
Subject: How to distribute some parts of Core in another library?

Posted by [cbpporter](#) on Mon, 05 Sep 2016 15:49:41 GMT

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

I think mentioned this before, but Vector, VectorMap and CParser are my favorite classes in U++. This goes only for Core, there are tons of favorites in CtrlLib.

CParser in particular has served me very well for years now, but alas, it is time to move on. Not because there is something wrong with it, but because of one of the projects is:

a. Not C++

b. It could still use CParser, but it feel like overkill but doable to hack C++ inter-op just for one class. The rest of Core is not needed here.

So I'm working on a leaner CParser like class, not in C++, which is re-implemented, so not a simple copy and paste, but in principle it is still the same thing. Sure, it may look a bit different, but on a logical level it is still CParser.

I would also like to OSS this class, under Apache License 2.0.

I have some weird memories of asking this before, but I can't find the thread, so here it goes: how would one go about this?

Subject: Re: How to distribute some parts of Core in another library?

Posted by [mirek](#) on Tue, 06 Sep 2016 15:26:14 GMT

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

cbpporter wrote on Mon, 05 September 2016 17:49 I think mentioned this before, but Vector, VectorMap and CParser are my favorite classes in U++. This goes only for Core, there are tons of favorites in CtrlLib.

CParser in particular has served me very well for years now, but alas, it is time to move on. Not because there is something wrong with it, but because of one of the projects is:

a. Not C++

b. It could still use CParser, but it feel like overkill but doable to hack C++ inter-op just for one class. The rest of Core is not needed here.

So I'm working on a leaner CParser like class, not in C++, which is re-implemented, so not a simple copy and paste, but in principle it is still the same thing. Sure, it may look a bit different, but on a logical level it is still CParser.

I would also like to OSS this class, under Apache License 2.0.

I have some weird memories of asking this before, but I can't find the thread, so here it goes: how would one go about this?

If it is re-implemented (in what, btw?), I guess you do not even have to ask or mention the original author.

I will certainly NOT sue you :)

Subject: Re: How to distribute some parts of Core in another library?

Posted by [cbpporter](#) on Tue, 06 Sep 2016 17:30:14 GMT

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

Well, it is certainly re-implemented, but visually it can range from different to 99% identical.

Some simple functions like Char, IsId and what-not are 100% identical in a language with C like syntax and there is no way around it.

And to be honest, for some functions, I did start of with cop and pasting the relevant code from CParser, making it compile as is and then redesigning it.

With redesign I mean things like making comments nest optionally (can give you the code for CParser too :)), making it not null-terminated, making it more Unicode and so on.

But it is the same design.

And this is why it is confusing. I certainly do not wish to appropriate something that is not mine, nor to not pay proper respects.

Nor do I want a 30 page license like: [http://www.ultimatepp.org/app\\$ide\\$About\\$en-us.html](http://www.ultimatepp.org/appideAbout$en-us.html)

Subject: Re: How to distribute some parts of Core in another library?

Posted by [koldo](#) on Wed, 07 Sep 2016 07:24:31 GMT

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

Quote:Nor do I want a 30 page license like: [http://www.ultimatepp.org/app\\$ide\\$About\\$en-us.html](http://www.ultimatepp.org/appideAbout$en-us.html)

:) :)

In fact those ~30 page are for plugin and for GCC-MINGW. U++ BSD license is only a few lines.

Theoretically even if you use just GCC you would have to include a big part of those ~30 page ;)

Subject: Re: How to distribute some parts of Core in another library?

Posted by [cbpporter](#) on Wed, 07 Sep 2016 11:12:26 GMT

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

koldo wrote on Wed, 07 September 2016 10:24

Theoretically even if you use just GCC you would have to include a big part of those ~30 page ;)

What?

Licenses are way to complicated!

Since when does the compiler you choose have to do anything with the license of software you use? The tool versus the object of the license?

For convenience I am also packaging TDM in a ZIP, as U++ does, so you don't have to go to the TDM site, use that installer, wait for the download and so on. Does this have license implications?

How about Clang? MSC? Clang on Linux? GCC on Linux?

Subject: Re: How to distribute some parts of Core in another library?

Posted by [mirek](#) on Wed, 07 Sep 2016 11:16:38 GMT

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

cbpporter wrote on Wed, 07 September 2016 13:12koldo wrote on Wed, 07 September 2016 10:24

Theoretically even if you use just GCC you would have to include a big part of those ~30 page ;)

What?

Licenses are way to complicated!

Since when does the compiler you choose have to do anything with the license of software you use? The tool versus the object of the license?

We distribute mingw with U++, that is why there is mingw license included.

BTW, for app released, there is now a nice tool "File/Project licenses", which scans packages and creates a complete list of licenses used...

Mirek

Subject: Re: How to distribute some parts of Core in another library?

