
Subject: Programming for Layout Objects
Posted by [InterKnight](#) on Fri, 24 Mar 2006 03:22:25 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello, everyone.

I've been working with U++ for just a day or so now, and I am finally on the way to creating an application for an old friend. I previously didn't think I would get this far with GUI design, but I am glad to see that I was wrong.

My question pertains to objects which can be added to the Layout GUI Editor (for example: buttons, labels, etc).

What I need to know is how to program events for buttons. A simple example would be: How would I program the "Cancel" button so that it exits the program?

I've learned how to create menus for applications (with File, View, Help, etc), but I am not yet sure exactly how to implement options.

So, just for simplicity sake, how would I add in code to tell the program that the "Cancel" button exits the program?

Thanks greatly for any help/suggestions.

Take care.

Subject: Re: Programming for Layout Objects
Posted by [mirek](#) on Fri, 24 Mar 2006 08:45:34 GMT
[View Forum Message](#) <> [Reply to Message](#)

InterKnight wrote on Thu, 23 March 2006 22:22Hello, everyone.

I've been working with U++ for just a day or so now, and I am finally on the way to creating an application for an old friend. I previously didn't think I would get this far with GUI design, but I am glad to see that I was wrong.

My question pertains to objects which can be added to the Layout GUI Editor (for example: buttons, labels, etc).

What I need to know is how to program events for buttons. A simple example would be: How would I program the "Cancel" button so that it exits the program?

I've learned how to create menus for applications (with File, View, Help, etc), but I am not yet sure exactly how to implement options.

So, just for simplicity sake, how would I add in code to tell the program that the "Cancel" button

exits the program?

Thanks greatly for any help/suggestions.

Take care.

There are some very simple basic things about widgets:

If you have widget "foo": (e.g. `TextField foo;`)

* to assign it a value, use

`foo <= value;`

* to retrieve its value, use

`value = ~foo;`

* to add default event, which is invoked when user changes the state of widget:

`foo = callback(...)`

Now those "callback(...)" can vary depending on event, however in the most common case, you will have your widget inside some dialog object class and then it will look like:

```
struct Dialog {
    Button cancel;

    void Cancel();

    typedef Dialog CLASSNAME;

    Dialog() {
        cancel <= THISBACK(Cancel);
    }
};
```

Note that as of U++ 602, there is Assist++ THISBACKs... and Layout code generator.. functions that can help you to generate some of that code...

Mirek

Subject: Re: Programming for Layout Objects
Posted by [InterKnight](#) on Sat, 25 Mar 2006 05:17:10 GMT

I'm still not exactly sure what I am doing. I am sorry...very new to GUI design methods. I am going to attach my layout, header, and main.cpp files that show where I am at. I used the layout designer to create the buttons, textboxes, labels, etc, but I don't know how to do the coding for them yet. It will take some work to get the Next Entry>> and the Done buttons to do what I need them to do (add input information to an array of class MedicalData for NextEntry>> and send all input information to a formatted .rtf file formatted into columns). I think that if I start off simple by seeing how to program the Cancel button, I can go from there.

This is my first GUI application, and here is a bit of background on it:

A close friend of mine has Type-2 diabetes. He is in his late 50s and has poor sight. He has to put information into a text file (date, blood pressure, etc) daily. He knows very little about how to use computers, and he manually formats his text file by spacing out his columns and everything else along those lines. It takes him hours to do what he needs to do. I want to write this program so that he can just fill in the information for the labels (Date, Blood pressure, etc), click Next Entry>> (if he has more than one day's worth of information), or just Cancel out of the application. What I want to do is create a class called MedicalData which will be declared as an array (MedicalData dataArray[x] which will store the values for each entry. When the Done button is clicked, all of those entries would be sent out to an .rtf file and formatted nicely into columns. I think this would make life a lot easier for my friend (he is like a father to me), and I want to do what I can...so this is my first project and I think it will be a milestone for me.

Any suggestions?

Thanks for the help.

The layout editor is very amazing!

Take care.

PS: All source files are in a 7-Zip (.7z) archive.

File Attachments

1) [Jims_Medical.7z](#), downloaded 2325 times

Subject: Re: Programming for Layout Objects
Posted by [mirek](#) on Sat, 25 Mar 2006 09:35:34 GMT
[View Forum Message](#) <> [Reply to Message](#)

Ok, here we go:

The only trouble with your code so far is that you have put definitions of your methods to .h file. Save_tot_Text, FileMenu, About, HelpMeny should be in .cpp

Now about that Cancel button:

Add Cancel method to your class:

```
class Jims_Medical : public WithJims_MedicalLayout<TopWindow> {  
....  
    void Cancel();  
....  
};
```

define it e.g. like this (in .cpp):

```
void Jims_Medical::Cancel()  
{  
    if(PromptYesNo("Do you want to exit?"))  
        Break();  
}
```

and add this line to your constructor:

```
Jims_Medical::Jims_Medical()  
{  
.....  
    bttncancel <=<= THISBACK(Cancel);  
.....  
}
```

To read the value, use

```
void Jims_Medical::Next()  
{  
    int blood_pressure = ~txtbld_press;  
    PromptOK(AsString(blood_pressure));  
}
```

(Connect this Next to bttnext_ent the same way as Cancel to bttncancel).

