
Subject: Re: U++ 2008.1 rpm
Posted by [amrein](#) on Sat, 16 Aug 2008 22:08:50 GMT
[View Forum Message](#) <> [Reply to Message](#)

(THE CORRECT THEIDE-48.PNG IS AT THE END OF THIS MESSAGE)

luzr wrote on Sat, 16 August 2008 15:31amrein wrote on Sat, 16 August 2008 09:11
I'm talking about the web site

There is always a room for improvement. The problem is lack of manpower to do them.

Quote:
, the licence

GPL or LGPL would ruin U++ business-wise.

MIT is the most likely.

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

I asked a question (for free... hard to achieve) to a lawyer of my ex employer (a friend). His short answer is:

BSD license = Yours + "If you modify U++ source, neither the name of the U++ main teams and contributors may be used to endorse or promote products derived from this software without specific prior written permission."

BSD license = Do whatever you want but: (1) if you release modified source code, don't use our names (2) if you release the binary generated with the unmodified source code and your software doesn't link with it, you must provide the original and complete source code of our software too.

MIT license = Yours + "The above copyright and license must be included in all copies or substantial portions of any released source code." - "If the binary is provided as this, if your software doesn't link with it, if you have made no modification into U++ source code, you must publicly acknowledge the use of the software in your software/doc and must make the U++ source code available publicly."

MIT licence = Do whatever you want but if you release a file including part of our source code, this license and the copyright holders (U++ team and contributors) must be in the source file including our source code.

Quote:
Quote:

, the doc

Working on it. No doxygen for several reason (reconsidered many times).

It would be so cool to have topic++ been able to help people edit doxygen documentation directly into the source. A complete doxygen editor + topic++ (replacing doxygen completely).

It's hard to maintain source code documentation. It's also hard to come back to old source and not find any `//` or `/* */` to comment the source. As soon as you are more than 2 developers, adding comments in source code is really welcome.

Quote:

Quote:

, the source directory structure,

Do not expect this to change. It is one of main selling points of platform.

Besides, do not expect U++ to "play by rules" in all cases. That is the point.

Sure, or it wouldn't be able to evolve quickly.

Quote:

Quote:

the "static link only" and but no official dynamic library,

"static link only" is only the most logical and straightforward path.

I would like somebody started "official dynamic library". The only problem is that:

- I am not sure HOW to do that. U++ is highly modular, it will be difficult to decide how these modules translate to .so (having 50 libraries out of U++ does not seem feasible

- Personally, I am not even INTERESTED in doing it. It does not solve a single problem and only adds a couple of them, at least for me.

(the fact that I am not interested does not mean I would not support such thing).

Thanks about info about virtualbox etc...

I have now to consider the next "infra" phase - those nightly builds.

In fact, another problem there is how to make possible for maintainers to maintain releases via svn. I suspect we should use something like specific folder in svn, which gets exported each night

and "run-parts" applied on it...

Mirek

You welcome.

Many thanks for all those answers.

Concerning svn, you can use user groups to prevent someone from touching part of the repository. You could create different branches (like Linux kernel, like upp2008.1_amrein) and give them full access only to those branches (using group policy).

The big problem: this is not open enough to be interesting and also hard to maintain. This project is not big enough for multiple branches. Having separated branches will cause problems for merging if contributors go in too different paths. This is less man power in the main trunk. Anyway, if really required, for new feature for example, it can be interesting to have another branch, but without big permission restriction.

Anyone can play at home with a svn local copy for small patch. Here is how most FOSS teams work:

- Other FOSS project give permission in a meritocracy way.
- Only frequent and trusted contributors have write access.
- External contributors (anyone) can send svn diff (better for trunk) or just diff. They send them via the mailing list, the bug tracker or via email if any.
- If a contributor with write access want to do something bigger than a small patch (move/rename files, change the way TheIDE works, ...), he must discuss it first publicly and get approval from the team + the svn/project admin.
- They use groups to prevent modifications only if they have very critical parts and they give write access to those parts only to senior contributors.

File Attachments

1) [theide-48.png](#), downloaded 765 times

