thing about CppCon and similar sounds good to me, I sometimes check the videos too (especially Stroustrup's)

Subject: Re: What next?
Posted by mirek on Fri, 18 Dec 2015 09:59:45 GMT
View Forum Message <> Reply to Message

Novo wrote on Fri, 18 December 2015 05:19Two main competitors of U++ are Qt and wxWidgets. Both support Mac and can be built with all popular build systems.
There is also a new generation of GUI frameworks rising rapidly.

Another thing. It is almost impossible to find any reference to U++ anywhere on Internet. It is just not listed among GUI and C++ frameworks.

Hm, I thought this one is quite covered. Where is it missing?

Mirek

Subject: Re: What next?
Posted by mirek on Fri, 18 Dec 2015 10:01:50 GMT
View Forum Message <> Reply to Message

dolik.rce wrote on Thu, 17 December 2015 21:08
I remember some experiments with converting upp files to Visual Studio projects, waf based builds and of course the universal makefile, but frankly, all of those are quite difficult to use without previous knowledge of U++. Making a library that could be just linked against would help a lot, but I know this has already been tried several times before, with mixed results.

Well, I am afraid this is still mostly the question of INITBLOCK.

Should we drop INITBLOCK and require manual registration of modules used?

Mirek

Subject: Re: What next?
Posted by Novo on Sat, 19 Dec 2015 03:37:42 GMT
View Forum Message <> Reply to Message

mirek wrote on Fri, 18 December 2015 04:59

Hm, I thought this one is quite covered. Where is it missing?

Mirek

Sorry, I'm wrong. U++ is listed on wikipedia. The rest is not important

---

Subject: Re: What next?
Posted by deep on Sat, 19 Dec 2015 13:23:54 GMT
View Forum Message <> Reply to Message

One of the idea to increase visibility of U++

I think can we create projects / apps using U++ on Github / Sourceforge.

When people see useful app on Source forge developed using U++ it will increase visibility of the project.

There are many projects with wxWidgets on source forge

Subject: Re: What next?
Posted by cbpporter on Sat, 19 Dec 2015 13:40:02 GMT
View Forum Message <> Reply to Message

I saw this thread when it was created, but it took me a few days to gather my thoughts on the subject. Doubly so, since I'm almost in the same boat, but probably where U++ was 10+ years ago: I too am developing a solution for some perceived problems taking the form of a library (no GUI though), an endeavor mostly fueled by dissatisfaction with current popular solutions. I am still in the development phase, but I'm dreading the day I will actually have to promote the project.

Because promotion is very hard, especially in this domain. Really really hard. So much harder than writing code , which is easy and fun and has deterministic outcome. And especially now, when C++ is no longer as common as it used to be. Everybody is using JS now and we can't even hire competent new people at my workplace. The "if you build it, they will come" approach does not work. Consistently outputting a quality product does not attract people. It may be great at retaining people, but the first exposure most come from somewhere else.

U++ is a great tool with great productivity and generally a good design, but is is also often weird, quirky and some parts of it are downright alien. If you ask me, U++ is equally well designed and well designed to drive off random enthusiasts who just heard of the project and want to give it a quick try. The barrier of entry is too high.

I think that this is mostly due to TheIDE and a lack of structure that puts some of the best parts of U++ to the forefront.

U++ has great value as a GUI framework. But GUI work is relatively rare in this day and age. But it also has huge amounts of value outside of GUI. I love CodeEditor as an example, because it saved me from ritting thousand of lines of code, but you won't attract new people with CodeEditor or ArrayCtrl. You'll do it with Core at first on the non GUI side and with simple GUI samples that show of accessibility. GUI can be complicated and downright threatening looking in some toolkits, but creating a TopWindow (which should be called Window anyway) and plopping down a few Buttons can be inviting.

But I still think Core needs special attention. Core is super valuable but quite inaccessible.

Here is an example: I don't think that there is a seasoned C++ developer out there who really loves std::vector. Most people that I know hate it and iterators with pointer syntax in general. U++ has Vector, which is a lot more pleasant to use and potentially faster. I do not understand why after so many years, a random person can't just pick a random C++ source file, add a #include <upp/Vector> or something, and happily go on and use Vector. No mess, no fuss, no dependencies, no pulling in full packages, no random allocator, no almost proprietary IDEs. Or String. Or VectorMap. People do need hash maps all the time. std::map is so lacking. Or CParser. CParser is pure gold! I don't like pointless hyperbole, but CParser did change a small part of my life for the better! It is ridiculously convenient to have this power in a few lines of code without

having to use some lexer lib and grammar. In a language notorious for having zero parsing ability without blowing up some pointers and barely being able to convert an int to a string. Or Socket. Very useful!

