Subject: [FEATURE] WhenLostFocus in EditField
Posted by koldo on Thu, 27 Jan 2011 13:16:43 GMT
View Forum Message <> Reply to Message

Hello all

When filling fields, I would like to check if the value is valid not when keying letters but just when losing the focus.

To avoid subclassing to use virtual LostFocus(), would it be possible to have a WhenLostFocus in EditField?. Or, is there any other way to validate the value of a field?

Subject: Re: [FEATURE] WhenLostFocus in EditField
Posted by Didier on Thu, 27 Jan 2011 20:07:35 GMT
View Forum Message <> Reply to Message

Hi Koldo,

I think this subject has already been discussed the answers were that you have to to derive the class.
But I completely agree with you. This situation is kind of common so adding WhenFocusLost would be a good thing from my point of view.

So +1

Subject: Re: [FEATURE] WhenLostFocus in EditField
Posted by koldo on Thu, 27 Jan 2011 20:51:02 GMT
View Forum Message <> Reply to Message

Hello Didier

That is it. And the problem is that I would have to subclass DropTime, EditString, EditInt and others just to say the least .

As all of them are a subclass of EditField, adding the callback there would solve it all.

Subject: Re: [FEATURE] WhenLostFocus in EditField
Posted by mirek on Sat, 29 Jan 2011 19:40:50 GMT
View Forum Message <> Reply to Message

koldo wrote on Thu, 27 January 2011 08:16Hello all

When filling fields, I would like to check if the value is valid not when keying letters but just when losing the focus.

To avoid subclassing to use virtual LostFocus(), would it be possible to have a WhenLostFocus in EditField?. Or, is there any other way to validate the value of a field?

There are also quite hand virtuals ChildLostFocus/ChildGotFocus which IME in most cases cover this issue quite well.

Subject: Re: [FEATURE] WhenLostFocus in EditField

Posted by koldo on Sun, 30 Jan 2011 15:49:58 GMT

View Forum Message <> Reply to Message

mirek wrote on Sat, 29 January 2011 20:40koldo wrote on Thu, 27 January 2011 08:16Hello all

When filling fields, I would like to check if the value is valid not when keying letters but just when losing the focus.

To avoid subclassing to use virtual LostFocus(), would it be possible to have a WhenLostFocus in EditField?. Or, is there any other way to validate the value of a field?

There are also quite hand virtuals ChildLostFocus/ChildGotFocus which IME in most cases cover this issue quite well.

Thank you Mirek

For now is valid for me (in fact I used the same checking function for all field changes).

Perhaps in the future something could be included

Subject: Re: [FEATURE] WhenLostFocus in EditField

Posted by kohait00 on Wed, 09 Feb 2011 09:55:18 GMT

View Forum Message <> Reply to Message

Check out WithEnterAction in bazaar...
it's got that.., well almost. it invokes the WhenAction on LostFocus and when EnterKey is pressed.

u simply specify WithEnterAction<EditInt> instead of EditInt..

maybe that helps

Subject: Re: [FEATURE] WhenLostFocus in EditField

Posted by koldo on Wed, 09 Feb 2011 11:44:11 GMT
View Forum Message <> Reply to Message

Hello Kohait00

Yes, although it is a pity to use a new class and losing .usc representation for layout file (or having to create a new that is really a copy of EditString, EditInt, and so on...)

For EditFields and Enter I use the WhenEnter callback. This way WhenAction works always the same.

Subject: Re: [FEATURE] WhenLostFocus in EditField
Posted by dolik.rce on Wed, 09 Feb 2011 13:07:57 GMT
View Forum Message <> Reply to Message

koldo wrote on Wed, 09 February 2011 12:44Yes, although it is a pity to use a new class and losing .usc representation for layout file (or having to create a new that is really a copy of EditString, EditInt, and so on...)
Usc supports templates, so you don't have to lose anything Example can be found in CtrlLib.usc, look for WithDropChoice.

Anyway, I agree that it is unnecessary in this case and that it is much more elegant to use WhenEnter and ChildLostFocus (which is great idea which never occured to me before Mirek mentioned it).

Honza

Subject: Re: [FEATURE] WhenLostFocus in EditField
Posted by koldo on Wed, 09 Feb 2011 17:09:02 GMT
View Forum Message <> Reply to Message

Hello Honza

dolik.rce wrote on Wed, 09 February 2011 14:07Usc supports templates, so you don't have to lose anything Example can be found in CtrlLib.usc, look for WithDropChoice.
You can also create your new .usc with things like this:

```
ctrl MyEditStringLostFocus {
 >EditString;
}
ctrl MyEditIntLostFocus {
 >EditInt;
}
ctrl MyEditDoubleLostFocus {
 >EditDouble;
}
```

...
However this overcrowd layout menu unnecessarily with controls that are almost clone.

dolik.rce wrote on Wed, 09 February 2011 14:07Anyway, I agree that it is unnecessary in this case and that it is much more elegant to use WhenEnter and ChildLostFocus (which is great idea which never occured to me before Mirek mentioned it).
Well, I agree. However it is possible that somebody would prefer something more detailed for a higher modularization degree. In addition, remember that specially when it is computationally expensive to check the values of all fields, with ChildLostFocus you do not know which field has lost the focus.

Subject: Re: [FEATURE] WhenLostFocus in EditField
Posted by koldo on Wed, 16 Feb 2011 17:15:13 GMT

View Forum Message <> Reply to Message

Hello Mirek

I am using ChildLostFocus() but it is called twice per click. Here I enclose you the functions called:

```
1st
LRESULT Ctrl::WindowProc(UINT message, WPARAM wParam, LPARAM lParam) {
 Image Ctrl::DoMouse(int e, Point p, int zd)
  ...
  void EditField::LeftDown(Point p, dword flags)
   bool Ctrl::SetFocus()
    bool Ctrl::SetFocus0(bool activate)
     void Ctrl::DoKillFocus(Ptr<Ctrl> pfocusCtrl, Ptr<Ctrl> nfocusCtrl)
2nd
LRESULT Ctrl::WindowProc(UINT message, WPARAM wParam, LPARAM lParam) {
 void Ctrl::KillFocusWnd()
  void Ctrl::DoKillFocus(Ptr<Ctrl> pfocusCtrl, Ptr<Ctrl> nfocusCtrl)
```

Do you know what I am doing wrong?

Subject: Re: [FEATURE] WhenLostFocus in EditField
Posted by koldo on Wed, 16 Feb 2011 17:32:34 GMT

View Forum Message <> Reply to Message

Ouff, I see the problem

If this, no problem

```
 virtual void ChildLostFocus() {
  a = 27;
 }
```

but with an "Exclamation()" or other operation that changes the focus, this is required

```
 virtual void ChildLostFocus() {
  static bool inside;

  if (inside)
   return;

  inside = true;

  Exclamation("Lost focus");

  inside = false;
 }
```

The problem is that the Exclamation() gets the focus, so function is called twice:
- When the child looses the focus
- When the Exclamation() window gets it