Subject: Zooming font of RichText to the size of StdFont
Posted by Novo on Sat, 03 Jan 2009 05:45:15 GMT
View Forum Message <> Reply to Message

Do I understand correctly that zoom below will make size of the font of RichText equal to the size of StdFont?

Zoom zoom(StdFont().GetHeight(), ScreenDraw(false).GetPixelsPerInch().cy);

Subject: Re: Zooming font of RichText to the size of StdFont
Posted by mirek on Sun, 04 Jan 2009 14:07:43 GMT
View Forum Message <> Reply to Message

No.

If I remember well, it should be

Zoom(StdFont().GetHeight(), 100);

for the default font size, which is 100 dots (pixels).

Anyway, result still does not have to be perfect, because final height and requested height are two different things...

Mirek

Subject: Re: Zooming font of RichText to the size of StdFont
Posted by Novo on Wed, 07 Jan 2009 04:20:55 GMT
View Forum Message <> Reply to Message

Thanks!

ScreenDraw(false).GetPixelsPerInch().cy returns 96 on all my Windows computers.

Actually, zooming looks better in case of 96. In this case letters do not merge with each other.

Subject: Re: Zooming font of RichText to the size of StdFont
Posted by mirek on Wed, 07 Jan 2009 12:29:47 GMT
View Forum Message <> Reply to Message

Novo wrote on Tue, 06 January 2009 23:20Thanks!

ScreenDraw(false).GetPixelsPerInch().cy returns 96 on all my Windows computers.

It is because it is set as screen property. If you set "Large fonts" in windows screen dialog, it changes to 120dpi (AFAIK).

Really, I have said it before, but people for some strange reason put too much emphasis to this basically meaningless value (GetPixelsPerInch for screen). It is really meaningful only for printer.

Mirek

Subject: Re: Zooming font of RichText to the size of StdFont
Posted by Zeth on Sun, 08 Feb 2009 10:46:02 GMT
View Forum Message <> Reply to Message

GetPixelsPerInch returns 96 even though I have set dpi to 120 in (Desktop) properties -> Settings -> Advanced. I'm using Windows XP. I tried ScreenDraw(false).GetPixelsPerInch().cy and ScreenDraw(false).GetPixelsPerInch().cx

---

Subject: Re: Zooming font of RichText to the size of StdFont
Posted by mirek on Mon, 09 Feb 2009 07:03:59 GMT
View Forum Message <> Reply to Message

Check

void Draw::LoadCaps()

- we are only reading info from GDI. If it returns 96, it is because GDI says 96 (IMO).

However, I still insist that this value is basically useless. I guess Win32 guys think the same

Mirek

Subject: Re: Zooming font of RichText to the size of StdFont
Posted by Zeth on Mon, 09 Feb 2009 08:39:32 GMT
View Forum Message <> Reply to Message

I've read about it and it seems to be a Windows XP specific problem with GetDeviceCaps always returning 96. I ended up checking my main window size (it isn't resizable):

```
if(this->GetSize().cx > 690){ //Assume Large font
   DPI = 120;
}
float Multiplier = DPI / 96;
```

I've attached a small testcase which shows my problem. If you load it with 96 dpi all is ok, but when you change it to use 120dpi then you can see the scrollArea StaticText control is not automatically resized and the buttons are no longer of the same size. Basically all the manually added controls are not resized according to the current dpi. How do you guys handle this?

File Attachments

1) Testcase.zip, downloaded 570 times

Subject: Re: Zooming font of RichText to the size of StdFont
Posted by mirek on Fri, 13 Feb 2009 14:11:04 GMT
View Forum Message <> Reply to Message

Zeth wrote on Mon, 09 February 2009 03:39I've read about it and it seems to be a Windows XP specific problem with GetDeviceCaps always returning 96. I ended up checking my main window size (it isn't resizable):


```
if(this->GetSize().cx > 690){ //Assume Large font
   DPI = 120;
}
float Multiplier = DPI / 96;
```

I've attached a small testcase which shows my problem. If you load it with 96 dpi all is ok, but when you change it to use 120dpi then you can see the scrollArea StaticText control is not automatically resized and the buttons are no longer of the same size. Basically all the manually added controls are not resized according to the current dpi. How do you guys handle this?

FontZoom.

Check Ctrl::

```
 static int  HorzLayoutZoom(int cx);
 static int  VertLayoutZoom(int cy);
```

The basic idea is to query the size of specific text with current default GUI font. Knowing its "default size" (as defined by its size in WinXP with small fonts, but that is hardly relevant) you compute zooming ratio.

Mirek

Subject: Prompt with non-system dpi for text at first
Posted by Sender Ghost on Fri, 27 Nov 2009 08:41:15 GMT
View Forum Message <> Reply to Message

Hello.
On my system I have 108 dpi instead of 96 dpi by default for fonts.
When I use Prompt function before any TopWindow class in the GUI_APP_MAIN with following test case:

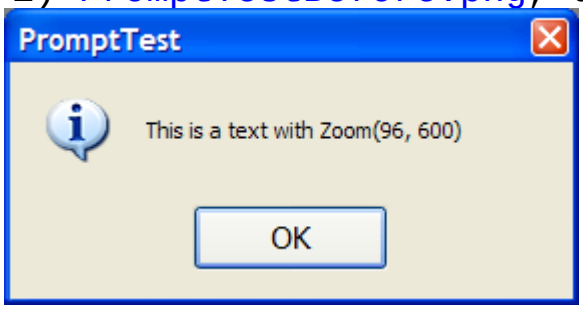
```
#include <CtrlLib/CtrlLib.h>

using namespace Upp;

GUI_APP_MAIN
{
 Zoom z = GetRichTextStdScreenZoom();
 PromptOK(NFormat("This is a text with Zoom(%d, %d)", z.m, z.d));
 z = GetRichTextStdScreenZoom();
 PromptOK(NFormat("This is the same text with Zoom(%d, %d)", z.m, z.d));
}
```

I have non-system dpi at first activation of the Prompt function.

