Subject: how to communicate between windows?
Posted by bonami on Tue, 02 Nov 2010 09:18:55 GMT
View Forum Message <> Reply to Message

I have two classes inherited from TopWindow.
class 1 need to tell class 2 to go back or forward in processing.
they OpenMain() then I use Ctrl::EventLoop().

Solution 1,
  class 1 envoke a member of class 2. class 2 needs a private lock. I would rather not use this solution.

Solution 2,
  From MS' view, class 1 can SendMessage() or send an event to class 2. How to achieve this in U++? How to change class 2 (TopWindow)'s processing?

Thank you.

Subject: Re: how to communicate between windows?
Posted by mrjt on Tue, 02 Nov 2010 10:29:37 GMT
View Forum Message <> Reply to Message

Is this a threading problem?

How about a simple event interface:

```
class EventHandler
{

private:
 static Vector<EventHandler *> clients;
public:
 typedef enum { SOME_EVENT, ANOTHER_EVENT } EventType;
public:
 EventHandler() {
  clients.Add(this);
 }

 virtual ~EventHandler() {
  for (int i = 0; i < clients.Getcount(); i++)
   if (clients[i] == this) {
    clients[i].Remove(i);
    return;
   }
 }

 static void SendEvent(EventType event, int param1)
```

```
 {
  for (int i = 0; i < clients.GetCount(); i++)
   clients[i]->HandleEvent(event, param1);
 }

 virtual void HandleEvent(EventType event, int param1) { }
};
```

Any class that inherits from EventHandler would be able to recieve global events.

Subject: Re: how to communicate between windows?
Posted by bonami on Wed, 03 Nov 2010 01:45:02 GMT
View Forum Message <> Reply to Message

My question is exactly how to implement this HandleEvent for a TopWindow derived class.

Subject: Re: how to communicate between windows?
Posted by mrjt on Wed, 03 Nov 2010 10:22:44 GMT
View Forum Message <> Reply to Message

Then I don't understand the question. 'How to change class 2 (TopWindow)'s processing?' is too vague a question.

And you haven't answered whether this is a threading problem or not.

You will need to provide substantially more information to get any sort of useful help.

Subject: Re: how to communicate between windows?
Posted by bonami on Thu, 04 Nov 2010 03:29:22 GMT
View Forum Message <> Reply to Message

sorry i did not make it clear, since i thought it is simple.
here's my case, i have two TopWindows A & B. If user click a button in A, B shows something. If user click the other button in A, B shows something else. I need TopWindow::OpenMain(), then I donno whether it is multi-threaded. Anyway, my real case is more complicated and it IS multi-threaded and I want the code below is multi-thread compatible, too.

```
class B : public TopWindow
...

class A : public TopWindow
{
  B b;
  void shown() { b.OpenMain(); }
```

```
  Button ButA;
  Button ButB;
  void Button_A();
  void Button_B();
...

GUI_APP_MAIN
{
  A a;
  a.OpenMain();
  a.shown();
  Ctrl::EventLoop();
}
```

in Button_A() or _B(), A can tell B about which button is clicked, such as setting a flag. But how can B notice this? If i derive A & B from your EventHandler, how can I implement B's HandleEvent? This IS a threading problem. Maybe in a word, it is how to add my own procedure in TopWindow's main thread processing.
Thank you.

Subject: Re: how to communicate between windows?
Posted by andrei_natanael on Thu, 04 Nov 2010 07:33:36 GMT
View Forum Message <> Reply to Message

Hello Bonami,

Here is your example, modified to notify other window about a event. I've used PostCallback to do that, no MT

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

using namespace Upp;

class B : public TopWindow
{
 public:
 B()
 {
  SetRect(220, 10, 200, 200);
  Add(l.VCenterPos(30).HCenterPos(180));
  l.SetText("Test");
 }
 Label l;
 void ShowMessage(const String& msg)
 {
  l.SetText(msg);
 }
```

```
};

class A : public TopWindow
{
  typedef A CLASSNAME;
  public:
  A()
  {
    SetRect(10, 10, 200, 200);
    ButA.SetLabel("Button A");
    ButB.SetLabel("Button B");
    Add(ButA.LeftPos(5, 80).TopPos(5, 25));
    Add(ButB.LeftPos(5, 80).TopPos(35, 25));
    ButA <<= THISBACK(Button_A);
    ButB <<= THISBACK(Button_B);
  }
  B b;
  void shown() { b.OpenMain(); }
  Button ButA;
  Button ButB;
  void Button_A()
  {
 b.PostCallback(callback1(&b, &B::ShowMessage, "Button A pressed"));
  }
  void Button_B()
  {
   b.PostCallback(callback1(&b, &B::ShowMessage, "Button B pressed"));
  }
};

GUI_APP_MAIN
{
  A a;
  a.OpenMain();
  a.shown();
  Ctrl::EventLoop();
}
```

Andrei

Subject: Re: how to communicate between windows?
Posted by bonami on Thu, 04 Nov 2010 09:27:02 GMT
View Forum Message <> Reply to Message

It looks like the thing I wanted. This callback will be executed in class b's thread/execution, right?
my code generates error LNK2019: unresolved external symbol "public: void __thiscall

B::set(enum B::tp)"

```
class B: public TopWindow {
public:
  enum tp
  {
    TPA,
    TPB
  };
  void set(enum tp);
  ...
}
class A: public TopWindow {
...
private:
  B *b;
  void Button_A()
  {
 b->PostCallback(callback1(b, &B::set, B::TPA));
  }
...
}
```

Subject: Re: how to communicate between windows?
Posted by andrei_natanael on Thu, 04 Nov 2010 20:47:48 GMT
View Forum Message <> Reply to Message

In class B set function signature should be:
void set(tp x) // note, no enum before tp

Quote:This callback will be executed in class b's thread/execution, right?
Yes.

Andrei

Subject: Re: how to communicate between windows?
Posted by bonami on Fri, 05 Nov 2010 02:05:35 GMT
View Forum Message <> Reply to Message

all right. with or without enum have no differences. it compiles right. i just did not provide set()'s implementation. can i be more stupid?
I think b->PostCallback() and ::PostCallback() are same. right?
thank you very much.

Subject: Re: how to communicate between windows?
Posted by andrei_natanael on Fri, 05 Nov 2010 05:16:58 GMT
View Forum Message <> Reply to Message

They are same. Only difference is that ::PostCallback get executed 1 ms later than b->PostCallback (that's what code say)