Subject: DropList STDFONT size changing after Drawn on screen
Posted by devilsclaw on Thu, 20 Apr 2023 16:06:47 GMT
View Forum Message <> Reply to Message

So I have dynamic elements in my UI and when I select a option button it changes what is in a DropList.

Based off the content of the DropList I resize it.

Before the TopWindow is displayed on screen the STDFONT on my systems height is 13. Once it is displayed the STDFONT height is 11, which also affect the width

Why would the height change once the windows is displayed? it makes it impossible to actually calculate the width of something.

```
int egui_width(Upp::Font f, const char* s, size_t length, int minsize) {
  int width = 0;
  for(int i = 0; i < length; i++) {
    width += f[s[i]];
  }
  if(minsize > -1 && width < minsize) {
    width = minsize;
  }
  return Upp::DPI(width);
}

int egui_width(Upp::Font f, Upp::String s, int minsize) {
  return egui_width(f, s, s.GetCount(), minsize);
}

int egui_width(Upp::DropList& c, int minsize) {
  Upp::Font f = c.GetPreeditFont();
  int width = 0;
  for(int i = 0; i < c.GetCount(); i++) {
    int twidth = egui_width(f, c.GetValue(i));
    if(twidth > width) {
      width = twidth;
    }
  }

  width += Upp::DPI(c.GetMinSize().cx);
  width += Upp::DPI(24);

  if(minsize > -1 && width < Upp::DPI(minsize)) {
    width = Upp::DPI(minsize);
  }
```

```
  return width;
}
```

Subject: Re: DropList STDFONT size changing after Drawn on screen
Posted by devilsclaw on Thu, 20 Apr 2023 16:11:43 GMT
View Forum Message <> Reply to Message

NOTE: the origin being 13 and the post display being 11 but what is drawn on screen is still the size of 13 but it reports 11

Subject: Re: DropList STDFONT size changing after Drawn on screen
Posted by devilsclaw on Thu, 20 Apr 2023 16:19:00 GMT
View Forum Message <> Reply to Message

I guess it to do the fact the the returned font changes based on various variables before it shows on screen it returns StdFont() without anything passed to it

As a temp fix for droplist i always use StdFont(); now but it seems to me what is reported should be what is used to draw on screen

```
Font Ctrl::GetPreeditFont()
{
 static int pheight = -1;
 static Font pfont;
 if(!focusCtrl)
  return StdFont();
 int height = max(focusCtrl->GetCaret().GetHeight(), DPI(7));
 if(height != pheight) {
  pheight = height;
  while(pheight > 0) {
   pfont = StdFont(height);
   if(pfont.GetCy() < pheight)
    break;
   height--;
  }
  if(height == 0)
   pfont = StdFont();
 }
 return pfont;
}
```