Posted by [cbpporter](#) on Wed, 07 Sep 2016 11:41:27 GMT

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

mirek wrote on Wed, 07 September 2016 14:16

We distribute mingw with U++, that is why there is mingw license included.

BTW, for app released, there is now a nice tool "File/Project licenses", which scans packages and creates a complete list of licenses used...

Mirek

Thanks! Had no idea about that option. In TheIDE the only options from File I have ever used is Set main package, Edit file and Statistics :).

BTW, I have never understood why if Zlib is a plugin, plugin/z, then why must it be part of Core too? Instead of simply adding a new package which uses plugin/z if and only if you need zlib support?

I have never used Zlib, ZCompressStream and ZDecompressStream, but they look like something very useful to have in their own package.

But that just as an idea, I really don't care one way or the other.

So I can use "File/Project licenses" and copy&paste that into a license.txt for the command line tool binaries? And do the same + Help menu option for GUI?

That leaves only the source code for the rest.

Still juggling 4 different license sets, excluding mingw.

Subject: Re: How to distribute some parts of Core in another library?

Posted by [mirek](#) on Wed, 07 Sep 2016 13:26:01 GMT

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

Zlib is used by HttpRequest (because pages can be gzipped), that is why it needs to be in Core.

BTW, it even gets worse, xxhash (for HashBySerialize) and LZ4 (FastCompress) are now in Core too.

Mirek

Subject: Re: How to distribute some parts of Core in another library?

Posted by [koldo](#) on Thu, 08 Sep 2016 07:14:55 GMT

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

Quote:BTW, for app released, there is now a nice tool "File/Project licenses", which scans packages and creates a complete list of licenses used... 8) 8)

Subject: Re: How to distribute some parts of Core in another library?

Posted by [cbpporter](#) on Thu, 08 Sep 2016 08:19:23 GMT

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

Licenses are far too complex for sure.

Anyway, for now I included 4 LICENSE text files in the project + MINGW.

One, is for the library which is OSS and the original object of the discussion. This is Apache License 2.0.

I believe Apache can be combined with U++ licenses?

The library does not use U++, but it does include the CParser inspired class that I want to distribute.

So can you confirm that this scenario is OK?

Then I have 3 executables, which for now are not OSSed, but will be someday. They all use the library and are U++, so I took the Apache License and added all the content from File/Project licenses, obtaining 3 different LICENSE text files, one for each executable, because they use different packages. These are bundled in the distribution.

Then, there is the MINGW package, which includes all the original COPYING and README files.

Next, I need to figure out how to do the same for Clang because I would like to package that too.

Subject: Re: How to distribute some parts of Core in another library?

Posted by [mirek](#) on Thu, 08 Sep 2016 08:56:47 GMT

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

Are there at least 4 lines of code identical to CParser? If not, you do not need to copy any license by my standards (although some line "Inspired by U++ CParser in the code would be nice...")

Subject: Re: How to distribute some parts of Core in another library?

Posted by [cbpporter](#) on Thu, 08 Sep 2016 09:13:03 GMT

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

mirek wrote on Thu, 08 September 2016 11:56 Are there at least 4 lines of code identical to CParser? If not, you do not need to copy any license by my standards (although some line "Inspired by U++ CParser in the code would be nice...")

Hard to tell :o.

You be the judge of that!

```
def IsChar(c1: Char, c2: Char, c3: Char): Bool; const {  
  return term[0] == c1 && term[1] == c2 && term[2] == c3;  
}
```

```
bool IsChar3(char c1, char c2, char c3) const { return term[0] == c1 && term[1] == c2 && term[2]  
== c3; }
```

On paper it might not be identical. But logically it is...

Subject: Re: How to distribute some parts of Core in another library?
Posted by [mirek](#) on Thu, 08 Sep 2016 09:22:13 GMT
[View Forum Message](#) <> [Reply to Message](#)

I do not see identical lines:)

Mirek

Subject: Re: How to distribute some parts of Core in another library?
Posted by [cbpporter](#) on Thu, 08 Sep 2016 09:44:56 GMT
[View Forum Message](#) <> [Reply to Message](#)

mirek wrote on Thu, 08 September 2016 12:22 I do not see identical lines:)

Mirek
:o

Thank you!

OK, but for now I'm NOT including this file yet. May include it later or might move it to the U++ enabled code, and then there is no problem with licensing. In that content it might count as a fork.

And speaking of licensing, I removed MINGW from the package and will not include Clang. I packaged MINGW separately, in its own archive, and will rely on auto-detect for Clang. From Program Files/LLVM.

And started reading:
<http://www.oreilly.com/openbook/osfreesoft/book/index.html>

Subject: Re: How to distribute some parts of Core in another library?
Posted by [cbpporter](#) on Wed, 12 Oct 2016 14:07:19 GMT

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

How do you feel about me forking U++ to reduce .so usage to a minimum and NOGTK flag? :)

Is NOGTK still supported?

Subject: Re: How to distribute some parts of Core in another library?

