Subject: Option Label on Left Side draw
Posted by devilsclaw on Thu, 03 Aug 2023 21:27:13 GMT
View Forum Message <> Reply to Message

So I figured this would be a simple patch but I can seem to find what is causing me the problem.

Original Code:

```
void Option::Paint(Draw& w) {
 Size sz = GetSize();

 if(!IsTransparent())
  w.DrawRect(0, 0, sz.cx, sz.cy, SColorFace);

 Size isz = CtrlsImg::O0().GetSize();
 Size tsz = GetSmartTextSize(label, font);
 int ix = 0;
 int d = tsz.cy >> 1;
 int ty = (sz.cy - tsz.cy) / 2;
 int iy = (tsz.cy - isz.cy) / 2 + ty;

 if(box) {
  ix = d + DPI(4);
  if(tsz.cy > isz.cy) {
   ty = 0;
   iy = (tsz.cy - isz.cy) / 2;
  }
  else {
   iy = 0;
   ty = (isz.cy - tsz.cy) / 2;
  }
 }
 else
 if(!showlabel) {
  ix = (sz.cx - isz.cx) / 2;
  iy = (sz.cy - isz.cy) / 2;
 }

 int q = GetVisualState();
 int g = (!notnull || threestate) && IsNull(option) ? CtrlsImg::I_O2
                                                    : option == 1 ? CtrlsImg::I_O1
                                                                  : CtrlsImg::I_O0;
 if(switchimage)
  g = option ? CtrlsImg::I_S1 : CtrlsImg::I_S0;

 w.DrawImage(ix, iy, CtrlsImg::Get(g + q));

 if(showlabel) {
```

```
  bool ds = !IsShowEnabled();
  DrawSmartText(w, ix + isz.cx + DPI(2), ty, tsz.cx, label, font,
                ds || IsReadOnly() ? SColorDisabled : Nvl(color, GetLabelTextColor(this)),
                VisibleAccessKeys() ? accesskey : 0);
  if(HasFocus())
   DrawFocus(w, RectC(ix + isz.cx + DPI(2), ty - DPI(1), tsz.cx + DPI(3), tsz.cy + DPI(2)) & sz);
 }

 if(box) {
  w.Begin();
  w.ExcludeClip(ix - DPI(3), 0, isz.cx + DPI(8) + tsz.cx, tsz.cy);
  PaintLabelBox(w, sz, Null, d);
  w.End();
 }
}
```

As you can tell it pre gets the size of the image and the label.

It should be just a simple as this to get it on the left side. This i only for testing

```
void Option::Paint(Draw& w) {
 Size sz = GetSize();

 if(!IsTransparent())
  w.DrawRect(0, 0, sz.cx, sz.cy, SColorFace);

 Size isz = CtrlsImg::O0().GetSize();
 Size tsz = GetSmartTextSize(label, font);
 int ix = 0;
 int d = tsz.cy >> 1;
 int ty = (sz.cy - tsz.cy) / 2;
 int iy = (tsz.cy - isz.cy) / 2 + ty;

 if(box) {
  ix = d + DPI(4);
  if(tsz.cy > isz.cy) {
   ty = 0;
   iy = (tsz.cy - isz.cy) / 2;
  }
  else {
   iy = 0;
   ty = (isz.cy - tsz.cy) / 2;
  }
 }
 else
 if(!showlabel) {
  ix = (sz.cx - isz.cx) / 2;
```

```
 iy = (sz.cy - isz.cy) / 2;
}

int q = GetVisualState();
int g = (!notnull || threestate) && IsNull(option) ? CtrlsImg::I_O2
                                        : option == 1 ? CtrlsImg::I_O1
                                                      : CtrlsImg::I_O0;
if(switchimage)
 g = option ? CtrlsImg::I_S1 : CtrlsImg::I_S0;

w.DrawImage(ix + tsz.cx + DPI(2), iy, CtrlsImg::Get(g + q));

if(showlabel) {
 bool ds = !IsShowEnabled();
 DrawSmartText(w, ix, ty, tsz.cx, label, font,
               ds || IsReadOnly() ? SColorDisabled : Nvl(color, GetLabelTextColor(this)),
               VisibleAccessKeys() ? accesskey : 0);
 if(HasFocus())
  DrawFocus(w, RectC(ix, ty - DPI(1), tsz.cx + DPI(3), tsz.cy + DPI(2)) & sz);
}

if(box) {
 w.Begin();
 w.ExcludeClip(ix - DPI(3), 0, isz.cx + DPI(8) + tsz.cx, tsz.cy);
 PaintLabelBox(w, sz, Null, d);
 w.End();
}
}
```

But this results in anything beyond where the checkbox button image not drawing anything on click

This code shows exactly what I mean when testing a app

```
void Option::Paint(Draw& w) {
 Size sz = GetSize();

 if(!IsTransparent())
  w.DrawRect(0, 0, sz.cx, sz.cy, SColorFace);

 Size isz = CtrlsImg::O0().GetSize();
 Size tsz = GetSmartTextSize(label, font);
 int ix = 0;
 int d = tsz.cy >> 1;
 int ty = (sz.cy - tsz.cy) / 2;
 int iy = (tsz.cy - isz.cy) / 2 + ty;
```

```
 if(box) {
  ix = d + DPI(4);
  if(tsz.cy > isz.cy) {
   ty = 0;
   iy = (tsz.cy - isz.cy) / 2;
  }
  else {
   iy = 0;
   ty = (isz.cy - tsz.cy) / 2;
  }
 }
 else
 if(!showlabel) {
  ix = (sz.cx - isz.cx) / 2;
  iy = (sz.cy - isz.cy) / 2;
 }

 int q = GetVisualState();
 int g = (!notnull || threestate) && IsNull(option) ? CtrlsImg::I_O2
                                        : option == 1 ? CtrlsImg::I_O1
                                                      : CtrlsImg::I_O0;
 if(switchimage)
  g = option ? CtrlsImg::I_S1 : CtrlsImg::I_S0;

  if(option != 1) {
  w.DrawRect(0, 0, sz.cx, sz.cy, Red());
  } else {
  w.DrawRect(0, 0, sz.cx, sz.cy, Green());
  }

 if(showlabel) {
  bool ds = !IsShowEnabled();
  DrawSmartText(w, ix, ty, tsz.cx, label, font,
                ds || IsReadOnly() ? SColorDisabled : Nvl(color, GetLabelTextColor(this)),
                VisibleAccessKeys() ? accesskey : 0);
  if(HasFocus())
   DrawFocus(w, RectC(ix, ty - DPI(1), tsz.cx + DPI(3), tsz.cy + DPI(2)) & sz);
 }

 if(box) {
  w.Begin();
  w.ExcludeClip(ix - DPI(3), 0, isz.cx + DPI(8) + tsz.cx, tsz.cy);
  PaintLabelBox(w, sz, Null, d);
  w.End();
 }
}
```

There is something choosing where the control painting will get updated and only painting that part. I have no clue what code is doing that.