Mirek

Subject: Re: Programming for Layout Objects
Posted by [gprentice](#) on Thu, 30 Mar 2006 11:10:24 GMT
[View Forum Message](#) <> [Reply to Message](#)

luzr wrote on Fri, 24 March 2006 20:45InterKnight wrote on Thu, 23 March 2006 22:22Hello, everyone.

I've been working with U++ for just a day or so now, and I am finally on the way to creating an application for an old friend. I previously didn't think I would get this far with GUI design, but I am glad to see that I was wrong.

My question pertains to objects which can be added to the Layout GUI Editor (for example: buttons, labels, etc).

What I need to know is how to program events for buttons. A simple example would be: How would I program the "Cancel" button so that it exits the program?

I've learned how to create menus for applications (with File, View, Help, etc), but I am not yet sure exactly how to implement options.

So, just for simplicity sake, how would I add in code to tell the program that the "Cancel" button exits the program?

Thanks greatly for any help/suggestions.

Take care.

There are some very simple basic things about widgets:

If you have widget "foo": (e.g. `EditField foo;`)

* to assign it a value, use

`foo <<= value;`

* to retrieve its value, use

`value = ~foo;`

* to add default event, which is invoked when user changes the state of widget:

`foo = callback(...)`

Now those "callback(...)" can vary depending on event, however in the most common case, you will have your widget inside some dialog object class and then it will look like:

```
struct Dialog {
```

```

Button cancel;

void Cancel();

typedef Dialog CLASSNAME;

Dialog() {
    cancel <<= THISBACK(Cancel);
}
};

```

Note that as of U++ 602, there is Assist++ THISBACKs... and Layout code generator.. functions that can help you to generate some of that code...

Mirek

Mirek - you have written above that <<= is used to assign foo a value - but the Button widget uses <<= to assign a callback.

On the "whetting your appetite" web page you have this

```

Days::Days()
{
    CtrlLayout(*this, "Days");
    date1 <<= THISBACK(Compute);
    date2 <<= THISBACK(Compute);
}

```

where date1 is an EditDate.

How do you find out which operators do what for a widget.
e.g. where is the <<= operator for an EditDate declared.

Graeme

Subject: Re: Programming for Layout Objects
 Posted by [gprentice](#) on Thu, 30 Mar 2006 11:26:43 GMT

Quote:

How do you find out which operators do what for a widget.
e.g. where is the <<= operator for an EditDate declared.

With the help of slickedit I see it's in the Ctrl class with overloads that take either Value or Callback.

It's not all that easy to find if it's re-defined in a derived class - for some reason, slickedit won't find me the definition of <<= when I use it in an expression e.g. EditDate d1; d1 <<= THISBACK(something);

Does Ultimate++ impressive code browsing facility allow me to jump to definition of <<= somehow, or to find it ??

Graeme

Subject: Re: Programming for Layout Objects
Posted by [mirek](#) on Thu, 30 Mar 2006 19:51:44 GMT

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

gprentice wrote on Thu, 30 March 2006 06:26Quote:
How do you find out which operators do what for a widget.
e.g. where is the <<= operator for an EditDate declared.

With the help of slickedit I see it's in the Ctrl class with overloads that take either Value or Callback.

It's not all that easy to find if it's re-defined in a derived class - for some reason, slickedit won't find me the definition of <<= when I use it in an expression e.g. EditDate d1; d1 <<= THISBACK(something);

Does Ultimate++ impressive code browsing facility allow me to jump to definition of <<= somehow, or to find it ??

Graeme

Not yet, sorry...

Mirek

Subject: Re: Programming for Layout Objects
Posted by [gprentice](#) on Thu, 30 Mar 2006 23:01:05 GMT
[View Forum Message](#) <> [Reply to Message](#)

luzr wrote on Fri, 31 March 2006 07:51gprentice wrote on Thu, 30 March 2006 06:26Quote:
How do you find out which operators do what for a widget.
e.g. where is the <<= operator for an EditDate declared.

With the help of slickedit I see it's in the Ctrl class with overloads that take either Value or Callback.

It's not all that easy to find if it's re-defined in a derived class - for some reason, slickedit won't find me the definition of <<= when I use it in an expression e.g. EditDate d1; d1 <<= THISBACK(something);

Does Ultimate++ impressive code browsing facility allow me to jump to definition of <<= somehow, or to find it ??

Graeme

Not yet, sorry...

Mirek

Well I found a way! Query for operator<<= and it finds them all, including the ones in ::Ctrl. Not bad! (Slickedit tagging doesn't handle operator functions - I guess they're a bit of a problem).

Graeme