Subject: Re: U++ talk
Posted by amrein on Tue, 02 Sep 2008 18:47:45 GMT
View Forum Message <> Reply to Message

As long as upp version is not higher than 0.99.999..., the API can broke. That doesn't matter much. A present It could be 0.8.1. No problem with anyone compiling with 2008.1.
2009.0 then? Anyone using static linking will certainly need to rebuild their application too. In FOSS, when a lib is updated to a new release, all distro rebuild their packages if they need it.
U++ don't need to care about API stability. Really. There are a lot of libraries modifying their API and some of them broke it for each release. If U++ API is broken from version X to version Y, than it's mean that new U++ version is better. Other FOSS software will just have to follow.

The only problem could be if the U++ team wanted to backport fix to the 2008.1 branch for example. Most distro do it themselft and submit patches (or you can get them from their source packages). They also see by themselves from svn what is a fix and what will break the official release. They don't get all fix because a perfect product (for them) are not perfect to sell (no need to update to next release).

Quote:I'm sorry, but I consider lower case names as deprecated as I do non-case sensitive paths. Even under Windows I always use proper cased names for files, and also except to get such names. So "Ctrl.h" is not a caseless or lower case name which was written with a capital "C" because Windows is not case-sensitive, it is really called "Ctrl.h" and referring to it as "ctrl.h" is an error IMO. And since all Unixes are case sensitive and Windows is case neutral, I don't think we have any issues left with this scheme. Also, if we change to lower case on Unix, all programs will cease to compile until we manually modify all include clause.

Yes. "#include" clean-up + a strict policy for all new .cpp and .h names.
You can find this policy in Qt, Epoc, Symbian, Gtk, wxWindows, WinCE... Most cross platform libraries use it.
You are used to U++ source code. You don't see it as I do.

Quote:The same applies to using more longer and intuitive names. How can we change the name of a file and except old code to compile. How can we change the name of a class and still except to link with the .so that we are going to produce?

A source clean-up + a strict lower-case name policy for all new .cpp and .h doesn't mean rename internal class. No need to change their names.
The exceptions: class using abbreviations shouldn't use abbreviations. No need to get used to those abbreviations and source code is easier to understand.

Quote:Doxygen: I'm all for reusing technology which works (i.e. doxygen) and wouldn't mind using it instead of Topic++. Topic may be more intimately tied to A++, but I don't think this mattes that much since docs are not autogenerated.

On the other hand, using doxygen would increase compilation times. I'll go crazy if I use another 0.5 second time per build. I would hate to have to abandon both the language and framework that I have grown to love and switch over to .NET because of trivial issues like compilation speed which should have been fixed decades ago.

Doxygen comment are just C++ comment. The C++ parser is optimised to skip them.
Things like dependencies, methods, attributes, class hierarchy, call graph are computed by doxygen.

Note: When you recompile, you don't recompile U++ (BLITZ...). Just your own application. We could say just your modified files.

How to include an image in the final doc: html tag in the doxygen source comment.

How to include a link to another doc or source code(tutorial...): html tag in the doxygen source comment.

How to include a link to load and see the original source code: one option to activate in the doxygen configuration file.

Is it possible to create an external documentation referring to the html output. Something like we already have on ultimatepp.org? Yes.

Do you know a powerful tools that will extract, insert and manage automatically source code comments for doxygen? No

How can people still add new documentation without touching the source code? Well. You will hate me. An applications like linguist (from Qt) use one tool output, lupdate tool (from Qt too), to extract all translation from .cpp and output a xml file. You can then do the translation job on this file with linguist. It's like TheIDE translation tool but with 2 stand alone applications. They don't reinsert the translation, they load it at run time to save memory (only one language for each xml file because with big application you can have several Mo in one file), but, in a documentation tool like Topic++, this metadata could be validated and reinserted by the documentation project manager.

Then, why still use doxygen: Because doxygen can do a lot of things that Topic++ can't right now. Because if Topic++ become an universal doxygen source code documentation tool it will have a wider audience and will also promote TheIDE and U++.

I include a simple doxygen file to test. Download it in ~/upp, cd into ~/upp then type "doxygen". Doc will be generated in ~/upp/documentation. I used doxywizard (from doxygen package) to create it.
I didn't add "search support" because I guess all people here don't use apache+php on their Pc.

File Attachments

1) Doxyfile, downloaded 339 times