Subject: Changing EditFields from header files
Posted by nlneilson on Thu, 29 Nov 2012 11:21:54 GMT
View Forum Message <> Reply to Message

How do you change the EditFields content from different header file?

In java it is something like this to get access:
Server srv = new Server(11811);
TrackLayer tlayer = new TrackLayer(wwd);

I have several apps created using U++ and combined them.
There is about 2300 lines of code plus 5 header files that takes and returns data.

I would like to be able to change the data in the EditFields from header files.

I thought it might be easier if the GUI was in a header file.

Here is a test app that only has one EditField in H1.h but cannot change the data from H2.h

File Attachments
1) Test2.rar, downloaded 370 times

Subject: Re: Changing EditFields from header files
Posted by navi on Thu, 29 Nov 2012 12:56:11 GMT
View Forum Message <> Reply to Message

the object t1 is within the class scope of H1. and it is a private member so access is restricted only within the class. if data need to be set from outside the class you have to implement a interface for that. usually through a public function like
void Set_t1(int a){ t1<=a; }
option two is to make the global function AddNum2() to return the result as string and set the return value of the function to t1 in H1() contractor. see the example in the attachment.
its nothing to do with U++. you need to read up on C++ scope of variables and specially class scope.

File Attachments
1) Test2.rar, downloaded 397 times

Subject: Re: Changing EditFields from header files
Posted by nlneilson on Thu, 29 Nov 2012 18:45:27 GMT
View Forum Message <> Reply to Message

Thanks navi for the response

Quote:you need to read up on C++ scope of variables and specially class scope.

Yes I do. I have C++ Language Tutorial from cplusplus.com
If you know something better let me know.

Quote:its nothing to do with U++
How U++ handles EditFields may be different from M$VC, which I don't know very well either.

My post apparently was not as clear as it should have been but this may relate to it:
Quote:if data need to be set from outside the class you have to implement a interface for that.
usually through a public function like
void Set_t1(int a){ t1<=a; }
With this in H1.h if it can be called from H2.h it may work.

In java with H1 h1 = new H1(String);
then in H2 it would just be h1.t1<<= "2 OK";

I tried to make t1 global but that did not work.

How you modified the code:
in H1.h
void Set_t1(int a){ t1<<=a; }
in H2.h instead of
return AsString("2 OK");

maybe have
H1.h
void Set_t1(String a){ t1<<=a; }
H2.h
if (a > b) Set_t1("2 OK");

but get this error
H2.h(16) : error C3861: 'Set_t1': identifier not found
but in H1.h it is under public:

To clarify this is in GPS receiver code in a separate thread so there will be no 'return' but up to five EditFields will be changed but it needs to be done in that thread.

I cleaned up the code.
The t1 or the Set_t1(...) needs to be accessed in H2.h

File Attachments

1) Test2.zip, downloaded 357 times

[quote title=nlneilson wrote on Thu, 29 November 2012
maybe have
H1.h
void Set_t1(String a){ t1<<=a; }
H2.h
if (a > b) Set_t1("2 OK");

but get this error
H2.h(16) : error C3861: 'Set_t1': identifier not found
but in H1.h it is under public:
[/quote]

its not a valid scope to call Set_t1() this way. Set_t1 is in H1 Class scope so you need to call it using a object such as my_h1.Set_1();
but your H1 object is bound in main function and not accessible from global scope, since its only exist inside the main function.

```
// in h1.h file
struct H1 : public WithTest2Layout<TopWindow> {
public:
 typedef H1 CLASSNAME;
 H1();
 void AddNum(int a, int b){
  if(a>=b)
   t1<<="1 OK";
 }

       void AddNum2(int a, int b);
 void Set_t1(String a) {t1<<=a;}
};


// in h2.h file

#include "h1.h"

// it is now valid to call Set_t1 as AddNum2 is now also in the
// same class scope.

void H1::AddNum2(int a, int b){
 if ( a > b )
//  t1<<="2 OK";
  Set_t1("2 OK");
```

}

Subject: Re: Changing EditFields from header files
Posted by nlneilson on Thu, 29 Nov 2012 20:40:03 GMT
View Forum Message <> Reply to Message

I made the 2 changes and now get this:

c:\myapps\test2\H2.h(9) : error C2653: 'H1' : is not a class or namespace name
c:\myapps\test2\H2.h(13) : error C3861: 'Set_t1': identifier not found

Subject: Re: Changing EditFields from header files
Posted by nlneilson on Thu, 29 Nov 2012 21:26:49 GMT
View Forum Message <> Reply to Message

In the first post the Test2.rar has a commented line I tried that also gave the:
// H1::t1 <<= "2 OK"; // error: 'H1' : is not a class or namespace name

I am on Win7 64bit running U++ 5556
On Win7 32bit and U++ 5584 the problems are the same.

