Subject: Re: BUG? EditInt / PromptOK not functioning as thought?
Posted by mirek on Sat, 24 Jun 2017 14:00:06 GMT

View Forum Message <> Reply to Message

> ptkacz wrote on Tue, 16 May 2017 05:40That's interesting, the Rejector.
>
> I'm guessing that the functionality should really work by popping up the message when the field loses focus? Similar to how the field turns red when validation is applied against it.

I do not think that would be good - as general solution. Sometimes the error is a result of combination of values, so associating it with value of single field is probably not a good idea.

In general, the 'annoying behaviour' is mostly the result default settings.

In U++, you dialog can be 'breaked' (no errors displayed, nothing else is done), 'accepted' (errors are checked and dialog does not exit if there are errors found) or 'rejected' (no errors displayed AND dialog is restored to values before Run). See http://www.ultimatepp.org/srcdoc$CtrlLib$Tutorial$en-us.html ch. 18, 20

Now the problem is with the default settings of WhenClose callback that gets invoked by clicking "window close" [X] button. Default is to call TopWindow::Close virtual method, which is defined this way:

```
void TopWindow::DefaultBreak()
{
 if(FindAction(IDCANCEL) || close_rejects)
  RejectBreak(IDCANCEL);
 else
 if(FindAction(IDNO))
  RejectBreak(IDNO);
 else
 if(FindAction(IDEXIT))
  AcceptBreak(IDEXIT);
 else
 if(FindAction(IDYES))
  AcceptBreak(IDYES);
 else
  AcceptBreak(IDOK);
}
```

That means it reacts on presence of different types of 'breakers' and if there is none defined, defaults to 'accept'. However, fix is simple, all you need is to override Close or set WhenClose to something else, e.g.

```
app.WhenClose = [=] { app.Break(); };
```