Subject: lostfocus
Posted by djesus on Sun, 29 Oct 2006 10:35:36 GMT
View Forum Message <> Reply to Message

Hi I would like to do any after edit field has a lost focus. In other toolkit there is a event like onlostfocus, but I don`t know if I can do it in ultimate.

Thanks a lot

Subject: Re: lostfocus
Posted by fallingdutch on Sun, 29 Oct 2006 10:47:52 GMT
View Forum Message <> Reply to Message

use the virtual method
virtual Image MouseEvent(int event, Point p, int zdelta, dword keyflags)
where event == MOUSELEAVE, so eg this:

```
Image MyCtrl::MouseEvent(int event, Point p, int zdelta, dword keyflags) {
  if(event == MOUSELEAVE) {
    //do something here
    return NULL;
  }
  else {
     return Ctrl::MouseEvent(event, p, zdelta, keyflags);
  }
};
```

[edit: just saw, that EditCtr has the virtual function "LostFocus" - just use that]

Bas

Subject: Re: lostfocus
Posted by zsolt on Sun, 29 Oct 2006 11:35:24 GMT
View Forum Message <> Reply to Message

You can overload
virtual void ChildGotFocus ()
virtual void ChildLostFocus ()
functions in the container.

Subject: Re: lostfocus
Posted by djesus on Sun, 29 Oct 2006 12:58:12 GMT

Thanks, but if I have in the layout editor a edit named myedit how I can to overload functions.

Sorry for my ignorance

Subject: Re: lostfocus
Posted by mirek on Sun, 29 Oct 2006 19:47:34 GMT

Well, if you are about to overload ChildLostFocus, you do not have to change anything in Layout, just overload it for your dialog class:

```
struct MyDlg : MyLayout<TopWindow> {
   virtual void ChildLostFocus();
};
```

Anyway, the trouble of this method is that at the moment ChildLostFocus is called, focus is already out, so you have hard time to detect whether it is your editfield that lost the focus.

Therefore you will probably have to follow a little bit more complicated path and derive from EditField, overloading LostFocus method.

To have your new widget in Layout, just add it as user class. There is a little catch - class has to be defined when layout file is included. If for some reason you do not want that, you can still have it in the layout, just leave the name of class empty and add a public member to your dialog class with the same name as defined in layout, with proper class type.

Mirek