The contents of U++ should be filtered and a "core core" should be made very accessible. Not everything that is in the packages though. Just the super useful high quality stuff. Color is nice and I guess it does the job, but it is not a selling point. My Color lib is 40k code of code without conversion functions and I still find it lacking, so a simple Color class is not an attraction like the containers, serialization, Xml support, Unicode and so on.

Now I'm not saying that we no longer use packages or TheIDE. I'm saying the message for new initiates should be "U++ Core is awesome because these classes are great. You can use TheIDE for a modular approach at developing programs, but first let me show you how awesome Vector is using the Linux command line/CMake/Visual Studio. You'll love it and never touch std::vector again! And BTW, Vector is compatible with std, so you can still mix it with other code!".

Neither am I saying we break up Core into 2 separately maintained code bases. There is probably a way to reorganize sources so that stand alone classes can be included with standard C++ paradigms (some thought must be put into how to break up libs for non templates, probably the boost model) one by one and still have Core.h which resides in a U++ package and provides the entire functionality.

Subject: Re: What next?
Posted by Sgifan on Sat, 19 Dec 2015 23:31:53 GMT
View Forum Message <> Reply to Message

Communicate about the good things about UPP:

Productivity gains are huge for a C++ developer. The large scope of the framework really helps also in this area.
Being able to use a unique framework for serial communication, databases handling, charts and reports for instance
has allowed me to gain a great deal of time.

Try to enhance some not so good points:

TheIDE is great but could try to be a bit more simple concerning navigation (finding references, definitions, usages, etc...) (I quite like NetBeans these days in this area).
I would like to see theIDE be capable of having multiple targets defined inside a project, and one of these target could be a unit tests.
Also having unit tests directly supported in theIDE gui.

The web site is a bit old fashioned and I found Github so nice I really think there is good inspiration to take from there.

Having a more dynamic web site to be able to use it from a mobile device would be great.

The advertising on the site is also a bit strange (I get for instance a lot of dating advertising on the left side of the screen)

Subject: Re: What next?
Posted by Mindtraveller on Sun, 20 Dec 2015 09:00:35 GMT
View Forum Message <> Reply to Message

I guess it is time we (U++ users) should help Mirek.

1. First of all, we must update comparison on U++ site adding more comparisons showing more cases of U++ usage vs Qt/wxWidgets. Comparisons must be 'honest' even if U++ shows little/no advantage in the particular scenario.

2. Second, we must start writing 'U++ red book' showing methodology and techniques of different aspects of U++ programming. Especially those which are very different from widely used ones. This is something very different from U++ help. It is more a teaching book than a reference.
I believe this is something U++ needs a lot. U++ is very hard to learn mostly because the most of knowledge is scattered among this forum and is hard to find.

3. Each U++ user should write 1-2-3 or more articles on different interesting aspects of U++, preferably one in it's own language and the second one in English. Topics include: new U++ features (for the recent year), Rainbow/Mac/Android compatibility, Core containers, etc.

This is really something we (users) may do without passing it on to Mirek.

Subject: Re: What next?
Posted by Mindtraveller on Sun, 20 Dec 2015 09:08:23 GMT
View Forum Message <> Reply to Message

Klugier wrote on Thu, 17 December 2015 21:16- One of the problem is the heterogeneity of the brand. We have got too many brands that is hard to remember. We have Ultimate++, U++, TheIDE etc.. Especially the last brand (TheIDE) may cause misunderstandings. Maybe it should be something else that shows it is related to Ultimate++ like U++IDE Ultimate++ IDE etc..

I agree. My proposal is that we concentrate on 'U++'. U++ is a framework. 'U++ ide' for the ide.
I doubt if calling 'Ultimate' is a good approach but it IMO it is completely up to U++ authors to decide.
Frankly speaking I guess it is not very important at all.

Subject: Re: What next?
Posted by mirek on Sun, 20 Dec 2015 09:46:10 GMT
View Forum Message <> Reply to Message

Thanks everybody for posting ideas. I will try to summarize now, and also try to estimate 'costs'

PR and Docs

- try to publish articles on codeproject or else. Cost: very low
- explain Core, compare with standard library. Cost: medium
- write Red Book. Might be the same as previous. Cost: medium
- fix names/brands. Cost: low to medium. Moderate risk involved.
- update wikipedia article. Cost: low. Low risk involved
- update website. Cost: high
- more comparisons, benchmarks. Cost: low

Public events/people

- universities. Cost: unknown.
- C++ meetings. Cost: low (local) to high (intercontinental)

Programming "ex" U++

* create libraries based on U++
  - just allocator. Cost: low
  - Core subset. Cost: medium
  - whole packages. Cost: high
- promote U++ by creating public projects in it: Cost: very high

Programming U++ (this, however, is sort of "business as usual"

- support MacOS: Cost: very high, but it should definitely happen
- support Android: Cost: very high, but it should definitely happen
* improve theide
  - support more targets. Cost: medium
  - further improve Assist++ (to support references etc.). Cost: very high (but most fun

Subject: Re: What next?
Posted by mirek on Sun, 20 Dec 2015 09:59:48 GMT
View Forum Message <> Reply to Message

Mindtraveller wrote on Sun, 20 December 2015 10:00I guess it is time we (U++ users) should help Mirek.

