
Subject: Collaborative U++ Projects

Posted by [captainc](#) on Fri, 19 Sep 2008 14:14:28 GMT

[View Forum Message](#) <> [Reply to Message](#)

I think we need to get a few Open Source applications made with U++ out there. And I think U++ developers should collaborate on some projects together (good development collaboration practice!). I have 2 ideas for applications to work on:

1. Hierarchically Structured Digital Notebook

- Think of it as a digital version of an x-subject notebook. Top level sections, each with sub-sections. We can use Tab control and/or tree-view. The notebook can have full capabilities of UWord.
- T++ integration by being able to link to other notebook pages/sections (wiki-style).
- Optional features: note synchronization, math calculation integration, password protection / encrypted notes.

2. E-mail client

- POP, IMAP, and SMTP support
- Filters
- Address Book
- RSS Feeds (with our new handy RssCtrl!)
- Calendar -> maybe separate project (Since Thunderbird development has halted, why not create something in its place?)

Any other ideas?

The beauty of U++ is that these application could be portable and cross-platform. It won't have many dependencies. We could post on PortableApps.com and all that.

Let's create some teams, work together, and make 1 or 2 applications that really show off U++'s power! Who's in?

Subject: Re: Collaborative U++ Projects

Posted by [cbpporter](#) on Fri, 19 Sep 2008 15:06:40 GMT

[View Forum Message](#) <> [Reply to Message](#)

Sure, getting some open-source projects out here could help popularity.

But I see some problems with that:

1. Who should do that? I don't think there are a lot of people (using U++ or not) who actually have time to sit down and just start coding on some random project. I for one only code new projects in three distinct cases: it brings me money, it is really fun (happens rarely, but it does) or I need a small tool for something that the normal tools can't quite hack (and in such cases I'm usually done in some days; I have a lot of small apps that do data processing mostly or organize stuff, and for some strange reason I never wrote one in a language more suited for such tasks). And I think

something similar to this applies to all working U++ developers.

2. What exactly to code. It should be useful, yet should not reinvent the wheel. I think that hundreds of programs that do the same stuff, and also forking have in general a quite negative effect on software as a whole, and especially on open-source. Please install Linux, and install all the calculators you can find in your distro. Then install all the Paint like programs, then the text editors. And the list goes on and on. If all the people who ever contributed to a text editor would have pulled their effort, we would have 2-3 extraordinary editors (which hopefully would not have turned out a monstrosity like emacs) and not 10 GUI ones installed almost by default, out of which 5 are from Qt, 3 Gtk+ and 2 are just plain X.

If someone wants to write a new text editor for example, having a revolutionary design or at least one that caters for special needs of a subgroup of people and which is feature-wise quite different from the ones that exist, than that someone just might have a good reason to do that. If instead the reason is that some toolkit is better than the rest (in reality or just perceived), and there is not a program to solve the same task written in that toolkit, that that someone's reason is not that good and IMO will just harm open source by causing more redundancy.

3. Rewriting something in U++ would give the best results when that certain something is written in C. Unfortunately, most software out there is written in plain C, and most of the time not even in a civilized subset of it. Rewriting something like that would show off our capabilities quite well and lead to a better piece of software. But rewriting something which uses STL, Qt, Gtkmm or wxWidgets would not bring any significant benefit. Sure, you could probably pinpoint some code sections, but in the end, the whole effort would not be too justified.

I had the pleasure of working a little with two open-source C code bases, which I'm not going to name. One of them is one of the most horrible pieces of C software in regard to its design and the way C is used to make it unreadable (but its functionality is exceptionally good, making it required software). This project, quite used and probably installed on your favorite distro would in principle benefit hugely from a rewrite, and would get at least 80% of that benefit from using a string class. Any string class, but something smart like String would reap the best rewards. The second one is very well written, and I don't think that using anything, not even the absolutely perfect language and library that don't exist would have a major benefit on it, neither on functionality, nor on code cleanliness. Sure, being C, there are tons of stuff you could clean up, like using modules and namespaces, getting rid of macros, a little const correctness, etc., but this could be done in about a week or two and the function implementations would be virtually unaltered.