Posted by [mirek](#) on Wed, 12 Oct 2016 15:11:49 GMT

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

cbpporter wrote on Wed, 12 October 2016 16:07How do you feel about me forking U++ to reduce .so usage to a minimum and NOGTK flag? :)

Is NOGTK still supported?

Sure go on.

Actually, it might be fun if some serious fork existed.

Mirek

Subject: Re: How to distribute some parts of Core in another library?

Posted by [cbpporter](#) on Wed, 05 Jul 2017 14:50:53 GMT

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

OK, I've been googling like crazy but still don't get OSS licenses. Even the best of sites with info don't explain stuff as if I neither understand or care about licenses.

So what is the difference between the license of a package and the license of its dependencies.

Here is a simple example: Let's say I create my package Foo that uses U++. I need to:

1. Select a license for it.
2. Make sure it is compatible with all the dependencies, i.e. the packages used from U++.

At this point lets say Foo goes under MIT.

I want to put Foo in GitHub. In gitHub , I select the MIT license.

So at this point my project has MIT license, or MIT + about 4 other that come from U++? Some I can't even recognize, some are BSD. Are all these BSD licenses the same?

Where do I put my own license and where do I put U++ licenses. In the github repo. How about in the pure binary distribution.

How do I determine the copyright year and do I need to register somewhere as a copyright

owner?

Can I change licenses after the first release?

God, the world of OSS license is a clusterfuck. I need an explanation as clear as 1 + 2.

Subject: Re: How to distribute some parts of Core in another library?

Posted by [mirek](#) on Wed, 05 Jul 2017 16:38:06 GMT

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

AFAIK (and as far as I care):

Foo has a license and it ends there.

Now if you create an application (where you do have multiple packages, each with its own license), that is the point you should start investigating whether licenses are 'compatible' and what are requirements.

In U++, only packages that are allowed to be used freely in commercial application without the requirement to provide sources are allowed to upp.src (there used to be an exception, but that was just helper package for theide, so no impact on applications created). Anyway, you still should check requirements.

TheIDE even has a tool for it: File/Project licenses...

Subject: Re: How to distribute some parts of Core in another library?

Posted by [cbpporter](#) on Wed, 05 Jul 2017 22:48:08 GMT

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

So a package has its own license and that's it.

And you should check compatibility, but beyond that you shouldn't care about dependencies?

So far I've done only independent OSS (no dependencies) or non-OSS binary distributions (lumped every single U++ license in a single file called license.txt).

Subject: Re: How to distribute some parts of Core in another library?

Posted by [mr_ped](#) on Thu, 06 Jul 2017 01:22:51 GMT

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

Generally speaking MIT/Apache/BSD type of license allows you basically anything, just a credit in documentation is desirable.

GPL requires you to publish modified sources. But only to your customers, doesn't mean you must provide them as free download on Internet for everyone.. but the very first customer who will download it may freely post it on Internet, if he wish so, so count GPL as "must publish src". Also if you use just small library under GPL, it will "infect" the whole app src with the license, so you must publish the whole source, not just that small library.

LGPL, when used for some library, and you link the library dynamically from your app, does NOT require you to publish src of your app, only the source of the library is enough. When linked statically, IIRC it means you have to publish also app src.

That's how I understand the OSS licenses, this is of course not a lawyer quality advice...

Subject: Re: How to distribute some parts of Core in another library?

Posted by [cbpporter](#) on Thu, 06 Jul 2017 07:28:17 GMT

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

mr_ped wrote on Thu, 06 July 2017 04:22

GPL requires you to publish modified sources. But only to your customers, doesn't mean you must provide them as free download on Internet for everyone.. but the very first customer who will download it may freely post it on Internet, if he wish so, so count GPL as "must publish src". I didn't know that bit that you were not required to publish it yourself. Thanks!

Otherwise this is high level stuff I found on my own. I'm having problems googling the detail stuff.

The bit where a package is its own license and not the sum license of its dependencies was one such detail.

Another is if you need to register as a copyright owner. I'm about to release a new hand written C++ bit of code on GitHub next week. I can choose its license, but how do I determine the copyright. Do I put just 2017 because it is 2017? Who do I put as the copyright owner? Does the year count? When does it expire? Are some years netter than others or is it not important.

I also want to upload some generated C++ code. The input is not C++, the output is C++. Is the output the same license as the input? Does the license of the generator impact the license of the output?

Many questions!

Subject: Re: How to distribute some parts of Core in another library?

Posted by [mirek](#) on Thu, 06 Jul 2017 16:16:17 GMT

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

cbpporter wrote on Thu, 06 July 2017 09:28
I'm having problems googling the detail stuff.

That perhaps because nobody really cares that much. The lore is "GPL is bad for bussiness, rest is OK". (Note that "bad for bussiness" does not need to imply a negative thing in all cases). IMO thats all that is to be known.
