Subject: Re: namespace agnostic layouts refactored
Posted by mirek on Sat, 07 May 2022 11:27:52 GMT
View Forum Message <> Reply to Message

Klugier wrote on Sat, 07 May 2022 09:58Hello Mirek,

It seems that with new approach Upp namespace is populated. Here is tutorial/Gui16b/main.cpp file that compiles fine and it shouldn't:

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

#define LAYOUTFILE <Gui16b/dlg.lay>
#include <CtrlCore/lay.h>

struct MyAppWindow : public WithDlgLayout<TopWindow> { // Upp:: prefix no longer need and it should...
 MyAppWindow() {
  CtrlLayout(*this, "MyDialog");
 }
};

GUI_APP_MAIN
{
 TopWindow top; // Upp:: prefix no longer required here...
 MyAppWindow().Run();
}
```

In context of placing anonymous namespace in header file, a lot of linters detect this as an warning. More info here. BTW, I compiled with CLANG and GCC. The second compiler produces a lot of warnings:

```
In file included from
/home/klugier/upp/.cache/upp.out/tutorial/CtrlLib/GCC.Debug.Debug_Full.Gui.Shared/CtrlLib$blitz.cpp:238:
/home/klugier/upp/git/uppsrc/CtrlLib/PrinterJob.cpp:228:7: warning: 'Upp::PrinterDlg' has a base 'Upp::{anonymous}::WithPrinterLayout<Upp::TopWindow>' whose type uses the ano
    nymous namespace [-Wsubobject-linkage]
  228 | class PrinterDlg : public WithPrinterLayout<TopWindow> {
      |
```

IMO, we shouldn't add this warning to the blacklist like we did for "-Wno-logical-op-parentheses" for Clang.

Can not we just follow old approach, but add new types (frames) like requested in #73? Anonymous namespace approach seems to have a lot of drawbacks.

Klugier

OK, the easiest thing to do is to go back for Upp:: for types (that is actually just CtrlLib.usc setting) and keep "using namespace Upp;" (inside functions Set_Layout) for values.

Any other idea? The ideal would be "using namespace Upp;" inside WithXXXLayout struct, but that is not valid C++...

Mirek

---