Subject: Re: Strange behaviour of my modal dialog
Posted by forlano on Tue, 25 Apr 2006 21:33:34 GMT
View Forum Message <> Reply to Message

luzr wrote on Tue, 25 April 2006 23:10Maybe I did not get it completely, but I guess the mistake you have done is that you think that "<<=" somewhat "binds" the variable with EditField (or other Ctrl).

However, that is not true - it is simple assignment, it sets the value of EditField, but the value of "TD.TOURNAMENT" is further unchanged by the EditField.
I was aware of it. I wanted put in EditField the value TD.TOURNAMENT and nothing else. I wanted to know the current value of TD.TOURNAMENT. Later another method will get what I've written in the EditField and then update TD.TOURNAMENT. My surprise is that TD.TOURNAMENT that finish in the EditField is the very old one. It seems the widget is not aware that TD.TOURNAMENT has already changed at some moment.

Quote:
 If you need to have it changed, you have to perform "backward" assignment like

TD.NAMETONEO = ~newt.editTourn;

Of course, doing so for each variable would be boring, so U++ gives you better options.

First of all, you can consider whether you really need to have separate variable for tournament name. Sometimes you can simply use the dialog to store data and do not store to variable at all. (Data are stored in widgets regardless it is "open").

This is very interisting! I was not aware of it. In fact I get tired to use a parallel structure to store variables (I was afraid to loose them when the widget disappeared!)
Quote:Another option is to use "CtrlRetriever". This provides a kind of binding you perhaps have expected:

CtrlRetriever r;
r(newt.editTorneo, TD.TORNEO)
 (newt.anotherField, TD.ANOTHER)
....
;
newt.Run();
r.Retrieve();

First past, using operator(), simply performs

ctrl <<= val;

and stores references to both to the list. Then Retrieve method goes through those references and performs

val = ~ctrl;

Primitive, but effective

Mirek

I'm going to try it!

Luigi