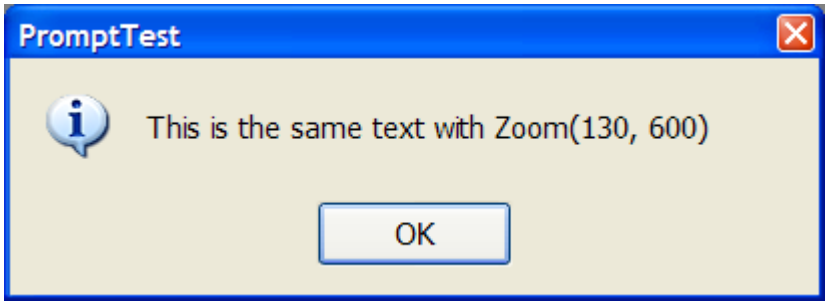
Before introduction of new Drawing system I didn't have this problem, because U++ application was used system dpi.
Current tested revision is 1724.

File Attachments

1) PromptTestBefore.png, downloaded 1364 times



2) PromptTestAfter.png, downloaded 1385 times



Subject: Re: Prompt with non-system dpi for text at first
Posted by Sender Ghost on Fri, 27 Nov 2009 14:13:49 GMT
View Forum Message <> Reply to Message

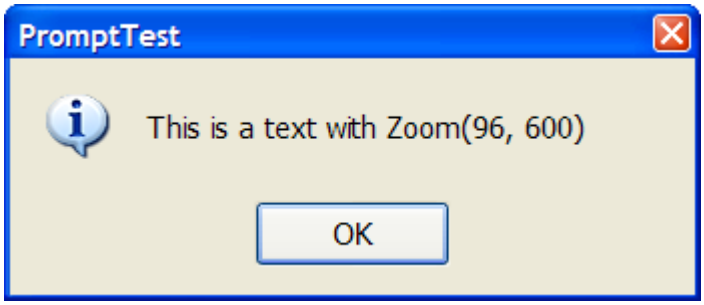
I found the following solution for my problem:

```
 int Prompt(Callback1<const String&> WhenLink,
           const char *title, const Image& iconbmp, const char *qtf, bool okcancel,
           const char *button1, const char *button2, const char *button3,
      int cx,
      Image im1, Image im2, Image im3)
 {
+ int bcy = Ctrl::VertLayoutZoom(24);
+ int bcx = Ctrl::HorzLayoutZoom(72);
  int fcy = Draw::GetStdFontCy();
  PromptDlgWnd__ dlg;
  RichTextCtrl qtfctrl;
  Icon        icon;
  qtfctrl.WhenLink = WhenLink;
  icon.SetImage(iconbmp);
  Button b1, b2, b3;
  qtfctrl.SetQTF(String("[G1 ") + qtf, GetRichTextStdScreenZoom());
- int bcy = Ctrl::VertLayoutZoom(24);
- int bcx = Ctrl::HorzLayoutZoom(72);
  if(button1)
   bcx = max(2 * fcy + GetTextSize(button1, Draw::GetStdFont()).cx, bcx);
  if(button2)
   bcx = max(2 * fcy + GetTextSize(button2, Draw::GetStdFont()).cx, bcx);
  if(button3)
   bcx = max(2 * fcy + GetTextSize(button3, Draw::GetStdFont()).cx, bcx);
```

Its just reordering of control flow. So, Ctrl::Csizeinit() can be initialized before GetRichTextStdScreenZoom().

File Attachments

1) PromptTestBeforeWithFix.png, downloaded 1394 times



Subject: Re: Prompt with non-system dpi for text at first
Posted by mirek on Sun, 29 Nov 2009 22:31:43 GMT
View Forum Message <> Reply to Message

Thank you for reporting the problem.

I believe even better solution is to put Csizeinit call to GUI_APP_MAIN.

Subject: Re: Prompt with non-system dpi for text at first
Posted by Sender Ghost on Thu, 10 Dec 2009 14:52:52 GMT
View Forum Message <> Reply to Message

luzr wrote on Sun, 29 November 2009 23:31Thank you for reporting the problem.

I believe even better solution is to put Csizeinit call to GUI_APP_MAIN.
Thank you for your solution.

Considering new topic I propose following modified solution:
- Make public static SizeInit function to call inlined Csizeinit function.
- Use created SizeInit function in the CtrlCore/Win32Wnd.cpp::InitWin32, CtrlCore/X11App.cpp::InitX11 functions.

Normal diff for 1796 revision:

File: CtrlCore\Ctrl.cpp
770a771,775
> void Ctrl::SizeInit()
> {
> Csizeinit();
> }
>

File: CtrlCore\CtrlCore.h
1328a1329,1330
>

> static void SizeInit();

File: CtrlCore\Win32Wnd.cpp
288c288
< Csizeinit();

> SizeInit();

File: CtrlCore\X11App.cpp
429c429
< Csizeinit();

> SizeInit();

Subject: Re: Prompt with non-system dpi for text at first
Posted by mirek on Thu, 10 Dec 2009 18:16:28 GMT
View Forum Message <> Reply to Message

Well, I have just removed "inline"...

Mirek
