

---

Subject: Re: Porting a Delphi Application

Posted by [cbpporter](#) on Mon, 17 Sep 2007 17:40:54 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

luzr wrote on Mon, 17 September 2007 10:03

Usual custom here, when you post some sources like this, is to use package and perhaps provide another one as demonstration - that way downloaders can just unpack to some package nest, start theide, choose the test and run it.

This is not a complaint, just a note - I guess it is time to write U++ netiquette summary

I will post another improved version (current one gives error when trying to use it as base for a template layout because it misses a SetRect function with a Rect parameter) once it is ready and I'll try to have a proper package format. Previous post was only so people get an idea of what I'm trying to do (and if can be done easier point me in the right direction).

But I have a functional (barely) DropPane and now I'm trying to insert inside an array of 2x4 ToolButtons (I'm not using button because I need them to be flat-looking), like in the attached picture.

Layout and positioning is very important, so I did not use a ToolBar component, instead I hand placed all the buttons. I also need them to check when the user click on them. I managed to do this, but unfortunately I had to use pointers (old habits die hard). I wonder if there is a better way to do this in a U++ style?

layout

```
LAYOUT(ToolsLayout, 60, 200)
ITEM(ToolButton, btnMove, LeftPosZ(8, 22).TopPosZ(24, 22))
ITEM(ToolButton, btnEdit, LeftPosZ(30, 22).TopPosZ(24, 22))
ITEM(ToolButton, btnLine, LeftPosZ(8, 22).TopPosZ(46, 22))
ITEM(ToolButton, btnBezier, LeftPosZ(30, 22).TopPosZ(46, 22))
ITEM(ToolButton, btnRectangle, LeftPosZ(8, 22).TopPosZ(68, 22))
ITEM(ToolButton, btnEllipse, LeftPosZ(30, 22).TopPosZ(68, 22))
END_LAYOUT
```

code

```
struct ToolsDropPane: public WithToolsLayout<DropPane> {

    typedef ToolsDropPane CLASSNAME;

    ToolsDropPane();

    void SetActive(ToolButton *btn);
};
```

```

ToolsDropPane::ToolsDropPane()
{
    CtrlLayout(*this);

    btnMove.Image(CBPIImages::Move()).Text("Modify");
    btnMove.WhenAction = THISBACK1(SetActive, &btnMove);
    btnEdit.Image(CBPIImages::Edit()).Text("Edit");
    btnEdit.WhenAction = THISBACK1(SetActive, &btnEdit);
    btnLine.Image(CBPIImages::Line());
    btnLine.WhenAction = THISBACK1(SetActive, &btnLine);
    btnBezier.Image(CBPIImages::Bezier());
    btnBezier.WhenAction = THISBACK1(SetActive, &btnBezier);
    btnRectangle.Image(CBPIImages::Rectangle()).Text("Rectangle");
    btnRectangle.WhenAction = THISBACK1(SetActive, &btnRectangle);
    btnEllipse.Image(CBPIImages::Ellipse()).Text("Ellipse");
    btnEllipse.WhenAction = THISBACK1(SetActive, &btnEllipse);

    //btnLine.Check(true);
}

void ToolsDropPane::SetActive(ToolButton *btn)
{
    btn->Check(true);
}

```

On the other hand, you will notice that the third and fourth button doesn't have a Text. This is intentional. When I don't specify a text to a button inserted like this into a layout, I get a weird tool tip for it: "(Alt+Shift+address)". Does this mean anything?

## File Attachments

1) [panel2.PNG](#), downloaded 1803 times

---