

---

Subject: Re: Does the provided upp.spec works for you and on which distro?

Posted by [amrein](#) on Tue, 02 Sep 2008 10:45:01 GMT

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

---

luzr wrote on Tue, 02 September 2008 10:06...

In fact, it is considered that we perhaps might create a "runtime environment" for U++, to make it behave even more like scripting language. It would just use GCC in the process

Mirek

Binding for Python, Perl, Rubby, Tcl, Gambas (<http://gambas.sourceforge.net/>), Java... Could be done in hours if we had a dynamic library.

It should be possible to create a new package like "libupp" statically linking with all upp, and the output should be a dynamic library libupp.so.0.8.1 instead of an application.

Resume

Since the beginning, I talked about website, doc, dynamic linking, version number, directories clean up, file names policy, class names policy, ... to answer a simple expectation (mine): "How can U++ have a wider audience and become mainstream in FOSS". What amaze me the most now: I'm trying to answer my question, but nobody, except me, see my solutions as interesting suggestions. I have no more idea to submit. The answer I got are completely justified in that context:

Website left menu: Generated from Topic++. Modifications should go there first.

Doc with title and plan: Get it and improve it, if the result is cool, we will use it.

Version number x.y.z (0.year-2000.svntag?): We prefer Year.Release. Use what you want if you work on a dynamic library.

U++ as a dynamic library too: Too much work. U++ and TheIDE API change too often. TheIDE use static linking.

Manage Topic++ doc as doxygen comments in the source: We prefer external Topic++ documentation.

Cpp an .h file names policy (all lower case for better portability): No need for policy, our programmers are used to current file names.

Class names without abbreviations (like Ctrl->Control): No thanks, this is what we are used to.

Directory clean up: Well perhaps TheIDE could be in its own Package/Assembly/Nest. Perhaps U++ directories structure are not clean enough for you but we are comfortable with it and we works on it each days, as many other U++ users

Something easier than "Assembly/Nest/Package": "Assembly/Nest/Unit"?

(...)

I have no more ideas. I won't be able to help as I thought I could. At least, I can help for the rpm build process.

---