Subject: How to manage UPP namespace

Posted by Didier on Sat, 03 May 2014 21:22:12 GMT

View Forum Message <> Reply to Message

This is a problem on which I stumble regularly.

Some packages are inside UPP namespace, some have they're own namespace, some don't have any namespace at all and some have 'using namespace Upp;' in they're headers !!
This leads to name clashes and ambiguous problems when using some packages or when including external resource code.

I think their should be some recommendations on how to correctly manage this (if you wan't you're package to be usable by others of coarse), like the following:

always put code inside namespace Upp (at least for classes meant to be used by other packages) Never put 'using namespace xxx;' inside headers

Subject: Re: How to manage UPP namespace

Posted by dolik.rce on Sun, 04 May 2014 05:56:01 GMT

View Forum Message <> Reply to Message

Hi Didier,

I agree that there should NEVER be "using namespace" in header that is meant to be used by others. I don't really care if some code is in Upp namespace and some in global, sometimes it is hard to tell whether it is really part of U++, e.g. for packages in bazaar, it is kind of blurry line.

Another related thing related to namespaces that I would like to point out is the existence of NONAMESPACE flag. It can be used to move all U++ code to global namespace, which might be preferable for someone who doesn't like to deal with "using namespace Upp;" in every file. But for this flag to work, all the code must use correct macros, defined in Core.h:#ifdef flagNONAMESPACE

```
#define NAMESPACE_UPP
#define END_UPP_NAMESPACE
#define UPP
#else
#define NAMESPACE_UPP     namespace Upp {
#define END_UPP_NAMESPACE };
#define UPP               Upp
#endif
```

There is a lot of code (some even in Core IIRC), that does not follow this convention, making NONAMESPACE flag useless.

Best regards,
Honza