
Subject: Re: Upp GUI size in Win and Ubuntu

Posted by [Sender Ghost](#) on Wed, 29 May 2013 09:45:38 GMT

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

nlnelson wrote on Tue, 28 May 2013 20:54I filed a bug as each time the package is opened up it adds the 'Z's back in.

I think, this is not a bug. This is how it works by default - layout zooming:

Logical positionsAnother issue that is related to positions is layout zooming. The problem here is that application can use font of different size or type than standard one used in layout designer when designing layout. That is why layout logical position need to be zoomed. Zoom factor is determined by ratio of text "OK Cancel Exit Retry" in runtime standard font size to Size(99, 13) - size of this text using standard font.

Zooming layout positions is done using placement methods with Z suffix (like LeftPosZ etc...).

Therefore, if you want to use layouts from layout designer, you have about two methods (for your specialized case):

1. Ctrl::NoLayoutZoom();
2. SetStdFont(Font font), where font must be the same (even DPI settings (for screen shots)) on different platforms to achieve the same sizes of layout widgets.

If you want to use placement methods without Z suffix, you could use them manually (as I already said):

Toggle Spoiler

```
#include <CtrlLib/CtrlLib.h>
```

```
using namespace Upp;
```

```
class CustomDlg : public TopWindow {
```

```
public:
```

```
    typedef CustomDlg CLASSNAME;
```

```
    CustomDlg();
```

```
    EditField Point1, Point2, Angle1, Angle2, Distance,  
              Data1, count, timezone;
```

```
    Label units, l1, l2, l3, l4, l5;
```

```
    Button P1, P2;
```

```
};
```

```
CustomDlg::CustomDlg()
```

```
{
```

```
    Title("Custom Dialog");
```

```
    SetRect(Size(260, 80));
```

```
    MinimizeBox();
```

```
    Add(Point1.WantFocus(false).LeftPos(44, 200).TopPos(0, 16));
```

```

Point1.SetFont(StdFont(12).Bold());
Add(Point2.WantFocus(false).LeftPos(44, 200).TopPos(16, 16));
Point2.SetFont(StdFont(12).Bold());
Add(Angle1.WantFocus(false).LeftPos(44, 96).TopPos(32, 16));
Angle1.SetFont(StdFont(12).Bold());
Add(Angle2.WantFocus(false).LeftPos(149, 96).TopPos(32, 16));
Angle2.SetFont(StdFont(12).Bold());
Add(Distance.WantFocus(false).LeftPos(44, 96).TopPos(48, 16));
Distance.SetFont(StdFont(12).Bold());
Add(units.LeftPos(138, 30).TopPos(48, 16));
Add(I1.LeftPos(0, 44).VSizePos(0, 67));
I1.SetLabel(t_(" Point 1")).SetFont(StdFont(11));
Add(I2.LeftPos(0, 44).TopPos(16, 16));
I2.SetLabel(t_(" Point 2")).SetFont(StdFont(11));
Add(I3.LeftPos(0, 44).TopPos(32, 16));
I3.SetLabel(t_(" Angle 1")).SetFont(StdFont(11));
Add(I4.LeftPos(142, 8).TopPos(32, 16));
I4.SetLabel("2").SetFont(StdFont(11));
Add(I5.LeftPos(0, 44).TopPos(48, 16));
I5.SetLabel(t_(" Distance")).SetFont(StdFont(11));
Add(Data1.WantFocus(false).LeftPos(1, 259).TopPos(64, 16));
Add(P1.LeftPos(244, 16).TopPos(0, 16));
P1.SetLabel("<").SetFont(StdFont().Bold());
Add(P2.LeftPos(244, 16).TopPos(16, 16));
P2.SetLabel("<").SetFont(StdFont().Bold());
Add(count.AlignRight(true).LeftPos(208, 52).TopPos(48, 16));
count.SetFont(StdFont().Bold());
Add(timezone.AlignRight(true).WantFocus(false).LeftPos(184, 24).TopPos(48, 16));
timezone.SetFont(StdFont().Bold());
}

```

```

GUI_APP_MAIN
{
    CustomDlg dlg;
    dlg.Run();
}

```

In conclusion, what you wanted is feature, which is not implemented. This feature, like "the possibility to use zoomed and not zoomed versions of placement inside the same layout", might have own consequences.

Edit:

The feature added to Redmine: "Ide (LayDes): Zoomable and not zoomable layout items".
