Subject: My explaination of why Ultimate++ is not mainstream
Posted by amrein on Thu, 14 Aug 2008 15:46:25 GMT
View Forum Message <> Reply to Message

Here are two references on-line explaining most of the U++ raison for failure in main stream. My 3 last comments explain my mains issues and certainly the ones of the FOSS community too.

Let's start with issues documentation:

1. You can read this article on-line just to know how does FOSS works: Building an Open Source Community (by Phil Whitehouse).

https://fossbazaar.org/?q=content/building-open-source-commu nity

2. Also, Producing Open Source Software (by Karl Fogel) is the complete book on-line and a very interesting work (he sells this book in paper form too):

http://producingoss.com/en/

Now, my main issues:

3. The current Licence is not GPL compatible. GPL is compatible with the new BSD licence, not a derivative of the old one. No one will be able to add GPL code in U++ to release something better, nor a program using the GPL licence can link dynamically or not. with U++.

http://www.gnu.org/licenses/license-list.html#GPLIncompatibl eLicenses

4. The U++ old BSD like licence is even not OSI approved !!! (OSI = Open Source Initiative). So, U++ is not open source at all.

http://www.opensource.org/licenses/alphabetical

Conclusion, U++ is not FOSS. It's for all those companies releasing proprietary application. Those companies don't create community, they feed them. Why FOSS people would want to use U++ then?

5. And last, the website has a few issues in design. The newcomers flow could really be improved as people adhesion to the U++ community.

It's my main work at present to do website design, enterprise internal restructure to get back to profits + good marketing image. U++ developers are better then me in C++ coding. Here is why I can't do what they have done and why I try to help.

Just one thing: I will understand it you just say F... Y...
But, really, I wouldn't have done all this work and net search for free if I didn't think U++ and its developers didn't need it and deserve it.

Regards
Amrein-Marie Christophe

Subject: Re: My explaination of why Ultimate++ is not mainstream
Posted by mirek on Thu, 14 Aug 2008 15:58:41 GMT
View Forum Message <> Reply to Message

amrein wrote on Thu, 14 August 2008 11:46
It's for all those companies releasing proprietary application.

Actually, yes. Well, to be more specific, it is for programmers of such companies

Anyway, I agree about the license issue. You know, when we were about to supply it, I just scanned a couple of FOSS projects and picked what I liked - that is files from E17. They told "BSD licensed" in description of E17 that is why I supposed this is BSD. Then read some other BSD - they require the same with different wording.

Reading "the real" BSD license, the only thing I really care about is clause 1 (do not remove copyrights from sources, well, uhm, really, do not remove copyright files from directories). Is there any license like that?

Thinking about it, maybe we can just go PD?

Quote:
Just one thing: I will understand it you just say F... Y...

Fresh look of newcomers is always good.

Anyway, U++ has some mostly historical issues that need to be fixed first. E.g. that "svn" you are using is not the "main repository". That is why all that talk about "infrastructure server".

Maybe you could look around this forum, many things were discussed in the past...

Quote:
But, really, I wouldn't have done all this work and net search for free if I didn't think U++ and its developers didn't need it and deserve it.

No problem. You are welcome, there is a lot of work you can help with. (OTOH, you know that... "talk is cheap, show me the code" (Note: U++ website is U++ generated, that is why changing its structure is changing the code too

Mirek

Subject: Re: My explaination of why Ultimate++ is not mainstream
Posted by captainc on Thu, 14 Aug 2008 16:20:31 GMT
View Forum Message <> Reply to Message

Quote:Here are two references on-line explaining most of the U++ raison for failure in main stream
Firstly, I don't think there has been failure here.
Secondly, that statement assumes that it was trying to be mainstream in the first place. To my knowledge, there hasn't been any significant effort to market U++. Marketing is huge and U++ may not be ready for mainstream. This all may be on purpose, or it may not be a primary objective at this time. These are questions for guys like Mirek to answer, and we may need to have an open discussion on it. All in all, I think U++ has grown well simply by word of mouth. And the core of the open source U++ effort has been active with the participation of the small community, constant contact with core developers, and frequent releases. I for one, check the forums multiple times per day, and I know the developers do too, as they are very quick to respond and post updates (Thanks to Mirek).

At the end of the day, it would be good to know exactly where the core development team stands as to the direction they want the project to go. If they have long-term plans for this project, and these plans are stated and known, it may ease the reluctance of potential developers to begin/continue working with U++
.

