Subject: Custom check for EditIntXxxx controls
Posted by Mindtraveller on Tue, 16 Oct 2012 11:32:20 GMT
View Forum Message <> Reply to Message

Hello! Sorry if my question is noobish, but I failed to find how to achieve what I need - add "checking ability" to a number of different controls (like EditIntSpin). This means that when user edits the value, the control is coloured red if value is inappropriate (with some calculations involving program logic from different classes).

If I instanitate my own classes from EditIntSpin and etc. - how could I switch control to "red" mode? Is there any better ways to achieve it not creating addtional classes?

Thanks in advance!

Subject: Re: Custom check for EditIntXxxx controls
Posted by dolik.rce on Tue, 16 Oct 2012 13:09:41 GMT
View Forum Message <> Reply to Message

Hi Pavel,

Edit*Spin ctrls are unfortunately not templates, so I'm afraid it is not really simple to "inject" them with custom value checking code (aside from the default min/max/notNull constraints).

For Edit* ctrls, you can achieve this simply by writing custom Convert based class and declare the type using typedef. E.g. for EditInt with your custom logic:typedef EditValue<int, MyConvert> MyEditInt;The input will be marked as invalid if your MyConvert class returns ErrorValue (see docs and/or Core/Convert.h). If you also want the spin buttons, you'll probably have to add them yourself, but it should be easy if you follow the code in original EditIntSpin class.

Best regards,
Honza

Subject: Re: Custom check for EditIntXxxx controls
Posted by dolik.rce on Tue, 16 Oct 2012 16:39:49 GMT
View Forum Message <> Reply to Message

I couldn't resist I just had to try to implement the templated version of *Spin ctrls. I proposed it to Mirek for inclusion into CtrlLib, you can see it here: http://www.ultimatepp.org/redmine/issues/336 and of course, you can use it even if Mirek deems it unusable for U++...

Honza

Subject: Re: Custom check for EditIntXxxx controls

Posted by mirek on Fri, 19 Oct 2012 10:44:22 GMT
View Forum Message <> Reply to Message

Generic WithSpin template now available...

Mirek

Subject: Re: Custom check for EditIntXxxx controls
Posted by Mindtraveller on Thu, 25 Oct 2012 06:02:42 GMT
View Forum Message <> Reply to Message

OK, finally a little report on results.
Sorry to say that ErrorValue doesn't work very well here. This happens due to the fact it doesn't convert too well into integer value. So you may return ErrorValue(FormatInt(...)) and it will work. But when user pressses (inc) or (dec) spin buttons, the value is immediately switched to -INT_MAX, and there is no way to change this behaviour from user code.

Concluding my experiments, the best way to make control customly checkable is this:
1) Derive MyCustomConvert from one of Convert classes (e.g. ConvertInt). Make any checks in there.
2) For MyCustomConvert provide reference to target EditField derived class. MyCustomConvert clears/sets error state with EditField::Error().
3) Use any Edit*** controls and set your MyCustomConvert with EditField::SetConvert function() providing reference to your Edit*** control.