Same with U++ 5592
I may have overlooked something so here it is with the last changes.

File Attachments
1) Test2 (3).zip, downloaded 375 times

Subject: Re: Changing EditFields from header files
Posted by navi on Fri, 30 Nov 2012 04:49:44 GMT
View Forum Message <> Reply to Message

Quote:
H1::t1 <<= "2 OK"; // error: 'H1' : is not a class or namespace name

in C++ this is not how one access non static class members. its like this

```
H1 a;
a.t1 = "something";
```

Quote:
I am on Win7 64bit running U++ 5556
On Win7 32bit and U++ 5584 the problems are the same.

Same with U++ 5592
I may have overlooked something so here it is with the last changes.

again, these error has nothing to do with U++. they are all valid compiler errors for your wrong code.

Quote:I made the 2 changes and now get this:

c:\myapps\test2\H2.h(9) : error C2653: 'H1' : is not a class or namespace name
c:\myapps\test2\H2.h(13) : error C3861: 'Set_t1': identifier not found

H1::t1 <<= "2 OK"; // error: 'H1' : is not a class or namespace name
all the errors are scope related. if you right your class in plain c++ and use t1 as string variable and use cout to print out. you will still get the same errors. because they are from C++ compiler and has nothing to do with U++ or your Windows version.

```
class H1{
    public:
          int t1; // t1 is in the class h1's scope
          // this function also in h1's scope
          void AddFun1(int a, int b)
          {
             t1=a+b; // valid t1 access.

           }

};

// regardless weather the below function is in
// h1.h or h2.h or main.cpp or anyother file in
// that matter is still in global scope

void AddFun2(int a, int b)
{
   // this is function scope
  // t1 is not visible here
  // no object of H1 class also is non existent
  // since you did not create any.
```

```
  // but you can do this:

  H1 c;
  c.AddFun1(1,1);
  cout<<c.t1;

  c.t1=a+b;
  cout<<c.t1;

  // all t1 of c object of H1 class are valid but c gets
  // destroyed as soon as function scope is finish.


}
```

Subject: Re: Changing EditFields from header files
Posted by nlneilson on Fri, 30 Nov 2012 11:31:42 GMT
View Forum Message <> Reply to Message

I appreciate your time explaining this but I am confused.
I just get errors.
Passing arguments and changing a global is one thing.

I made Test4 and simplified the code.
No C++ variables are passed or returned except the String that will be placed in the EditField t1.

H2.h has this, removing any of the // lines just throws errors.

```
//H1 c;
void Try2(){
//  t1<<="2 OK";
//  c.Set_t1("H2 OK");
  ;
}
```

t1 is the U++ name for the U++ EditField

Could you modify the code so it works?
Comments in the code would be appreciated.

Then maybe I can understand.

File Attachments
1) Test4.zip, downloaded 355 times

Subject: Re: Changing EditFields from header files
Posted by navi on Fri, 30 Nov 2012 12:22:10 GMT
View Forum Message <> Reply to Message

Quote:No C++ variables are passed or returned except the String that will be placed in the EditField t1.
what do you mean?

Again, you CAN NOT access t1 from Try2(). C++ Scope does not work that way. you have to pass or return variable in order to set value into t1 from Try2(). There is no clean way around it.

And trust me it was the very reason why C++ makers created the 'class scope'! To protect the class members.

Attached modified code has example of how t1 can be modified from functions from h2.h without bending or breaking C++ rules.

File Attachments
1) Test4 Modified.rar, downloaded 347 times

Subject: Re: Changing EditFields from header files
Posted by nlneilson on Fri, 30 Nov 2012 13:28:19 GMT
View Forum Message <> Reply to Message

Thanks navi

I tried the code and just get a pop up that says:
Assertion failed in C:\upp\uppsrc\CtrlCore\Ctrl.cpp, line 573
Gui widgets cannot be global variables

This is with both line 18 and 19 with each commemted one at a time.

Passing a String FROM H2.h will be fine into something like you suggested.
// void Set_t1(String a) {t1<<=a;}

edit: After Clean UPPOUT, CLEAN and rebuild it works with MSC9 optimal. Debug throws the error.
Same with MSC10, optimal OK and debug error

Much Thanks! That gives me something to work with.

