Subject: namespace agnostic layouts refactored
Posted by mirek on Sat, 07 May 2022 07:28:19 GMT
View Forum Message <> Reply to Message

So this is about ability to use layouts without "using namespace Upp". In the past, I have tried to solve that issue in layout designer, adding Upp:: before typenames and stuff, only to forget something on each iteration.

Today I have got completely different idea:

https://github.com/ultimatepp/ultimatepp/commit/4f49f919a27855c9dee0ad33ec0864aa18de7144

and actually reverted layout designer to previous mode (of course, retained the code for ignoring those now obolsete Upp:: texts).

Do you see any problem with this new approach? I have read somewhere that unnamed namespaces in headers are bad, but I think in this particular case it should be fine...

Subject: Re: namespace agnostic layouts refactored
Posted by Klugier on Sat, 07 May 2022 07:58:57 GMT
View Forum Message <> Reply to Message

Hello 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

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

Subject: Re: namespace agnostic layouts refactored
Posted by mirek on Sat, 07 May 2022 15:14:47 GMT
View Forum Message <> Reply to Message

After rethinking the issue, you are right and I am completely wrong about this. Will revert whole "refactoring" and go proposed path.

Subject: Re: namespace agnostic layouts refactored
Posted by mirek on Sun, 08 May 2022 11:40:14 GMT
View Forum Message <> Reply to Message

OK, another refactor finished.

All ok now?

Subject: Re: namespace agnostic layouts refactored
Posted by Klugier on Sun, 08 May 2022 13:57:11 GMT
View Forum Message <> Reply to Message

Hello Mirek,

Basic scenario works correctly (Gui16b), however bugs raised on GitHub are still present. When you switch from default frame to InsetFrame in Layout designer it leads to compilation error. The reason is simple, layout designer just do not prefix InsetFrame() function:

```
ITEM(Upp::EditString, text, SetFrame(InsetFrame()).LeftPosZ(48, 92).TopPosZ(8, 19))
// Should be...
ITEM(Upp::EditString, text, SetFrame(Upp::InsetFrame()).LeftPosZ(48, 92).TopPosZ(8, 19))
```

Just cosmetic - could we change "default" -> "Default" for default frames. All frames start from capital letter.

Do you also plan to fix #74? In my opinion for empty class it should have no properties (just empty area) like for empty ctrl entry in usc file.

Klugier

Subject: Re: namespace agnostic layouts refactored
Posted by mirek on Sun, 08 May 2022 22:26:55 GMT
View Forum Message <> Reply to Message

Klugier wrote on Sun, 08 May 2022 15:57Hello Mirek,

Basic scenario works correctly (Gui16b), however bugs raised on GitHub are still present. When you switch from default frame to InsetFrame in Layout designer it leads to compilation error. The reason is simple, layout designer just do not prefix InsetFrame() function:

```
ITEM(Upp::EditString, text, SetFrame(InsetFrame()).LeftPosZ(48, 92).TopPosZ(8, 19))
// Should be...
ITEM(Upp::EditString, text, SetFrame(Upp::InsetFrame()).LeftPosZ(48, 92).TopPosZ(8, 19))
```

Just cosmetic - could we change "default" -> "Default" for default frames. All frames start from capital letter.

Do you have latest ide? Adding namespace to enums (like InsetFrame) is exactly what I did...

Quote:
Do you also plan to fix #74? In my opinion for empty class it should have no properties (just empty area) like for empty ctrl entry in usc file.

Klugier

I am not opposed to that.

Mirek

Subject: Re: namespace agnostic layouts refactored
Posted by Klugier on Mon, 09 May 2022 16:09:23 GMT
View Forum Message <> Reply to Message

Hello Mirek,

After rebuilding TheIDE with the latest sources there are no more problems with frames when there is no using namespace Upp. Thanks!

Klugier
