Subject: Re: Few questions
Posted by mr_ped on Fri, 05 Nov 2010 08:41:54 GMT
View Forum Message <> Reply to Message

OvermindDL1 wrote on Fri, 05 November 2010 05:48 - Is there any Boost package, preferably to the trunk?

No. Does it even need a package, isn't it just set of headers to include? (I don't use Boost, so I'm not sure)

About creating package... simply create package, include the .cpp/.h/other project files (.cpp inclusion is crucial for project build, .h and other directly non-build-able files just for easy edit trough IDE).
Then add the package to whatever other package where you want to use it.

If you refresh the sources of that boost package trough SVN, build process should detect changes in "modified" timestamp and rebuild everything related. (although with such fundamental classes used all over place it's more safe to hit Rebuild All, especially if you get some weird error after ordinary build)

I'm very likely wrong, but as long as Boost is just a helpful .h header, I don't see any problem with building it and using it in U++ (without any modification to source).

Quote:
 - Is U++ able to update the source of a project automatically using svn checkout/export (does it come with an svn client in other words that I can call from script?).

 TheIDE has SVN support in menu, it does use common command line client (svn.exe in win and svn in POSIX), which is IIRC not part of U++ installation. So TheIDE is just an front-end for classic SVN client, nothing more. (and if you prefer to have more control about what's happening under hood, I would suggest to use external tool like TortoiseSVN, TheIDE is a bit too simplified for my taste)

Quote:
 - If I do port boost/trunk to a reusable UPP package, would anyone be interested in it in the bazaar?

 Me: not now, nor planning it. But it never hurts to have something working in bazaar, so in case the need arise, you can try it out in ~3 clicks of mouse.

Quote:
 - Would also think of making a boost/upp package as well to include headers to let Boost work with many UPP structures, one of the nice things about boost is its extensibility to support other container, structure, etc... types, would this be welcome?

 This requires some work and R&D and experiments, right?
Of course anything of this is welcome and highly appreciated, just make sure your effort is well

documented, so you will save time of other U++ users and make a path for them.
(If your idea of "welcome" is that it will get heavily adopted soon... take a break, this is BSD world, you give things away and see what picks up, but don't hold your breath.)

Quote:
- Hard to get an indication from these forums (the bazaar forum appears to have even fewer submissions then Boost, and it is *really* hard to get a submission into Boost...), but how active is the community and so forth?

Community is small, but very active. Getting submission into bazaar is not very hard (even of lower quality or work in progress), getting patch into uppsrc is reasonably (much more) difficult (90% of good patches get in, and 90% of ballast get rejected, I don't think you can do much better with project of this size).
Bazaar pretty much follows it's name. If you dare to share your work, you will very likely get it there.

Quote:
- Is there a good third-party package repository of various libraries 'ported' to the U++ package format for ease of use U++ projects?

Bazaar is best place to start, then this forum. I'm not aware of anything for U++ being advertised elsewhere and not here, so I guess the answer is "no".

Quote:
- Is there a mailing list, IRC, newsgroup, etc..., or just these forums?

Forums are fastest and most active channel. There were also some IRC discussions, and also already a small live session in Prague, then PM on forums and some IM, but basically if you are active on forum, you shouldn't miss anything.

Quote:
My background, been programming in C++ for going on 18 years, huge advocate of Boost, regardless of its monolithicly huge compile times in certain libraries (Spirit is such a great parser...), and I have a tendency to push things to their limits. Generally program in Visual Studio with the Visual Assist plugin for Windows, *nix, and Mac programming work (Virtual Machines are great) due to how amazingly productive it is, Visual Assist practically reads my mind it seems, testing out U++ + TheIDE to see if it will make a better 'native' multi-platform build engine and IDE, and as stated above, I am impressed with the layout of how builds work, tag based systems can be very nice.

Nice to have you here, you can surely contribute by feedback of your U++ experience (as it's usually quite a rough ride at the beginning) and I hope the community will be able to help you. Although with your experience and being a avid boost user you are not truly "compatible" with U++, so don't be surprised if it takes some time for you until you find a good way how to exploit U++ for your advantage. You will have to relearn some things for no obvious gain, and some things you will very likely do more efficiently in MSVS+boost, if you are good at it. Still if you ask me, U++ is worth a try, especially if you like to push things on limit, there are areas where U++ is clear winner.

Quote: - Have you though about including clang++ as a build engine? You could even import its sema library to provide full VisualAssist-level and above code verification, analysis, intellisense, etc...?
clang in latest version is capable to build TheIDE, dolik is toying with it. Basically the building with clang already works, search forum to see what additional setup is needed. (you need working GCC build method and then just add clang build method as an replacement variant to GCC build)

And, one more thing about additions to Bazaar: BSD/MIT license or compatible is a good start for new package. (From reading the boost license right now I would say it's BSD compatible, but I didn't dig deep into it) If it's not compatible, you should mark it clearly as such (or maybe not include it into bazaar at all and push it just to forums), because most of the U++ users expect everything in bazaar to be free to use in commercial apps without consequences.