1. First of all, we must update comparison on U++ site adding more comparisons showing more cases of U++ usage vs Qt/wxWidgets. Comparisons must be 'honest' even if U++ shows little/no advantage in the particular scenario.

2. Second, we must start writing 'U++ red book' showing methodology and techniques of different aspects of U++ programming. Especially those which are very different from widely used ones. This is something very different from U++ help. It is more a teaching book than a reference.
I believe this is something U++ needs a lot. U++ is very hard to learn mostly because the most of knowledge is scattered among this forum and is hard to find.
[/code]

Well, the idea always was that 'reference' contains all that knowledge in form of simple examples...

Mean

a) teaching book might take 'reference' as sort of outline for 'red book'
b) it would be worth to somehow gather all issues not covered there

[code]
3. Each U++ user should write 1-2-3 or more articles on different interesting aspects of U++, preferably one in it's own language and the second one in English. Topics include: new U++ features (for the recent year), Rainbow/Mac/Android compatibility, Core containers, etc.

This is really something we (users) may do without passing it on to Mirek.

It would be great. I am not so sure about other languages than English, it proved rather hard to maintain.

Anyway, for the best effect, I would propose publishing articles elsewhere first (codeproject?). It is simple to move it to our website later.

I believe that something like that would have the most impact of all.

The source of inspiration can easily be examples/reference...

Subject: Re: What next?
Posted by Mindtraveller on Sun, 20 Dec 2015 10:21:29 GMT
View Forum Message <> Reply to Message

One more thing we should possibly do is to analyze user's opinions on U++.

1. Search Internet for articles/reviews on U++. Then we make, say, google document on all these reviews containing links found.
2. Extract PROs and CONs from actual reviews we gathered.
3. Analyze CONs and PROs and make forum thread on possible ways to remove CONs and tell

about PROs to more users.

And one more thing: the links on articles written should also be collected per user and in one place (say, google doc). So that we see some statistics.

Subject: Re: What next?
Posted by mr_ped on Sun, 20 Dec 2015 20:29:49 GMT
View Forum Message <> Reply to Message

Mindtraveller wrote on Sun, 20 December 2015 11:21One more thing we should possibly do is to analyze user's opinions on U++.

1. Search Internet for articles/reviews on U++. Then we make, say, google document on all these reviews containing links found.
2. Extract PROs and CONs from actual reviews we gathered.
3. Analyze CONs and PROs and make forum thread on possible ways to remove CONs and tell about PROs to more users.

And one more thing: the links on articles written should also be collected per user and in one place (say, google doc). So that we see some statistics.

I tried to do "1." few months back, and I didn't found anything really interesting. I think there were like 1-3 blog-like short reviews. Maybe I didn't dig hard enough into search results, but I think it's more like the whole framework is almost invisible to the world (as it happens with SW ... like PHP Nette framework, top quality stuff, yet hardly used outside of CZ).

But if I ever saw some PROs, it usually followed my own list: lean and small codebase (and small download), download was easy to install and compile first example (now even easier with "portable" release).

After that CONs show usually fast: the learning curve kicks in (obviously, it's C++, even my own old source looks alien to me), "mandatory" usage of TheIde, and while U++ is sort of condensed and delivering a lot over small space - it's not as wide as f.e. Qt, lacking thing here and there.

I'm now probably too long here to asses the *real* impact of those things. For me personally the biggest annoyance is debugger (and I rather learned to not need it most of the time), and to include 3rd party libraries in TheIde at source level (I maintain one project with Tesseract OCR lib, included as sources... takes me some 1-2h to update the files and project, as it's not simple "compile everything what's on disc"), and missing refactoring tools (none). Overall I'm quite happy with U++, unfortunately most of the work time I have to develop in other languages and IDEs (android studio :/).

Anyway, I think we should try to get some stories from newcomers, they will be probably quite different, and maybe unexpectedly complicated on things we don't see as any problem.

Subject: Re: What next?
Posted by cbpporter on Mon, 21 Dec 2015 17:07:17 GMT
View Forum Message <> Reply to Message

For some reason some people perceive calling something "ultimate" as tacky and it is also not easy to google, so we should grep the site and replace Ultimate++ with U++?

Subject: Re: What next?
Posted by mirek on Mon, 21 Dec 2015 19:55:33 GMT
View Forum Message <> Reply to Message

cbpporter wrote on Mon, 21 December 2015 18:07For some reason some people perceive calling something "ultimate" as tacky and it is also not easy to google, so we should grep the site and replace Ultimate++ with U++?

Actually, I was using just U++ for years, for exactly this reason.

