
Subject: Re: Upp GUI size in Win and Ubuntu
Posted by [Sender Ghost](#) on Thu, 09 May 2013 05:32:05 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello.

Didier wrote on Mon, 22 April 2013 11:17 I expect the dialog to STAY 320*240

He can't expect this for different user settings, than default (on the same operating system, window manager, etc.), at least for screen shot images. For example, if you change the DPI settings for your screen on Windows XP to some value, bigger than 100% (usually, 96), then you will have proportionally larger window size for GUI applications. Therefore, they might look different to the user with different settings (and to you, if (s)he sent screen shots of the GUI). But internally, the application will behave like it have the same size (320x240, in this case). In conclusion, this is what he could expect: the same (internal to application) window size, which might look proportionally smaller/bigger by user settings.

In case of Neil's screen shot, we have different fonts (The Wine is Tahoma; the Gnome is Liberation Sans) and their heights, but GUI is proportional, which is default behaviour of current U++ GUI applications.

To explicitly change the default font for the GUI, you have SetStdFont(Font font) function:

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

using namespace Upp;

GUI_APP_MAIN
{
    RDUMP(GetStdFont());

    Font fnt;
    fnt.FaceName("Tahoma").Height(15);
    SetStdFont(fnt);

    PromptOK("Hello World!");
}
```

To disable layout scaling by font size, you have Ctrl::NoLayoutZoom() static method:

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

using namespace Upp;

GUI_APP_MAIN
{
    PromptOK("With layout zoom");
    Ctrl::NoLayoutZoom();
    PromptOK("No layout zoom");
}
```

```
}
```

There are other static methods of Ctrl, which you might find interesting.

Edit: Changed Draw::SetStdFont(Font font) to SetStdFont(Font font) function.
