Subject: Problems when using layouts
Posted by 281264 on Mon, 28 Jun 2010 20:22:35 GMT
View Forum Message <> Reply to Message

Hi,

steps I have made:

1.- I created a GUI application with the U++, with its own lay out; in this step there is not problem;
2.- The I tried to add a new dialog by using the lay out designer; once I finished its design and used the Generate Code tool to get the class code generated;
3.-I pasted part of the code in a new header file and another part (the implementation) in a cpp file.

When I compile the application the outcome contains some errors. Please, be so kind to indicate what I am doing wrong.

Note: code is attached.

Many thanks,

Javier

File Attachments
1) prueba.7z, downloaded 231 times

Subject: Re: Problems when using layouts
Posted by dolik.rce on Mon, 28 Jun 2010 21:33:28 GMT
View Forum Message <> Reply to Message

Hi Javier,

The problem is that the code in dialogoDlg.h doesn't know about the layout template, because that was included in prueba.h. The solution requires moving the code between the files a bit. I would say the simplest solution (but not only one) is to move the content of dialogoDLG.h into prueba.h and use only this one file.

Additionally, in class prueba you have : dialogoDlg dialogo(); That is interpreted by compiler as function declaration, correctly it shoud be without the parenthesis: dialogoDlg dialogo;

Honza

Subject: Re: Problems when using layouts
Posted by 281264 on Mon, 28 Jun 2010 21:53:33 GMT
View Forum Message <> Reply to Message

Thank you. Your proposal works but I like the idea of having a header file per class. So, is there another way of doing it? I reckon that for large applications there should be other solutions.

Subject: Re: Problems when using layouts
Posted by sergeynikitin on Mon, 28 Jun 2010 23:46:06 GMT
View Forum Message <> Reply to Message

In attachment - working code.
If you have tool for compare source file - compare line by line

Some comments:

In C++ isolation - by modules, because combine header files and combine layout files OR split header files and split layout files.
(I split dialogo and main window into different files)

Use Button variable name ok - and change from CtrlLayout to CtrlLayoutOK - logic of work added automaticaly.
(I change variable name from Cerrar to ok - and method Close is gone...)

Don't forget add
#define LAYOUTFILE <prueba/dialogo.lay>
#include <CtrlCore/lay.h> for every dialog
and for every class.

File Attachments

1) prueba_new.7z, downloaded 233 times

Subject: Re: Problems when using layouts
Posted by 281264 on Tue, 29 Jun 2010 15:04:55 GMT
View Forum Message <> Reply to Message

Sergey,

Thanks a lot. I tried your proposal out and I have some question:

chooses not to do anything), the multiplication in the TopWindow is performed (which is

CtrlLayoutOKCancel: what is actually doing this function? How to control whether the user has pressed ok or cancel and, therefore, actuate accordingly in the TopWindow?

2.- So the conclusion is that it is better (from the stand point of C++ modularity) to have different

3.- What is this?:

Result<<=(double)~dialogo.entrada_a*(double)~dialogo.entrada_b;

What is the role of ~ in the expression?

Thanks,

Javier

Subject: Re: Problems when using layouts
Posted by dolik.rce on Tue, 29 Jun 2010 15:36:56 GMT

View Forum Message <> Reply to Message

281264 wrote on Tue, 29 June 2010 17:042.- So the conclusion is that it is better (from the stand

respective dialogs.
I would say it is a matter of taste For example if you look at theide, which is pretty complex application, you'll see that there is only single layout file included in ide.h. The separation of modules is used only for .cpp files.

281264 wrote on Tue, 29 June 2010 17:043.- What is this?:

Result<<=(double)~dialogo.entrada_a*(double)~dialogo.entrada_b;

What is the role of ~ in the expression?

In Ctrl, operator~() is used as a shorthand for GetData() and operator<<=(Value v) for SetData(Value v). It can save you some typing and increase readability of the code.

Honza

Subject: Re: Problems when using layouts
Posted by mrjt on Tue, 29 Jun 2010 16:27:07 GMT

View Forum Message <> Reply to Message

1- The code is incorrect, instead of

```
 dialogo.Execute();
 Result <<= (int)~dialogo.entrada_a * (int)~dialogo.entrada_b;
```

it should be:

```
 if (dialogo.Execute() == IDOK)
```

```
   Result <<= (int)~dialogo.entrada_a * (int)~dialogo.entrada_b;
```
or:
```
 if (dialogo.ExecuteOK())
   Result <<= (int)~dialogo.entrada_a * (int)~dialogo.entrada_b;
```
so that the cancel response is filtered out.

CtrlLayoutOKCancel does 2 things that CtrlLayout doesn't:
- Assign standard OS icons to buttons (for X11 really). This uses the Button::OK() or Button::Cancel() functions.
- Give each button a callback that breaks the dialog's modal loop with a specific value (like IDOK). This value is then returned by Execute so that you can check which button was pushed (as above).

Using this method there is also additional behaviour when OK or Cancel is pushed by the user. OK calls Accept which validates all the ctrls on the dialog and stops the operation if any fail. Cancel calls Reject (which is used by some controls to end editing states) and if you have called Backup at any point it also calls Restore to revert to backed-up values.

Basically the idea is that by using CtrlLayoutOKCancel you get standard functionality automatically. Sometimes you want to over-ride the default OK handling, but this is easy to:

```
...
ok <<= THISBACK(OnOK);

void OnOK()
{
  if (!Accept())
    return;
  // Do something
  ...

  Break(IDOK);
}
```

Subject: Re: Problems when using layouts
Posted by sergeynikitin on Tue, 29 Jun 2010 17:19:15 GMT

View Forum Message <> Reply to Message

dolik.rce wrote on Tue, 29 June 2010 19:36281264 wrote on Tue, 29 June 2010 17:042.- So the

I would say it is a matter of taste For example if you look at theide, which is pretty complex application, you'll see that there is only single layout file included in ide.h. The separation of modules is used only for .cpp files.

281264 wrote on Tue, 29 June 2010 17:043.- What is this?:

Result<<=(double)~dialogo.entrada_a*(double)~dialogo.entrada_b;

What is the role of ~ in the expression?

In Ctrl, operator~() is used as a shorthand for GetData() and operator<<=(Value v) for SetData(Value v). It can save you some typing and increase readability of the code.

Honza

I say
Quote:In C++ isolation - by modules, because combine header files and combine layout files OR split header files and split layout files.
(I split dialogo and main window into different files)

Subject: Re: Problems when using layouts
Posted by sergeynikitin on Tue, 29 Jun 2010 17:28:25 GMT
View Forum Message <> Reply to Message

Read it:
http://www.ultimatepp.org/www$uppweb$overview$en-us.html

http://www.ultimatepp.org/srcdoc$CtrlLib$Tutorial$en-us.html

http://www.ultimatepp.org/srcdoc$Core$Tutorial$en-us.html

I read the articles several times a month,
until not understand everything before the last character.

If you're going to program in U + + I strongly recommend to do the same.

By the way thanks to the author of these articles.

And by the way, who is the author of these wonderful texts?

Subject: Re: Problems when using layouts
Posted by 281264 on Tue, 29 Jun 2010 18:28:55 GMT
View Forum Message <> Reply to Message

Cheers,

Javier

Subject: Re: Problems when using layouts
Posted by sergeynikitin on Tue, 29 Jun 2010 19:46:30 GMT
View Forum Message <> Reply to Message

Yes!