Subject: Re: My explaination of why Ultimate++ is not mainstream
Posted by Zardos on Thu, 14 Aug 2008 17:27:23 GMT
View Forum Message <> Reply to Message

luzr wrote on Thu, 14 August 2008 17:58
Reading "the real" BSD license, the only thing I really care about is clause 1 (do not remove copyrights from sources, well, uhm, really, do not remove copyright files from directories). Is there any license like that?

luzr wrote on Thu, 14 August 2008 17:58
Thinking about it, maybe we can just go PD?

What about beerware

```
/*
 * ----------------------------------------------------------------------------
 * "THE BEER-WARE LICENSE" (Revision 42):
```

* <phk@FreeBSD.ORG> wrote this file. As long as you retain this notice you
* can do whatever you want with this stuff. If we meet some day, and you think
* this stuff is worth it, you can buy me a beer in return Poul-Henning Kamp
* --
*/

http://en.wikipedia.org/wiki/Beerware

Subject: Re: My explaination of why Ultimate++ is not mainstream
Posted by amrein on Thu, 14 Aug 2008 19:20:15 GMT
View Forum Message <> Reply to Message

captainc wrote on Thu, 14 August 2008 18:20Quote:Here are two references on-line explaining most of the U++ raison for failure in main stream
Firstly, I don't think there has been failure here.
Secondly, that statement assumes that it was trying to be mainstream in the first place. To my knowledge, there hasn't been any significant effort to market U++.
...

My referenced post:

http://www.ultimatepp.org/forum/index.php?t=msg&th=94&am p;start=0&
http://www.ultimatepp.org/forum/index.php?t=msg&th=301&a mp;start=0&

It's not a failure for proprietary software companies at all.
There won't be any FOSS community around it, that's all.
FOSS is a a different way to do business. FOSS is commercial. Companies have no problem using FOSS. U++ could have a huge help from the FOSS community and fail to have a bigger one because things are going differently. FOSS is everywhere. Even in my cellphone and my aDSL modem.

luzr wrote on Thu, 14 August 2008 17:58amrein wrote on Thu, 14 August 2008 11:46
It's for all those companies releasing proprietary application.

Actually, yes. Well, to be more specific, it is for programmers of such companies

Anyway, I agree about the license issue. You know, when we were about to supply it, I just scanned a couple of FOSS projects and picked what I liked - that is files from E17. They told "BSD licensed" in description of E17 that is why I supposed this is BSD. Then read some other BSD - they require the same with different wording.

Reading "the real" BSD license, the only thing I really care about is clause 1 (do not remove

copyrights from sources, well, uhm, really, do not remove copyright files from directories). Is there any license like that?

Thinking about it, maybe we can just go PD?

So here is how things goes at present for U++ (all this is approximation of course):

- Source available and shareable
- Everyone can improve it or link with it but
- No one can redistribute the modified source or their own applications without your copyright in-build
- Fork is possible but anyone forking U++ will still need to write your copyright in each source file you have written and in their about menu
(+) you would like also people to not remove anything from your tarball to be sure that when someone see your copyright, they know it's yours.

Here is how things goes in FOSS:

- Source available and shareable
- Everyone can improve it or link with it as long as they follow the licence but
- No one can redistribute the modified source or their own applications with your copyright. If their modification or software are trash software, it's not your fault.
- Fork is possible, but anyone forking U++ must follow the licence in each source file (and show the licence in their about menu if you use GNU licence for example)
- If they fork and distribute the modified software, you can read/copy/share/modify/get back those modifications into your own software and continue to grow your way

Here are the addition for the 2 main GNU licence:
i
(+) GNU GPL : anything linking with it must be GPL licensed
(+) GNU LGPL : anything linking dynamically with it can use whatever licence, but if (software + U++) == (One .exe) then the software must be LGPL licensed
(+) GNU LGPL + exception (like the WxWindows licence) is not official from GNU but a short explanation could be: as long as you provide the modifications of the software we give you and that your are distributing now, we don't care about your own source code and your licence. Just let people know you have build your software with it and gave them back our software with ypur modifications of our source files.

With your old BSD like licence, there is no share possible between you and any other source code with the same BSD like licence with different copyright name. There is no share possible with other LGPLed libraries. No share possible between FOSS developers and you. Just a community of programmers from proprietary companies. A shrinking target.
To be short, U++ is talking to proprietary developers. Open source talk to proprietary developers as to all other developers. When I mean all other, a short list could be:

GOOGLE, YAHOO, IBM, INTEL, AMD, VIA, NOVELL, HP, DELL, ASUS, MSI, MOTOROLA, NEC, NTT DOCOMO, PANASONIC, SAMSUNG, NOKIA, VODAFONE, REDHAT, UBUNTU/CANONICAL, MANDRIVA, all other that works around your database (mysql, postgresql, sqlite...), your webpage (apache, php, python, ...), ... even governments. And they are so much more with all those anonymous contributors.