Anyway, the trouble is that U++ is hard to google and also there are (AFAIK) two other open-source libraries called U++.

Also, there is a problem with www domain. (www.uplusplus.com ? weird)

Thus I am open to complete name change.

Mirek

Subject: Re: What next?
Posted by Sgifan on Mon, 21 Dec 2015 22:31:53 GMT
View Forum Message <> Reply to Message

Concerning naming I find UPP as a name not so bad and really better than ultimate something...

On the question of the web site, please look at juce.com who is another niche C++ framework, they just modernized their web site

Subject: Re: What next?
Posted by Sgifan on Mon, 21 Dec 2015 22:35:39 GMT
View Forum Message <> Reply to Message

.. and once there is an android version of UPP, it would be cool to have raywenderlich.com doing a tutorial.
They have great visibility for a developer oriented site and following their tutorials is very interesting in general.

Subject: Re: What next?
Posted by cbpporter on Mon, 21 Dec 2015 22:39:57 GMT
View Forum Message <> Reply to Message

I did a tour of the entire site and I think it is pretty good. It could certainly do with some updated content and screenshots, especially the "Overview" tab. And it could do with some modern styling, but I wouldn't turn it into such a modern design with fullscreen images of trustworthy people (OK, maybe we can have one with Mirek) and laptops.

Subject: Re: What next?
Posted by Sgifan on Mon, 21 Dec 2015 22:50:31 GMT
View Forum Message <> Reply to Message

the "features" and the "tutorials" do not have full screen background images and are nonetheless nice and well layed out. And i do not mention the "live coding" in the ide that impressed me already 2 years ago when Julian did the demo (https://www.youtube.com/watch?v=imkVkRg-geI)

Subject: Re: What next?
Posted by cbpporter on Mon, 21 Dec 2015 22:57:55 GMT
View Forum Message <> Reply to Message

Sgifan wrote on Tue, 22 December 2015 00:50the "feattures" and the "tutorials" do not have full screen background images and are nonetheless nice and well layed out
Yeah, like I said, could do with a bit of styling. A nice selection of fonts and colors on top of Bootstrap 3 style-sheet, with some carousels and JS enabled site elements.

A few hard hitting samples using a pleasant style might be effective: one that shoes how short GUI code is (but people won't understand template layouts and Bar), one that shows the subpixel features of Painter how it works in GUI but as console code too and so on...

Subject: Re: What next?
Posted by Sgifan on Tue, 22 Dec 2015 22:46:39 GMT
View Forum Message <> Reply to Message

How would this be done as the UPP website is generated from uppweb and most of the content is qtf formated ?

Subject: Re: What next?
Posted by koldo on Wed, 23 Dec 2015 06:30:44 GMT
View Forum Message <> Reply to Message

Sgifan wrote on Tue, 22 December 2015 23:46How would this be done as the UPP website is generated from uppweb and most of the content is qtf formated ?If approved uppweb may have some abilities to change some formating items, like fonts.

