
Subject: Paste Properly Scaled Images Into RichEdit
Posted by [emve](#) on Sun, 24 Nov 2024 20:50:24 GMT

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

Hi,

I'm evaluating RichEdit using this code in main.cpp:

```
#include <CtrlLib/CtrlLib.h>
#include <RichEdit/RichEdit.h>
using namespace Upp;

GUI_APP_MAIN
{
    RichText txt;
    {
        RichPara para;
        RichPara::Format fmt;
        (Font&)fmt = Monospace(50).Bold();
        para.Cat("", fmt);
        txt.Cat(para);
    }

    RichEdit e;
    e.ShowCodes(Null); //hide enter marks
    e.Clear();

    HDC screen = GetDC(NULL); // Get device context for the screen
    int dpiX = GetDeviceCaps(screen, LOGPIXELSX); // Horizontal DPI
    int dpiY = GetDeviceCaps(screen, LOGPIXELSY); // Vertical DPI
    ReleaseDC(NULL, screen);

    e.Pick(pick(txt));

    TopWindow win;
    win.Add(e.SizePos());
    win.Run();
}
```

When I copy paste an image (I use GreenShot) into RichEdit, the pasted image shown is bigger.

Here I copied (PrintScreen using GreenShot) top left corner of the window and pasted it back into RichEdit:

When I paste the image from file, the result is the same.

Can it be solved anyhow? I asked chatgpt :) and it kinda hallucinated this solution (as there is apparently no WhenPaste event handler in RichEdit object):

```
class MyApp : public TopWindow {  
    RichEdit richEdit;  
  
public:  
    MyApp() {  
        Title("Adjust DPI for Pasted Image").Sizeable();  
  
        // Intercept the paste action and adjust the image DPI  
        richEdit.WhenPaste = [&]() {  
            // Check if the clipboard contains an image  
            if (ClipboardHas<Image>()) {  
                Image img = ReadClipboardImage();  
                if (!img.IsEmpty()) {  
                    // Get system DPI  
                    Size dpi = GetPrimaryScreenDPI(); // Get DPI of the primary screen  
                    double scaleFactor = dpi.cx / 96.0; // Assuming 96 DPI as the base  
  
                    // Scale the image  
                    int newWidth = img.GetWidth() * scaleFactor;  
                    int newHeight = img.GetHeight() * scaleFactor;  
                    Image scaledImage = Rescale(img, newWidth, newHeight);  
  
                    // Insert the scaled image into the RichEdit control  
                    richEdit.PasteImage(scaledImage);  
                }  
            } else {  
                // Default paste behavior for non-image content  
                richEdit.Paste();  
            }  
        };  
        Add(richEdit.SizePos());  
    }  
};
```

Michał

File Attachments

1) [2024-11-24 21_38_02-Clipboard.png](#), downloaded 245 times

Subject: Re: Paste Properly Scaled Images Into RichEdit

Posted by [emve](#) on Sun, 24 Nov 2024 20:55:17 GMT

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

Here is the image with border:

File Attachments

1) [2024-11-24 21_52_02-.png](#), downloaded 205 times

Subject: Re: Paste Properly Scaled Images Into RichEdit

Posted by [emve](#) on Mon, 25 Nov 2024 09:20:38 GMT

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

I'm just wondering where those numbers came from:

But anyway, someone fix the proper image pasting to RichEdit, please.

Michal

File Attachments

1) [UPP-InitSize.png](#), downloaded 215 times

Subject: Re: Paste Properly Scaled Images Into RichEdit

Posted by [mirek](#) on Tue, 26 Nov 2024 13:03:42 GMT

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

emve wrote on Mon, 25 November 2024 10:20 I'm just wondering where those numbers came from:

But anyway, someone fix the proper image pasting to RichEdit, please.

Michal

TLDR: Nothing to fix, but I can add ScreenRichEdit

The final target of RichText is printer. So pixel values are sort of irrelevant. U++ has two coordinate systems (only...): Screen pixels and 600DPI laser printer "pixels" (these are called "DOTS" :). So what you see is an attempt to approximately convert screen image size to paper. As most displays are 96DPI, we get 600/96 ratio. And those 2000 values are just sanity limit. But also beware that those "dots" are then converted back, so e.g. changing this to 1/1 does not help.

What you want is possible, but needs some work. Adding to the queue, please remind me if nothing happens in 1-2 months.

Subject: Re: Paste Properly Scaled Images Into RichEdit

Posted by [emve](#) on Tue, 26 Nov 2024 16:05:21 GMT

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

mirek wrote on Tue, 26 November 2024 14:03emve wrote on Mon, 25 November 2024 10:20I'm just wondering where those numbers came from:

But anyway, someone fix the proper image pasting to RichEdit, please.

Michal

TLDR: Nothing to fix, but I can add ScreenRichEdit

The final target of RichText is printer. So pixel values are sort of irrelevant. U++ has two coordinate systems (only...): Screen pixels and 600DPI laser printer "pixels" (these are called "DOTS" :). So what you see is an attempt to approximately convert screen image size to paper. As most displays are 96DPI, we get 600/96 ratio. And those 2000 values are just sanity limit. But also beware that those "dots" are then converted back, so e.g. changing this to 1/1 does not help.

What you want is possible, but needs some work. Adding to the queue, please remind me if nothing happens in 1-2 months.

Ok, thank you for the clarification Mirek.
And yes, ScreenRichEdit will be welcome :).
Michal