So, when I read: "How can we get bigger community around U++", I don't need to think twice.

Note: The most near licence from FOSS, when comparing to yours, is the GNU LGPL + exception (like the WxWindows licence), but It doesn't have the old BSD viral copyright.

Will that really change your business? Yes! People will always be able to do the same thing with your source code but won't be able to di it without giving the same right to downloaders. Afraid about project forking? Get back their modifications into the main line. You don't like what they have done? Don't care, the best will survive. It's not you? You can still copy and adapt. Even proprietary companies will like to use your software. No change for them. Big changes for you.

Subject: Re: My explaination of why Ultimate++ is not mainstream
Posted by cbpporter on Thu, 14 Aug 2008 19:37:39 GMT
View Forum Message <> Reply to Message

I don't see a failure here either. I'm not aware of any publicity/marketing campaign, or even anything comparable to it, so I can't say we failed. Maybe Mirek is promoting it locally or something.

Sure, we are only a few that use it. Sure, we don't have a Wikipedia entry. Getting more users would definitely be a great. It would help extend U++ and make it even more general and easy to use.

The main advantages of U++ IMO are:

1. It's small, fast, powerful and reasonably crossplatfrom
It is exactly what you need if you want a lean and mean environment in which to work. You get everything out of the box, no need to use makefiles, set up paths, install runtimes.

And even if it is small, it is powerful and fast. It fits well with low processing power and memory requirements. You get to write short code, which does a lot and you have powerful and appropriate widgets. U++ does have some useful stuff for writing console code and containers are better than STL, but GUI code is where U++ strength lies.

With it's lack of popularity, U++ still manages to fill a niece and it was exactly what I needed when I was searching for a new C based toolkit that could replace and keep up with Delphi. IMO, the only other toolkit that can keep up with U++ is Qt. Qt is even better, but it is pretty huge, almost rivaling something which is "managed", like a .NET runtime. Deploying is also harder with Qt, and Windows support was not that great, but I like it in 4.3+. Some might be put of about the licensing cost also.

2. It is smart and clean C++ code
Often people seeing U++ code are surprised how clean it is. For some sections of code, you could swear it is Java code. It is also smart code and uses a lot advanced C++ features. I'm sure it is not even compilable on older and less complete C++ compilers. It also keeps your knowledge fresh. In U++ I found myself writing template code for the first time out of need and because it would simplify stuff, rather than out of necessity to interact with some code. It makes templates fun and useful.

3. It is not under GPL license.
FOSS people might not like this, but this is a huge advantage for me. I'm more pragmatic and don't really care about some people who sat down one day and said: "this is free software". Talking about U++ BSD license, as long as I can take the source code, do absolutely whatever I want to it to deploy my application, this is free software, even though I have to include some copyright notice and acknowledge the source. Of course, I'm not doing anything radical to the source tree, but I do have small patches here and there. This means "free" for me.

It depends a lot on what you think "free" means. I think that GPL is forced freedom, so it is not freedom at all. LGPL on the other hand does give you some extra rights that I need, but I only applies to dynamic linking, and AFAIK U++ uses static linking.

Anyway, I'm no lawyer, but maybe using BSD license and not BSD like would be a good idea.

And I don't want to start a license flame war .

4. It has a "code in your face" attitude
This can be considered an disadvantage, but I like the fact that the entire source code is a click away. It can be considered a surrogate documentation. And thanks to BLITZ, any modification is soon usable, without having to run makefiles and move libraries around.

But there are also some disadvantages:

1. Documentation is poor
This doesn't affect me that often, because I already found out most of the stuff the hard way, but new users will have problems. I won't say anything more regarding this subject, because we are working on it and I'm sure we'll have good documentation in a not so distant future.

2. It does have it's quirks and idiosyncrasies
Of course, this is just a matter of taste and others might not see it. Also, some parts are not as well designed as others, and some similar components do lack a similar interface, but in the long run it doesn't really affect users IMO. Also, some implementations are overcomplicated for no good reason, while others are just complicated to achieve the necessary functionality. This also contributes to turning new users away. It is just the result of the age of the project and multiple people working on it. Maybe some day we'll have something like 2015.1 Cleaned Up Edition, with deprecated stuff stripped out and all interfaces designed with uniformity in mind.

3. It is C++ code.
I don't have a problem with the language itself, rather with the compilation tool chain. It is

extremely unproductive. I'm talking especially about build times. This is such an issue, that I think that there is absolutely no reason to pick up a C++ based library for absolutely no reason. In theory. In practice, you are stuck with all those libraries you need to interact if you don't want to reinvent a huge wheel. So basically you're stuck with C/C++ and must learn to live with it.

