

---

Subject: Re: 2020.2rc1

Posted by [Klugier](#) on Thu, 15 Oct 2020 20:18:19 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Hello,

It seems that we have one more problem with template generation. By default in generating CtrlLib and Skylark application there is "using namespace Upp;" clausal in header file, which is wrong for obvious reasons.

Here is app generated with the template:

```
// ClockApp.h
#ifndef _Clock_Clock_h
#define _Clock_Clock_h

#include <CtrlLib/CtrlLib.h>

using namespace Upp;

#define LAYOUTFILE <Clock/Clock.lay>
#include <CtrlCore/lay.h>

class ClockApp : public WithClockLayout<TopWindow> {
public:
    ClockApp();
};

#endif

// main.cpp
#include "Clock.h"

ClockApp::ClockApp()
{
    CtrlLayout(*this, "Window title");
}

GUI_APP_MAIN
{
    ClockApp().Run();
}
```

It should be:

```
// ClockApp.h
#ifndef _Clock_Clock_h
```

```

#define _Clock_Clock_h

#include <CtrlLib/CtrlLib.h>

namespace Upp {

#define LAYOUTFILE <Clock/Clock.lay>
#include <CtrlCore/lay.h>

class ClockApp : public WithClockLayout<TopWindow> {
public:
    ClockApp();
};

}

#endif

// main.cpp
#include "ClockApp.h"

using namespace Upp;

ClockApp::ClockApp()
{
    CtrlLayout(*this, "Window title");
}

GUI_APP_MAIN
{
    ClockApp().Run();
}

```

There is one option I would see there, but this is out of release scope. Upp namespace agnostic template (as option):

```

// ClockApp.h
#ifndef _Clock_Clock_h
#define _Clock_Clock_h

#include <CtrlLib/CtrlLib.h>

namespace Upp
{
    #define LAYOUTFILE <Clock/Clock.lay>
    #include <CtrlCore/lay.h>
}

```

```
class ClockApp : public Upp::WithClockLayout<Upp::TopWindow> {
public:
    ClockApp();
};

#endif

// Clock.h
#include "ClockApp.h"

ClockApp::ClockApp()
{
    CtrlLayout(*this, "Window title");
}

GUI_APP_MAIN
{
    ClockApp().Run();
}
```

This option is the best for advanced users. In almost 90% your app shouldn't belong to Upp namespace. You could mix it easier with other namespaces like std. Only Core elements should belong there, but for simplicity we just put app code there...

In Turtle everything is fine. I think Oblivion fixed it some time ago.

---