Subject: EditMinMax ctrl implementation
Posted by pivica on Wed, 15 Feb 2006 19:52:24 GMT
View Forum Message <> Reply to Message

I have next problem. I'm using EditDouble ctrl with min and max limits. Current implementation of checking min/max boundaries isn't good for me. For example, boundary checking is done when window with EditDouble ctrl is closing. It would be much better to check limits when EditDouble ctrl lost its focus.
Well after couple of hours trying to implement some good solution in CtrlLib\EditField.cpp I give up. Problem is that I couldn't effectivly implement LostFocus() in EditValue class because EditString is inherit from it, too. In the end, here is something I've manage to implement:

```
template <class DataType, class Cv>
class EditMinMax : public EditValue<DataType, Cv> {
public:
 virtual void  LostFocus();

 // rest of class EditMinMax...
};

template <class DataType, class Cv>
void EditMinMax<DataType, Cv>::LostFocus()
{
 DataType dt;

 if (!EditMinMax<DataType, Cv>::text.IsEmpty() && IsNumber(dt)) { // Checking int and double.
  String s = EditMinMax<DataType, Cv>::text.ToString();

  CParser p(s);
  if (p.IsDouble())
   dt = p.ReadDouble(); // gcc warning: converting to `int' from `double'?
  else if (p.IsInt()) // Not doing well for doubles - returns true in 602b version?
   dt = p.ReadInt();

  if (dt < Cv::GetMin())
   s = AsString(Cv::GetMin());
  else if (dt > Cv::GetMax())
   s = AsString(Cv::GetMax());

  WString text1(s);
  EditMinMax<DataType, Cv>::text = text1;
 }

 EditField::LostFocus();
}
```

Anyway I don't like how EditMinMax ctrls are working. My opinion is that it would be much better to check limits when EditMinMax lost its focus. But if this can't be done in U++ level, I guess I will have to implement my own EditMinMax ctrl.

Subject: Re: EditMinMax ctrl implementation
Posted by mirek on Wed, 15 Feb 2006 20:19:06 GMT
View Forum Message <> Reply to Message

Personally, I find idea of always screaming programs not very palatable (it is not unlikely that I notice problem in another field, but I am not allowed to fix it until I fix the current - I do not like this).

Also, think about pressing the Cancel button... (I know, it can be resolved to a degree, OTOH it would not be possible in normal Win32 where pressing button moves the focus). I do not like the idea of not being able to cancel dialog until I fix values in it....

However, I think that it is worth considering adding "CheckOnFocus" or something like that in EditField - it is not that hard to do.

(But you will have to wait, I do not want to compromise new release...)

Mirek

Subject: Re: EditMinMax ctrl implementation
Posted by pivica on Wed, 15 Feb 2006 20:46:14 GMT
View Forum Message <> Reply to Message

> Personally, I find idea of always screaming programs not very palatable (it is not unlikely that I notice problem in another field, but I am not allowed to fix it until I fix the current - I do not like this).

I agree with you. Better would be to allow ctrl to automaticly fix wrong value.
So for example there is EditDouble with min -15 and max 145. If a user enter -20 then on lost focus ctrl should correct value to -15. Also when user enter 1000, on lost focus there would be 145.

Subject: Re: EditMinMax ctrl implementation
Posted by mirek on Wed, 15 Feb 2006 21:12:40 GMT
View Forum Message <> Reply to Message

IMO, this could lead into silently ignored problems...

Mirek

Subject: Re: EditMinMax ctrl implementation
Posted by pivica on Thu, 16 Feb 2006 06:44:39 GMT
View Forum Message <> Reply to Message

Yes, but in some situations it would be nice behavior. Best would be to have something like CheckOnFocus, as you wrote, and with that method to tell control should it fire pop up dialog or auto correct value.

Subject: Re: EditMinMax ctrl implementation
Posted by mirek on Thu, 16 Feb 2006 16:17:51 GMT
View Forum Message <> Reply to Message

Would need changes to Convert (something that did not happen for 5 years). Possible, but I am always reluctant to change core classes...

Meditate about this I will

Mirek

Subject: Re: EditMinMax ctrl implementation
Posted by mirek on Thu, 16 Feb 2006 16:19:15 GMT
View Forum Message <> Reply to Message

BTW, with the right Convert, you should be able to achieve this now (just implement adjusting in Scan).

Mirek

Subject: Re: EditMinMax ctrl implementation
Posted by pivica on Thu, 16 Feb 2006 16:40:50 GMT
View Forum Message <> Reply to Message

Will try..

Subject: Re: EditMinMax ctrl implementation
Posted by hojtsy on Sat, 18 Feb 2006 22:36:35 GMT
View Forum Message <> Reply to Message

Let me add one more viewpoint. I think it is good to show visual clues if some fields contain invalid values, but popup windows are too annoying. How about adding a reddish tint to the background color of EditFields which contain an invalid data? I made the implementation in EditField::Paint, by adding these two lines:

```
if(convert && convert->Scan(text).IsError())
  paper = Blend(paper, Color(255, 0, 0), 32);
```
It shows if the data is valid while you are typing it. I attach the modified file.

File Attachments

1) EditField.zip, downloaded 2218 times

Subject: Re: EditMinMax ctrl implementation
Posted by mirek on Sat, 18 Feb 2006 22:53:03 GMT
View Forum Message <> Reply to Message

Well, not a bad idea either, however, original idea was to "accept" the value(s) when user makes an action to "commit" data - either by pressing "OK" button or e.g. in ArrayCtrl when pressing Enter.

I am not sure whether when pressing "OK", the dialog should just make invalid fields red (also, the current protocol would make just one red...).

However, as an option, why not?

Mirek

Subject: Re: EditMinMax ctrl implementation
Posted by mirek on Sat, 18 Feb 2006 22:54:32 GMT
View Forum Message <> Reply to Message

OOps misinterpreted you. Well, when typing, it would look really nice

Mirek

Subject: Re: EditMinMax ctrl implementation
Posted by hojtsy on Sat, 18 Feb 2006 23:15:45 GMT
View Forum Message <> Reply to Message

luzr wrote on Sat, 18 February 2006 17:54OOps misinterpreted you. Well, when typing, it would look really nice
That's it! Try it out and see what I mean. This is not a replacement for the popup which comes when the window is closed, but an early clue so the user can avoid errors. By the way it does not only work for min-max ranges. Just try out a DateEdit after this modification: if you enter an invalid date it is displayed with red background.

Subject: Re: EditMinMax ctrl implementation
Posted by mirek on Sun, 19 Feb 2006 06:11:43 GMT
View Forum Message <> Reply to Message

Yes, that is how Convert was designed, is not it?

Mirek

Subject: Re: EditMinMax ctrl implementation
Posted by pivica on Sun, 19 Feb 2006 18:46:51 GMT
View Forum Message <> Reply to Message

I like hojtsy idea and I hope that something of all this will be in next upp version:)

Subject: Re: EditMinMax ctrl implementation
Posted by mirek on Sun, 19 Feb 2006 21:56:26 GMT
View Forum Message <> Reply to Message

```
if(convert && convert->Scan(text).IsError())
  paper = Blend(paper, Color(255, 0, 0), 32);
```

is now part of uppsrc. And indeed, it looks very nice.

I just resrve the right to remove it if during next week my users complain about it

Mirek