There is no other alternative. It will take years until D stabilizes, and maybe Walter Bright will get tired of it until then and return to C++ . And performance issues makes it hard to choose one of the popular and quite pleasant languages, like Ruby (and even then you would need a GUI toolkit; I don't think well have U++ Ruby bindings too soon).

Personally, I would love to see a Free Pascal port of U++. All we need is to gather a large community of Free Pascal enthusiast around U++, teach them C++, and the port will materialize itself in no time .

But seriously speaking, I am satisfied with U++. With better documentation and a better C++ parser coming along it will be all I need for quite some time. So what it it is not that popular. It does pretty much what I and other people that use it want and we'll stick with it! And if it does not do what we want, we can always come to this forum and have a nice chat about it, and if our demands are pertinent (and we have a patch on hand), we can get things changed.

Subject: Re: My explanation of why Ultimate++ is not mainstream
Posted by mirek on Thu, 14 Aug 2008 20:05:54 GMT
View Forum Message <> Reply to Message

amrein wrote on Thu, 14 August 2008 15:20
My referenced post:

> http://www.ultimatepp.org/forum/index.php?t=msg&th=94&am p;am p;start=0&
> http://www.ultimatepp.org/forum/index.php?t=msg&th=301&a mp;a mp;start=0&

Check dates.

But of course, in long term it would be nice to 'be mainstream'. OTOH, right now, I feel enormous community support. In fact, more that we, original developers, can handle at the moment

Mirek

Subject: Re: My explanation of why Ultimate++ is not mainstream
Posted by mirek on Thu, 14 Aug 2008 20:10:59 GMT
View Forum Message <> Reply to Message

cbpporter wrote on Thu, 14 August 2008 15:37
AFAIK U++ uses static linking.

Unfortunately, not in Linux. Libraries we need are not generally available as static. Plus, of course, THAT licensing issue...

Mirek

Subject: Re: My explaination of why Ultimate++ is not mainstream
Posted by cbpporter on Thu, 14 Aug 2008 20:20:19 GMT
View Forum Message <> Reply to Message

luzr wrote on Thu, 14 August 2008 23:10cbpporter wrote on Thu, 14 August 2008 15:37
AFAIK U++ uses static linking.

Unfortunately, not in Linux. Libraries we need are not generally available as static. Plus, of course, THAT licensing issue...

Mirek
I meant only for U++ packages. As in if I use Core, I will get static linking to Core. This makes LGPLing Core impossible, because it only applies to dynamic linking.

Subject: Re: My explaination of why Ultimate++ is not mainstream
Posted by mirek on Fri, 15 Aug 2008 09:27:03 GMT
View Forum Message <> Reply to Message

cbpporter wrote on Thu, 14 August 2008 16:20luzr wrote on Thu, 14 August 2008 23:10cbpporter wrote on Thu, 14 August 2008 15:37
AFAIK U++ uses static linking.

Unfortunately, not in Linux. Libraries we need are not generally available as static. Plus, of course, THAT licensing issue...

Mirek
I meant only for U++ packages. As in if I use Core, I will get static linking to Core. This makes LGPLing Core impossible, because it only applies to dynamic linking.

Yes, I guess it is a good point...

Mirek

Subject: Re: My explaination of why Ultimate++ is not mainstream
Posted by mirek on Fri, 15 Aug 2008 14:24:45 GMT

View Forum Message <> Reply to Message

License...

Well, I have spent some time investigating and for now, the most appropriate license I see is "MIT license".

It requires exactly what I want (do not remove copyrights) and it is widely known.

Subject: Re: My explaination of why Ultimate++ is not mainstream
Posted by amrein on Fri, 15 Aug 2008 15:50:17 GMT
View Forum Message <> Reply to Message

Ok

http://fedoraproject.org/wiki/Licensing#GPLCompatibilityMatr ix
http://en.wikipedia.org/wiki/MIT_License

But which one?

http://fedoraproject.org/wiki/Licensing/MIT

Or this one (OSI approved)?

http://www.opensource.org/licenses/mit-license.php

There is no much difference between the MIT licence and the modified BSD licence. It's a mess to know which version your software is using. Confusing. It's a good thing for lawyers thought, because if a company want to use your software, they will have to know what they can do and what they can't do with it.

-----8 < --- from wikipedia.org ------

The MIT License is similar to the 3-clause "modified" BSD license, except that the BSD license contains a notice prohibiting the use of the name of the copyright holder in promotion. This is sometimes present in versions of the MIT License, as noted above.

