

Hello Mirek,

That's great news!

Quote:

Now the hard part is to decide how to "parcel" the U++ into libraries. I am also afraid there will have to be like 12 versions of each library:

- USEMALLOC (not using U++ allocator)
- STDNEW (using standard new/delete, but U++ allocator where possible)
- with upp allocator
  
- Debug
- Release
  
- Linked with dynamic CRT
- Linked with static CRT

My point of view here is simple - the less variants we provide the better. In case of allocator I would choose only one option in context of .lib. Upp allocator should be exclusive for upp as a platform. So, I opt that we should use STDNEW only. Very good performance on library level, but user space is untouched. I think this is what most users want and do not create headache what option should be choose. In the very first version we could go with USEMALLOCK only.

In initial version we should only limit to RELEASE. In context of DEBUG works properly there is a need to distribute sources as well. In the first version the debugging of upp library should be only reserved to platform (TheIDE). I think initially there will not be many requests to add support for debug mode.

Providing CRT should be application responsibility. So, probably dynamic option is fine for us at least at the beginning. Mirek, do you see any objections difficulties here? The problem with static is that sometimes the CRT runtime is LGPL (Linux), so if we would like to go that path then we will need to make sure that we will do not link with libraries with that kind of license. I know that there is clang library that has BSD license. Anyway, this adds additional point of complexity, so initially I would focus only on dynamic version.

So, backing to math we could create only one variant instead of 12 and make whole solution simple :) This will also allow us to see how market will response and tune up in the future. KISS :)

-----

In context of includes - will include works like this:

```
#include <Upp/CtrlLib/CtrlLib.h>
```

or do you plan to

```
#include <CtrlLib/CtrlLib.h>
```

I know that people might have objections for the second variant, so we should avoid it.

Klugier

---