Subject: Some projects are ready to become open-source
Posted by Mindtraveller on Thu, 24 Apr 2014 15:07:52 GMT
View Forum Message <> Reply to Message

Hi! It's been a while since I've posted anything to forum, but still U++ is the best thing I've used.
There's a number of U++ projects I'd like to make open-source and make them as bazaar packages. But I don't really know if people really need them. Besides, some of them require additional work to become really cross-platform.
So I'll post a list of possible packages, and if someone considers them useful, please vote here.

1. First of all, I'd like to present multithreaded worker-based server class. I don't know if it duplicates any of Skylark functionality (never really used it). The main idea behind it was to make lightweight and easy to use solution for high load servers. Upon incoming http-request, a free worker thread is reused or new worker thread is automatically created. The class was made with http keep-alive support in mind.

2. Persistent storages. Rather simple wrapping templates around U++ storage classes adding functionality for accessing them in multithreaded environment with background auto-save to file system. Classes are very helpful for effective management of not-very-big amounts of data where SQL-based solutions seem too "heavy". There are plans to decrease memory usage and serialization effectiveness for these storages if it comes to it.

3. BNF (Backus-Naur form) classic implementation based on stack machine. Stable but really needs upgrade to support some jit-compiled code for internal functions. I'd be very grateful if anyone is interested. I don't use BNF package very frequently alone, instead it is widely used as a dependency of next package.

4. Industrial-grade, completely asynchronous serial communication library. Based on Apache-licensed serial library. It doesn't work on send/receive bytes level. Instead, it operates with BNF-defined protocols, parses them from file upon startup. User code operates with high-level things like input and output Value vectors. If you want some simple experiments with serial port, this is too heavy solution. But if you write big project where serial communications involved, this is what you might dream of. Very robust and well tested through years.

5. ToJSTime/FromJSTime functions for browser time representation support. Very small, doesn't worth creating package for them.

6. ConvertTextToUtf8 function. Charset detection function which makes utf-8 text from almost any encoding. Uses UCI & iconv libraries. Succesfully detects almost any encoding and converts text with at least 10 characters long (smaller text might give too big detection error).

7. U++ packaged htmlcxx library. Very lightweight but accurate HTML/CSS parsing library with LGPL license.
Added utility function SafeguardHTML (with utf8 conversion from previous function). The main idea is to create 100% safe html output from user input in any encoding.

As always, any suggestions or critics are welcome.