Subject: regex (assembly) from boost
Posted by hojtsy on Sat, 28 Jan 2006 09:22:16 GMT
View Forum Message <> Reply to Message

I am really missing the regular expression support I had in Qt. I am trying to use the boost regexp library in one of my u++ projects. I suppose that the clearest solution would be to create an assembly for boost, and packages for each module of it, with dependencies, etc. Is there anybody who already did this?

Subject: Re: assembly for boost
Posted by mirek on Sat, 28 Jan 2006 10:09:08 GMT
View Forum Message <> Reply to Message

Like this?

http://www.handhelds.org/~zecke/apidocs/qt/qregexp.html

Should I put that on ToDo list?

I think that regular expressions should definitely be part of Core, but so far I am not quite sure what syntax to choose.

Mirek

Subject: Re: assembly for boost
Posted by hojtsy on Sat, 28 Jan 2006 10:16:50 GMT
View Forum Message <> Reply to Message

luzr wrote on Sat, 28 January 2006 05:09Like this?
http://www.handhelds.org/~zecke/apidocs/qt/qregexp.html

Yeah, like this. Plus look at the find, findRev & replace methods of QString. Having both substring search&replace and regexp search&replace on overloaded methods of QString produces very clear and intuitive client code.
Quote:Should I put that on ToDo list?Yes please.
Quote:I think that regular expressions should definitely be part of Core, but so far I am not quite sure what syntax to choose.You mean the Regexp flavour, or the API syntax?

Subject: Re: assembly for boost
Posted by unodgs on Sat, 28 Jan 2006 16:07:12 GMT
View Forum Message <> Reply to Message

I think for u++ the best solution is to use PCRE:
http://www.pcre.org/
It has BSD license. I was playing a bit with it. It compiles and run without any problems. It provides Perl syntax for regular expressions (the best IMO). The library is widely known and used.

Subject: Re: assembly for boost
Posted by gprentice on Sat, 28 Jan 2006 20:09:10 GMT
View Forum Message <> Reply to Message

Performance comparison of PCRE with the two boost regexp libraries. Don't know what to make of this - xpressive seems to lose.
http://article.gmane.org/gmane.comp.lib.boost.devel/131338/m atch=pcre

Comments from John Maddock on xpressive
http://article.gmane.org/gmane.comp.lib.boost.devel/131762/m atch=pcre

Comments from A Alexandrescu in 2002 on boost::regex code size - 500KB compiled
http://article.gmane.org/gmane.comp.lib.boost.devel/75813/ma tch=pcre
(dunno if it's still a problem)

Graeme

Subject: Re: assembly for boost
Posted by hojtsy on Sun, 29 Jan 2006 13:08:38 GMT
View Forum Message <> Reply to Message

Even though regexps are interesting, boost also provides several other libraries. I would like to go back my initial question: how to integrate the boost libraries into an u++ application? It seems strange that nobody did this before because boost has a simillar concept of providing rich functionality with "agressive" use of c++ features, like u++. I suppose that anybody who finds u++ appealing would logicaly like boost too.

Subject: Re: assembly for boost
Posted by mirek on Sun, 29 Jan 2006 14:19:22 GMT
View Forum Message <> Reply to Message

First - to integrate boots, the best way IMHO would be to create a "boost" package. Just put boost into the directory and add all .cpp files to package...

Now for boost and aggresive use of C++ - in fact, boost follows quite different direction as it based on the premise that standard library / STL is the right way to go (actually, that is quite understandable).

U++ has taken quite different path, as we have found several problems in STL/standard library, that to us seems to severly influence both application development effectiveness AND runtime perfomance.

Means, U++ is compatible (why not , but follows own way in many things (and yes, for many people this is quite a problem, the only apology is that Qt seems to do the same

---