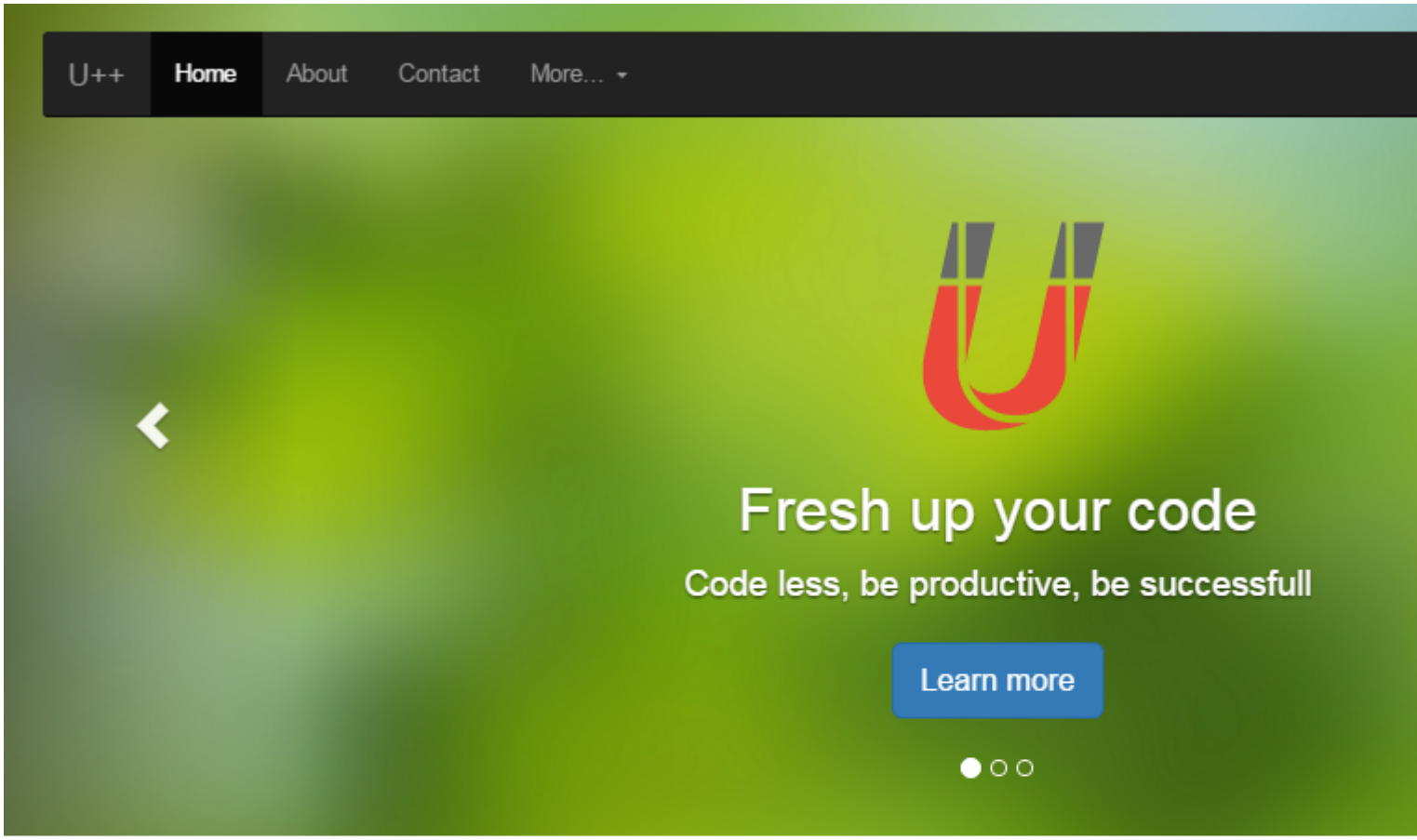
Subject: Re: What next?
Posted by Sgifan on Wed, 23 Dec 2015 14:10:44 GMT
View Forum Message <> Reply to Message

Web designer could easily come up with something nice

but datas are in .tpp files. that are not so easily accessible...

File Attachments

1) sample.png, downloaded 1696 times





Less Is more

Ultimate++ promises radical reduction of code complexity of typical desktop applications. The IDE compiles you code blazingly fast

View details »



Full featured

A full IDE is available for the framework. The framework itself provides a full set of services for typical desktop applications like, database access, report generations, full featured gui

View details »

Subject: Re: What next?
Posted by mirek on Fri, 01 Jan 2016 10:31:58 GMT
View Forum Message <> Reply to Message

As result of this debate I am now working or planning to work on following things:

- CodeProject articles. First one awaiting approval
http://www.codeproject.com/Articles/1068378/Simple-multi-req uest-web-crawler-using-Uplusplus
- Converting allocator to C to get it ready to be publishied (of course, in the process I have found a nice way how to optimize it
- Creating "U++ Core explained" document
- Probably with colaboration with others, U++ Core as library

---

Subject: Re: What next?
Posted by mdelfede on Sun, 21 Feb 2016 16:21:38 GMT
View Forum Message <> Reply to Message

Hi,

I'm cooperating with an Italian electronic magazine, which is (among others) very focused on IOT / embedded development.
We're pushing an arduino-compatible board with some success, and we're planning to provide many android apps to control it.
So, when UPP will be ready for android development, I guess I can write a series of articles about UPP on that magazine,
along with some developed Androids apps.
The magazine is on-paper in Italy, has on-paper associates in Spain and in France, and has an on-line companion in english, so the visibility should be good.

Ciao

Max

Subject: Re: What next?
Posted by nlneilson on Wed, 04 May 2016 22:53:25 GMT
View Forum Message <> Reply to Message

mirek wrote on Thu, 17 December 2015 06:54OK, so U++ is now better than ever, but still largely ignored.

So I am trying to figure out what to do next.

- I believe I have to write a document about U++ Core and its performance. I guess this one of early detractors from U++ and it should be properly explained why we need (and everybody should) avoid C++ standard library.

Any other ideas or comments?

U++ is probably 'better than ever' but could be 'more better'

"what to do next-write a document" YES 'u++ API'
API (application program interface) is a set of routines, protocols, and tools for building software applications. The API specifies how software components should interact and APIs are used when programming graphical user interface (GUI) components.

Even in win from the command line just entering 'help' will give the 'API' for the dos functions.

"why we need (and everybody should) avoid C++ standard library."

OK, Why should we avoid the C++ standard library ??

The PR to get someone to try u++ is one thing, getting them to stay is something else. Make it easier to get basic answers. Have the README that is now included in the download but blank include basic questions/answers short code examples and a brief explanation of what the function does, in alphabetical order like exit - ends the app or function.

u++ is very good but after not using it or doing any programming for a few years it can be very frustrating just like many new users can be frustrated.

