Subject: Re: What Label.WhenAction supposed to do?
Posted by forlano on Wed, 22 Mar 2006 16:24:51 GMT

View Forum Message <> Reply to Message

Hello,

the code you have posted was very important for me. I'am new of C++ and Ultimate++, but I've some experience with GUI using C API (Motif for example under Unix). Your code reminded me more or less what I've usually done in C style without to declare any class. Just each widget is child of someother and there are callback attached to them. I understand that the class method is elegant and powerful, but I'm a lazy man, , and I do not want to loose time with C++ and its intricasy (at least not now). So your approach for me was a surprise and showed that, perhaps, I can program in a sort of C style.
I would like to know, before to start to port one my program in U++ (the wich GUI is in C), if the approach you have showed could break at some moment, for example in complex interface. In other term, is it mandatory to implement class/es to let a GUI interface to work, or can I live without them in Ultimate++?
In case I can avoid the class I would like to see your code, or a similar one, in the documentation to let know to the beginner how it is easy to program with Ultimate++ in contrast with other GUI toolkits.

Luigi

fudadmin wrote on Mon, 20 March 2006 02:01

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

void testAction()
{ PromptOK("testLabelAction");
}

GUI_APP_MAIN
{
 TopWindow w;
 Button b;
 Label l;

 w.Add(b);

 b.SetLabel("button - testAction");
 b.SetPos(b.PosLeft(10, 100), b.PosTop(10, 30));
// b.WhenPush=callback(testAction);  //both work
 b.WhenAction=callback(testAction);

 w.Add(l);
 l.SetLabel("label - testAction");
 l.SetPos(l.PosLeft(150, 100), l.PosTop(10, 30));
 l.WhenAction=callback(testAction);
 w.Run();
}
```