---

nuvi:

```
in H1
    Try2(*this);

in H2
H1 a;
void Try2(H1& a){
    a.Set_t1("H2 OK");
}
```

That worked.

In a thread it could be:

```
    a.Set_t1("H2 OK");
    Sleep(1000);
    a.Set_t1("H2 OK");
    Sleep(1000);
    ...
```

The String in a.Set_t1("H2 OK"); would actually be a GPS NMEA sentence.
Fine except if it cannot be run in U++ Debug then it's a NoGo.

navi wrote on Thu, 29 November 2012 11:34

```
// in h2.h file

#include "h1.h"

// it is now valid to call Set_t1 as AddNum2 is now also in the
// same class scope.

void H1::AddNum2(int a, int b){
 if ( a > b )
//  t1<<="2 OK";
  Set_t1("2 OK");

}
```

I tried several variations of this.

```
H2.h
#include "H1.h"
void H1::Try2(){
    Set_t1("H2 OK");
```

}
but in H1.h
//#include "H2.h" // this must be commented out

The t1 EditField shows "H2 OK" when run in Debug.

Thanks again, I have more than 20 hours tinkering with this.

Subject: Re: Changing EditFields from header files
Posted by navi on Fri, 30 Nov 2012 20:21:51 GMT
View Forum Message <> Reply to Message

nlneilson wrote on Fri, 30 November 2012 18:35nuvi,

In a thread it could be:
```
a.Set_t1("H2 OK");
Sleep(1000);
a.Set_t1("H2 OK");
Sleep(1000);
...
```

I dont know if u r talking about U++ Thread class or not. if its U++ threads, then it is suggested not to change GUI from threads without proper care, GUI_Lock and PostCallback(). as chenging GUI from threads otherwise will end up in Gui Deadlock.

See the follwoings in example section:
GuiLock : This package demonstrates the use of Ctrl::Lock in MT applications
GuiMT : Using event queue for communication between worker threads and GUI

Quote:
The String in a.Set_t1("H2 OK"); would actually be a GPS NMEA sentence.
Fine except if it cannot be run in U++ Debug then it's a NoGo.

dont know what u mean.

Subject: Re: Changing EditFields from header files
Posted by nlneilson on Fri, 30 Nov 2012 21:05:09 GMT
View Forum Message <> Reply to Message

Yes it is the Thread class in U++ and I have to be able to debug the code.

I have five threads but three is about all that is open at one time.
edit: There is actually another thread that sends data from the C++ app to the java display app through a socket.

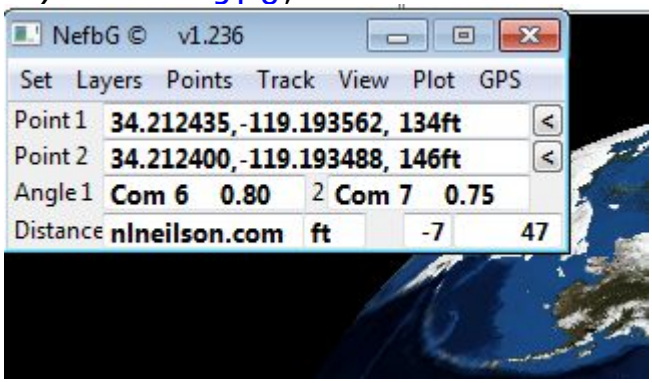
I have 5 header files now without problems.
I want to have the GPS receiver code in their own header files as my main .ccp file is over 2300 lines.

I have not had a GUI_Lock or GUI Deadlock AFAIK.
There is a lot of error handling and try-catch blocks.

Here are two GPS receivers running right now over wireless.
Here I have run the NEMA sentence through a Parser which is in a header file. This is actually tracking on the Globe also.
Each Points1 and Points2 update every second independently. Angle1 and 2 also update but visually only noticeable when the signal quality changes. The lower right is just a counter. So there are 5 boxes updating every second.

File Attachments

1) 2-GPS.jpg, downloaded 1018 times



Subject: Re: Changing EditFields from header files
Posted by nlneilson on Fri, 30 Nov 2012 22:37:56 GMT
View Forum Message <> Reply to Message

I will look at the GuiLock and GuiMT links, thanks.

Going back through the Test2 and the code before that my biggest problem was not commenting out
#include "H2.h" in H1.h.

As you mentioned in Test4 Modified.rar
// must not include h2.h here as then it will have Try3(H1&) deffination
// befor H1's deffination hanse producing an scope error.

// invalid indentifier or somthing like that...

I probably would not have figured that out.