Neil

Subject: Re: What next?
Posted by cbpporter on Mon, 16 May 2016 09:28:53 GMT
View Forum Message <> Reply to Message

Here is another potential obstacle I found for new users.

As I said on another thread, I'm updating my VS to the latest and giving up on the venerable VS2010.

So I googled VS express, installed what I got from the first link. BTW, it installed VS 2015 in a folder called VS 14. Microsoft have been horrible at naming these versions. I don't know why it is so hard for them to name things in a consistent non-confusing way.

I don't know where it should go, in "instant" or "legacy" setup, but I clicked instant and it took forever. This new directory search method is bad, compared to the old registry one.

Anyway, it eventually finished I think and I have no new build method, just the old MSC10 and MSC11.

I'd be surprised if a new user doesn't give up at this point. I'm not a new user but I certainly have given up for now. I'll be using 11 until I update my own detection code to detect this version too. It started from TheIDE detection code and it stops at MSC12.

Subject: Re: What next?
Posted by koldo on Tue, 17 May 2016 07:11:03 GMT
View Forum Message <> Reply to Message

Quote:I don't know where it should go, in "instant" or "legacy" setup, but I clicked instant and it took foreverYes, I would call it "Automatic" instead of "Instant".

Yesterday I installed a new MSC15 and finally I set the directories by hand, that was easy but not evident as thanks to Microsoft, in different computers the MSC15 directories are different

Subject: Re: What next?
Posted by mirek on Thu, 19 May 2016 09:14:07 GMT
View Forum Message <> Reply to Message

koldo wrote on Tue, 17 May 2016 09:11Quote:I don't know where it should go, in "instant" or "legacy" setup, but I clicked instant and it took foreverYes, I would call it "Automatic" instead of "Instant".

Yesterday I installed a new MSC15 and finally I set the directories by hand, that was easy but not evident as thanks to Microsoft, in different computers the MSC15 directories are different

ok, looks like they have updated some things...

Currently, U++ is trying to search for dirs based on names. I guess they have changed.

Could you give me overview of dirs in Program Files? Like

cd /Program Files
dir /AD /S

Mirek

Subject: Re: What next?
Posted by omari on Thu, 19 May 2016 15:17:31 GMT
View Forum Message <> Reply to Message

Hi,

I think, we can find all MSCxx installation paths, by enumeration of registry subkeys and values.

please, test the function below, and let us know if it success or fail to detect your.

on my win7 64, it show:

SDK found:

C:\Program Files (x86)\Windows Kits\10
C:\Program Files (x86)\Microsoft SDKs\Windows\v7.1A
C:\Program Files (x86)\Windows Kits\8.0
C:\Program Files (x86)\Microsoft SDKs\Windows\v8.0A
C:\Program Files (x86)\Windows Kits\8.1
C:\Program Files (x86)\Microsoft SDKs\Windows\v8.1A
Visual Studio found:
C:\Program Files (x86)\Microsoft Visual Studio 12.0\
C:\Program Files (x86)\Microsoft Visual Studio 12.0\
C:\Program Files (x86)\Microsoft Visual Studio 14.0\

the code:

```
#include <Core/Core.h>
using namespace Upp;

Vector<String> GetWinRegSubKeys(String reg_path, HKEY base_key =
HKEY_LOCAL_MACHINE)
{
 HKEY key = 0;
 DWORD  dwSize = 0;
 DWORD count; // subkeys

 Vector<String> v;

 if(RegOpenKeyEx(base_key, reg_path, 0, KEY_READ, &key) != ERROR_SUCCESS)
  return v;
 StringBuffer buf(2048);

 RegQueryInfoKey(key,NULL,NULL,NULL,&count,NULL,NULL,NULL,NULL,NULL,NULL,NULL);

    for(int i=0; i< count; i++){
        dwSize = 2048;
        RegEnumKeyEx(key,i,buf,&dwSize,NULL,NULL,NULL,NULL);
        v.Add( String(buf.Begin(), dwSize)) ;
    }

    RegCloseKey(key);
    return v;
}

Vector<String> GetWinRegValues(String reg_path, HKEY base_key = HKEY_LOCAL_MACHINE)
{
 HKEY key = 0;
 DWORD  dwSize = 0;
```

```
 DWORD count; // subkeys

 Vector<String> v;

 if(RegOpenKeyEx(base_key, reg_path, 0, KEY_READ, &key) != ERROR_SUCCESS)
  return v;
 StringBuffer buf(2048);

 RegQueryInfoKey(key,NULL,NULL,NULL,NULL,NULL,NULL, &count,NULL,NULL,NULL,NULL);

    for(int i=0; i< count; i++){
        dwSize = 2048;
  RegEnumValue(key, i, buf, &dwSize, NULL, NULL, NULL, NULL);
        v.Add( String(buf.Begin(), dwSize)) ;
    }

    RegCloseKey(key);
    return v;
}

String NormalizePathNN(const String& path)
{
 return IsNull(path) ? path : NormalizePath(path);
}

void FindMSC()
{
 Vector<String> v = GetWinRegSubKeys("SOFTWARE\\Microsoft\\Microsoft SDKs\\Windows",
HKEY_LOCAL_MACHINE);

 if(v.GetCount())
  LOG("SDK found:");
 else
  LOG("no SDK found");
 for(int i = 0; i < v.GetCount(); i++)
 {
  LOG(NormalizePathNN(GetWinRegString("InstallationFolder",
                                       "SOFTWARE\\Microsoft\\Microsoft SDKs\\Windows\\" + v[i],
                                       HKEY_LOCAL_MACHINE)));
 }

 v = GetWinRegValues("SOFTWARE\\Microsoft\\VisualStudio\\SxS\\VS7",
HKEY_LOCAL_MACHINE);

 if(v.GetCount())
  LOG("Visual Studio found:");
 else
  LOG("No Visual Studio found");
```

```
 for(int i = 0; i < v.GetCount(); i++)
 {
  LOG(GetWinRegString(v[i], "SOFTWARE\\Microsoft\\VisualStudio\\SxS\\VS7",
HKEY_LOCAL_MACHINE));
 }
}

CONSOLE_APP_MAIN
{
 FindMSC();
}
```