Subject: Re: Collaborative U++ Projects

Posted by [captainc](#) on Fri, 19 Sep 2008 16:01:08 GMT

[View Forum Message](#) <> [Reply to Message](#)

Quote: it is really fun (happens rarely, but it does)

That's basically the crowd I was aiming at. I enjoy collaborating on software projects... helps me learn too.

Quote:

If all the people who ever contributed to a text editor would have pulled their effort, we would have 2-3 extraordinary editors (which hopefully would not have turned out a monstrosity like emacs) and not 10 GUI ones installed almost by default,

I agree, but the main differences in many of the programs that have similar functionality is the programming language itself.

For example, TheIDE, it is another IDE for C++. Aren't there soo many of these? Why do we use it instead? Why not make a plugin for Eclipse that can auto-complete U++ and do what the hot keys do?

In my eyes, the main distinction with U++ is the development philosophy. I think it calls for a set of applications that cohere to its philosophy. Moreover, it is meant for cross platform compatibility. When you take into account the number of applications that are cross-platform, there are not too many of them.

I fully agree that the apps should be different and not reinvent the wheel.

Subject: Re: Collaborative U++ Projects

Posted by [cbpporter](#) on Fri, 19 Sep 2008 16:45:25 GMT

[View Forum Message](#) <> [Reply to Message](#)

Well, actually I would love to try something if I had time. It is more like a research project.

It would be something like RPython, i.e. a subset of a dynamic language that is strictly compilable/runnable under the host language, yet it is compilable to machine code. Proponents claim that you still get most of the productivity boost that such languages are said to offer, yet you get a performance boost between 20 and 200 times.

I would follow these steps:

1. Grab the sources and factor out any ugly stuff if present.
2. Rewrite it using U++ types.
3. Retarget it to llvm.
4. Do the actual productivity case study, probably by making the compiler host itself.

I'm also sure we would also get a bonus compilation time improvement between 5 and 10 times at least without any extra work, when compared to C.

Too bad that I don't have time .

Subject: Re: Collaborative U++ Projects

Posted by [Oblivion](#) on Fri, 19 Sep 2008 18:11:56 GMT

[View Forum Message](#) <> [Reply to Message](#)

Quote:

2. E-mail client

- POP, IMAP, and SMTP support

- Filters

- Address Book

- RSS Feeds (with our new handy RssCtrl!)

- Calendar -> maybe separate project (Since Thunderbird development has halted, why not create something in its place?)

Well, I have a good news then I have almost finished a POP3Mail class. (It conforms to the RFC 1939, though no APOP or TLS, or MIME parsing yet). I was considering to upload it to the Bazaar this weekend with a simple mail reader example. Maybe someone could help me with a MIME parser and a Standard "Mail" class, before I start my own.

Subject: Re: Collaborative U++ Projects
Posted by [bytefield](#) on Fri, 19 Sep 2008 18:41:04 GMT
[View Forum Message](#) <> [Reply to Message](#)

I'm in if i have time and mostly I'm doing this for learning purpose and for fun . I don't know if you accept me as a part of your team because of:

my U++ knowledge is not so advanced(know that from my post on forum and contributions to Upp);
i don't know C++ at a high level (i'm still learning it);
english isn't my mother tongue and sometimes i cannot couch what i think at(i'm still using dictionary for some words);

I was thinking to start an open-source project with Upp and i was thinking at a software for doing Debian like repositories and i will do it with or without any contributions from other Upp programmers(if they don't like my idea or simply don't want to be part of project) but now and for next month i simply don't have time to get in it, I'm begin second year at university and have other project to finish before get involved in another.