The original BSD license also includes a clause requiring all advertising of the software to display a notice crediting its authors. This "advertising clause" (since disavowed by UC Berkeley[5]) is only present in the modified MIT License used by XFree86.

The MIT License states more explicitly the rights given to the end-user, including the right to use, copy, modify, merge, publish, distribute, sublicense, and/or sell the software.

A 2-clause BSD-style license used by FreeBSD (and is the current preferred license for NetBSD) is essentially identical to the MIT License, as it contains neither an advertising clause, nor a promotional use of copyright holder's name prohibition.

-----8 < --- from wikipedia.org ------

From the two, I found the modified MIT licence published on the OSI website more easy to understand (http://www.opensource.org/licenses/mit-license.php, and it's FOSS compatible too). It's less restrictive.

Subject: Re: My explaination of why Ultimate++ is not mainstream
Posted by amrein on Fri, 15 Aug 2008 16:46:55 GMT
View Forum Message <> Reply to Message

One thing about the current licence:

Quote:(1) Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

(2) The above copyright notice and this permission notice shall be included in all copies of the Software and its Copyright notices. In addition publicly documented acknowledgment must be given that this software has been used if no source code of this software is made available publicly. This includes acknowledgments in either Copyright notices, Manuals, Publicity and Marketing documents or any documentation provided with any product containing this software. This License does not apply to any software that links to the libraries provided by this software (statically or dynamically), but only to the software provided.

(3) THE SOFTWARE IS PROVIDED "AS IS", ...

I study the present U++ licence to know exactly what it permit. If I understand the U++ licence well, at present, anyone can fork U++ or build software using any copylefted licence (copylefted licence from GNU.org: http://www.gnu.org/copyleft/copyleft.html). To be clear, I can fork U++, remove all your copyright from your files, then put LGPL or GPL in each of them with my modifications and my copyright then release my new U-- software (binary or source) as long as I use GPL or LGPL for example. Why? Because the modified software is not inbound in the paragraph (2) only the wall software. And also because LGPL or GPL force people to release the U-- code as required in section (2), so no need for your copyright. And you can still read the last sentence in (2) as a confirmation of what I just said.

So sorry if I said it wasn't GPL compatible. I'm not a lawyer. It's just not OSI approved and GNU approved (want to submit to know exactly if it's FOSS compatible?).

The MIT licence will give you the right to force people to let your copyright in any sources files from U++, even if they are modified. People will be able to distribute their application without acknowledgements in their about menu (nor show the licence for U++ binary if they release it as .dll).

I'm wrong?

Subject: Re: My explaination of why Ultimate++ is not mainstream
Posted by mirek on Fri, 15 Aug 2008 17:10:30 GMT
View Forum Message <> Reply to Message

amrein wrote on Fri, 15 August 2008 12:46
I'm wrong?

I do not know. I guess I have explained our 'licensing bussines' already

Seriously, I do not really care. All I want is to let people do with U++ whatever they want. The only thing I still care about is to leave our (c) somewhere and perhaps contact info....

But you are right that this thing sould be clearly presented - people are not lawyers and they are not aware about license details. That is why we have always presented this as "BSD" license because this is what most people know and consider permissive enough.

Anyway I think OSI approved "MIT" best expresses our minimal needs. The only problem I can see is that it is less known....

Mirek

Subject: Re: My explaination of why Ultimate++ is not mainstream
Posted by amrein on Fri, 15 Aug 2008 17:46:42 GMT
View Forum Message <> Reply to Message

MIT is "The Massachusetts Institute of Technology". They has mainly students in science, technology but also other areas of scholarship. Very good reputation in US.

The MIT licence is very easy to understand and complete. At least, this one will do what you want. Your old licence doesn't protect your copyright enforcement in a few situations (as soon as someone modify U++ source code).

Just add:

MIT licence is a very permissive licence. It's proprietary software friendly. It's also OSI approved (Open Source Initiative) and compatible with GNU GPL/LGPL licences too.

Only thing: there are no much software using it. Less than the BSD licence. When I begun this thread, I was thinking about releasing GPL software over U++. Then, I thought about merging things from GTK+/Glib too to have theming in-build in U++ (they use LGPL).

Subject: Re: My explaination of why Ultimate++ is not mainstream
Posted by mirek on Fri, 15 Aug 2008 18:02:33 GMT
View Forum Message <> Reply to Message

amrein wrote on Fri, 15 August 2008 13:46Then, I thought about merging things from GTK+/Glib too to have theming in-build in U++ (they use LGPL).

Ehm, I believe we have theming in-build in U++

Mirek