Subject: Re: What next?
Posted by koldo on Thu, 19 May 2016 15:20:29 GMT
View Forum Message <> Reply to Message

Hello Mirek

Here there are the files for two different computers with Windows 7. All files are under
- c:\Program Files (x86)\Microsoft Visual Studio 14.0
- c:\program files (x86)\windows kits

File Attachments
1) files.7z, downloaded 482 times

Subject: Re: What next?
Posted by cbpporter on Thu, 19 May 2016 15:30:02 GMT
View Forum Message <> Reply to Message

koldo wrote on Thu, 19 May 2016 18:20Hello Mirek

Here there are the files for two different computers with Windows 7. All files are under
- c:\Program Files (x86)\Microsoft Visual Studio 14.0
- c:\program files (x86)\windows kits
Those are the exact folders I'm trying to detect too. Under Windows Kits, we need the subfolder 10, not 8.1 or 8.0.

The only problem I'm facing is that the "lib" and "include" folder from "windows kits" has several subfolders, with different versions of the SDK. I have both 10.0.10150.0 and 10.0.10240.0. Inside is the ucrt folder. I'm thinking of sorting these subfolders and using the latest one? And I'm searching for a registry setting that could tell me which one to sue, but so far no luck.

Subject: Re: What next?
Posted by koldo on Fri, 20 May 2016 06:34:09 GMT
View Forum Message <> Reply to Message

Quote:
Those are the exact folders I'm trying to detect too. Under Windows Kits, we need the subfolder 10, not 8.1 or 8.0.

The only problem I'm facing is that the "lib" and "include" folder from "windows kits" has several subfolders, with different versions of the SDK. I have both 10.0.10150.0 and 10.0.10240.0. Inside is the ucrt folder. I'm thinking of sorting these subfolders and using the latest one? And I'm searching for a registry setting that could tell me which one to sue, but so far no luck.
I am doing just that, choosing the latest folder at every option.

Subject: Re: What next?
Posted by mirek on Sat, 21 May 2016 17:57:03 GMT
View Forum Message <> Reply to Message

koldo wrote on Thu, 19 May 2016 17:20Hello Mirek

Here there are the files for two different computers with Windows 7. All files are under
- c:\Program Files (x86)\Microsoft Visual Studio 14.0
- c:\program files (x86)\windows kits

At first glance I do not see anywhing wrong

Would it be possible to put go to InstantSetup.cpp 164

```
vc = df.ScanForDir("/vc", "", "bin/link.exe;bin/cl.exe;bin/mspdb140.dll", "bin/1033");
bin = df.ScanForDir(x64 ? "bin/x64" : "bin/x86", "/windows kits/", "makecat.exe;accevent.exe", "");
inc = df.ScanForDir("", "/windows kits/", "um/adhoc.h", "um;ucrt;shared");
lib = df.ScanForDir("", "/windows kits/", "um/x86/kernel32.lib", "um;ucrt");
```

and add

```
DUMP(vc);
DUMP(bin);
DUMP(inc);
DUMP(lib);
```

?

Thanks!

Subject: Re: What next?
Posted by cbpporter on Tue, 24 May 2016 09:23:34 GMT
View Forum Message <> Reply to Message

mirek wrote on Sat, 21 May 2016 20:57

DUMP(vc);
DUMP(bin);
DUMP(inc);
DUMP(lib);

Quote:
vc = c:/program files (x86)/microsoft visual studio 14.0/vc
bin = c:/program files (x86)/windows kits/8.1/bin/x86
inc =
lib =

VS is from MSC 14 but the SDK from 12? And inc and lib empty.