Subject: Re: Collaborative U++ Projects
Posted by [Mindtraveller](#) on Fri, 19 Sep 2008 20:46:57 GMT
[View Forum Message](#) <> [Reply to Message](#)

UltiMail++ client is the most useful of the list IMO. With some additions:

- Lightweight (no monstrous requirements, no heavy dependencies)
- Easy to use interface
- Should be able to run from USB stick
- Multiple proxy profiles inside (maybe with autoselecting proxy needed by computer IP mask on the start)
- Auto-update should be no harder than Mozilla products do
- IMAP support

This would be really handy thing.

My participation is too early to discuss. First of all I'll try to contribute to U++ manuals.

Subject: Re: Collaborative U++ Projects
Posted by [Mindtraveller](#) on Fri, 19 Sep 2008 20:49:04 GMT
[View Forum Message](#) <> [Reply to Message](#)

Add about mail client:
- Connection/fetching status interface.

Subject: Re: Collaborative U++ Projects
Posted by [captainc](#) on Fri, 19 Sep 2008 22:33:33 GMT
[View Forum Message](#) <> [Reply to Message](#)

Quote:Well, I have a good news then Smile I have almost finished a POP3Mail class. (It conforms to the RFC 1939, though no APOP or TLS, or MIME parsing yet). I was considering to upload it to the Bazaar this weekend with a simple mail reader example. Maybe someone could help me with a MIME parser and a Standard "Mail" class, before I start my own. Wink
That's just awesome; can't wait to check it out. We'll just need to make an IMAP class now too.

Subject: Re: Collaborative U++ Projects
Posted by [forlano](#) on Sat, 20 Sep 2008 08:37:12 GMT
[View Forum Message](#) <> [Reply to Message](#)

Oblivion wrote on Fri, 19 September 2008 20:11Quote:

2. E-mail client

- POP, IMAP, and SMTP support
- Filters
- Address Book
- RSS Feeds (with our new handy RssCtrl!)
- Calendar -> maybe separate project (Since Thunderbird development has halted, why not create something in its place?)

Well, I have a good news then I have almost finished a POP3Mail class. (It conforms to the RFC 1939, though no APOP or TLS, or MIME parsing yet). I was considering to upload it to the Bazaar this weekend with a simple mail reader example. Maybe someone could help me with a MIME parser and a Standard "Mail" class, before I start my own.

Hello,

I can be a beta tester. It would be very nice to have something able to run from a usb drive and able to synchronize your emails and Address Book.

I think that the first 3 steps

- POP, IMAP, and SMTP support
- Filters
- Address Book

would be a very good starting point and let the horizon to be quite visible and not very far.

Luigi

Subject: Re: Collaborative U++ Projects
Posted by [Oblivion](#) on Sat, 20 Sep 2008 09:07:08 GMT
[View Forum Message](#) <> [Reply to Message](#)

forlano wrote on Sat, 20 September 2008 11:37Oblivion wrote on Fri, 19 September 2008 20:11Quote:

2. E-mail client

- POP, IMAP, and SMTP support
- Filters
- Address Book
- RSS Feeds (with our new handy RssCtrl!)
- Calendar -> maybe separate project (Since Thunderbird development has halted, why not create something in its place?)

Well, I have a good news then I have almost finished a POP3Mail class. (It conforms to the RFC 1939, though no APOP or TLS, or MIME parsing yet). I was considering to upload it to the Bazaar this weekend with a simple mail reader example. Maybe someone could help me with a MIME parser and a Standard "Mail" class, before I start my own.

Hello,

I can be a beta tester. It would be very nice to have something able to run from a usb drive and able to synchronize your emails and Address Book.

I think that the first 3 steps

- POP, IMAP, and SMTP support
- Filters
- Address Book

would be a very good starting point and let the horizon to be quite visible and not very far.

Luigi

I agree, it would be nice to have that feature. As for the "horizon" I'm rather optimistic this time. As I mentioned above, I've almost finished the barebone Pop3 class. What's more is, This is a derivative of my "MailSocket" base class which can serve as a clear and familiar interface for handling the other Mail protocols. Actually, I "borrowed" code from Smtplib (I hope no one would mind) , modified it and created a common interface for deriving other mail protocols easily.
