
Subject: Re: Problems when using layouts
Posted by [mrjt](#) on Tue, 29 Jun 2010 16:27:07 GMT
[View Forum Message](#) <> [Reply to Message](#)

1- The code is incorrect, instead of
dialogo.Execute();
Result <=<= (int)~dialogo.entrada_a * (int)~dialogo.entrada_b;
it should be:
if (dialogo.Execute() == IDOK)
Result <=<= (int)~dialogo.entrada_a * (int)~dialogo.entrada_b;
or:
if (dialogo.ExecuteOK())
Result <=<= (int)~dialogo.entrada_a * (int)~dialogo.entrada_b;
so that the cancel response is filtered out.

CtrlLayoutOKCancel does 2 things that CtrlLayout doesn't:

- Assign standard OS icons to buttons (for X11 really). This uses the Button::OK() or Button::Cancel() functions.
- Give each button a callback that breaks the dialog's modal loop with a specific value (like IDOK). This value is then returned by Execute so that you can check which button was pushed (as above).

Using this method there is also additional behaviour when OK or Cancel is pushed by the user. OK calls Accept which validates all the ctrls on the dialog and stops the operation if any fail. Cancel calls Reject (which is used by some controls to end editing states) and if you have called Backup at any point it also calls Restore to revert to backed-up values.

Basically the idea is that by using CtrlLayoutOKCancel you get standard functionality automatically. Sometimes you want to over-ride the default OK handling, but this is easy to:

```
...  
ok <=<= THISBACK(OnOK);
```

```
void OnOK()  
{  
    if (!Accept())  
        return;  
    // Do something  
    ...  
  
    Break(IDOK);  
}
```