Subject: Re: What next?
Posted by koldo on Wed, 01 Jun 2016 14:28:00 GMT
View Forum Message <> Reply to Message

Sorry Mirek, I did not understood you.

- First computer
vc = c:/program files (x86)/microsoft visual studio 14.0/vc
bin = c:/program files (x86)/windows kits/10/bin/x86
inc = c:/program files (x86)/windows kits/10/include/10.0.10240.0
lib = c:/program files (x86)/windows kits/10/lib/10.0.10240.0

vc = c:/program files (x86)/microsoft visual studio 14.0/vc
bin = c:/program files (x86)/windows kits/10/bin/x64
inc = c:/program files (x86)/windows kits/10/include/10.0.10240.0
lib = c:/program files (x86)/windows kits/10/lib/10.0.10240.0

- Second computer
vc = c:/program files (x86)/microsoft visual studio 14.0/vc
bin = c:/program files (x86)/windows kits/8.1/bin/x86
inc =
lib =

vc = c:/program files (x86)/microsoft visual studio 14.0/vc
bin = c:/program files (x86)/windows kits/8.1/bin/x64
inc =
lib =

Subject: Re: What next?
Posted by mirek on Thu, 02 Jun 2016 09:32:02 GMT
View Forum Message <> Reply to Message

koldo wrote on Wed, 01 June 2016 16:28Sorry Mirek, I did not understood you.

- First computer
vc = c:/program files (x86)/microsoft visual studio 14.0/vc
bin = c:/program files (x86)/windows kits/10/bin/x86
inc = c:/program files (x86)/windows kits/10/include/10.0.10240.0
lib = c:/program files (x86)/windows kits/10/lib/10.0.10240.0

vc = c:/program files (x86)/microsoft visual studio 14.0/vc
bin = c:/program files (x86)/windows kits/10/bin/x64
inc = c:/program files (x86)/windows kits/10/include/10.0.10240.0
lib = c:/program files (x86)/windows kits/10/lib/10.0.10240.0

- Second computer
vc = c:/program files (x86)/microsoft visual studio 14.0/vc
bin = c:/program files (x86)/windows kits/8.1/bin/x86
inc =
lib =

vc = c:/program files (x86)/microsoft visual studio 14.0/vc
bin = c:/program files (x86)/windows kits/8.1/bin/x64
inc =
lib =

I am a little bit confused by "first" and "second" computer now... It looks like on "first" computer, Instant setup worked OK?

Mirek

Subject: Re: What next?
Posted by koldo on Fri, 03 Jun 2016 06:32:18 GMT
View Forum Message <> Reply to Message

Quote:It looks like on "first" computer, Instant setup worked OK?I cannot confirm you as I do not know how TheIDE works.
Really the working directories in this computer are:

MSC15

- BIN
c:/program files (x86)/microsoft visual studio 14.0/vc/bin
c:/program files (x86)/windows kits/10/bin/x86

- INCLUDE
c:/program files (x86)/microsoft visual studio 14.0/vc/include
c:/program files (x86)/windows kits/10/include/10.0.10240.0/um
c:/program files (x86)/windows kits/10/include/10.0.10240.0/ucrt
c:/program files (x86)/windows kits/10/include/10.0.10240.0/shared

- LIB
c:/program files (x86)/microsoft visual studio 14.0/vc/lib
c:/program files (x86)/windows kits/10/lib/10.0.10240.0/ucrt/x86
c:/program files (x86)/windows kits/10/lib/10.0.10240.0/um/x86

Subject: Re: What next?
Posted by cbpporter on Fri, 24 Jun 2016 09:03:35 GMT
View Forum Message <> Reply to Message

Could somebody with a working VS 2015/MSC14 please go to the appropriate build method in "Setup/Build methods" and list the paths that are included in the list boxes at the bottom of the tabs: "Path -executable directories", "INCLUDE directories" and "LIB directories".

I can't get anything to compile. Things like "windef.h" and "user32.lib" are no where to be found.

Thank you!

Subject: Re: What next?
Posted by cbpporter on Mon, 27 Jun 2016 11:01:46 GMT
View Forum Message <> Reply to Message

Whatever version of the SDK is linked on the homepage will try to install SDK version 6.0?

That can't possibly be right...

Subject: Re: What next?
Posted by cbpporter on Mon, 27 Jun 2016 15:59:10 GMT

Finally, after about a month or how much time, I can finally have a VS2015 capable setup.

Based on this, I also updated my own auto-detect code, a small lib that detects compiler versions and spits out data that can also be stored as an XML.

I attached such a sample. Anyone interested in the code? It is very experimental.

File Attachments

1) buildMethods.xml, downloaded 546 times