Subject: Can't use mutators of custom widgets in Layout Designer
Posted by jlfranks on Sat, 19 Jan 2008 19:27:26 GMT
View Forum Message <> Reply to Message

We're developing a touch screen app that uses lots of custom widgets.

We have added Set and Get methods for custom members in the specialized widget. While we can see and set these in the Layout Designer, the compiler is not happy and complains.

It looks like if these properties are not in the Ctrl base class, we should not expose them to the Layout Designer.

We've removed them from the USC file and now set them in the constructor. Sure would be nice to see and set them in the Layout Editor.

Is there a way around this limitation?

--jlf

Subject: Re: Can't use mutators of custom widgets in Layout Designer
Posted by mirek on Mon, 04 Feb 2008 14:41:22 GMT
View Forum Message <> Reply to Message

jlfranks wrote on Sat, 19 January 2008 14:27We're developing a touch screen app that uses lots of custom widgets.

We have added Set and Get methods for custom members in the specialized widget. While we can see and set these in the Layout Designer, the compiler is not happy and complains.

It looks like if these properties are not in the Ctrl base class, we should not expose them to the Layout Designer.

We've removed them from the USC file and now set them in the constructor. Sure would be nice to see and set them in the Layout Editor.

Is there a way around this limitation?

--jlf

Yes. The problem is ordering. E.g. LeftPos returns Ctrl&, so you cannot bind derived class method on this in the chain.

Means, the most derived methods must be .lay stored first. Sometimes this happens naturally, if not, you can use "ordering operator" @. See e.g.:

```
ctrl Label {
 group "Static";

 GetMinSize() { return XMinSize(); }
 GetStdSize() { sz = GetTextSize("Label", .StdFont); sz.cy += 6; return sz; }

 >TextCtrl;

 Doc    SetLabel ? "Label of control" ;
 Align  SetAlign = ALIGN_LEFT @2;
 Font   SetFont = StdFont() @2;
 Color  SetInk = :SBlack @2;
 Frame  SetFrame @3;
```

Here @2 or @3 "pushes" SetFrame or SetAlign "up" to base class, means they will be before SetLabel in .lay.

More specifically, the level for each attribute is the depth of bases introduced by ">" (here >TextCtrl). Means for Label, attributes in TextCtrl will have level 1 for Label, unless they are too altered by @. Each defined attribute starts at 0, but adds @.

Uff, I do not see a better way how to explain this

Mirek
