Subject: ImageBuffer Paint problem
Posted by kodos on Wed, 24 Sep 2008 16:42:26 GMT
View Forum Message <> Reply to Message

Hi,

I have some problems with my Paint method where I simply draw an ImageBuffer. I always get the following error message after i drag the window to the border of the screen an move it a bit (so that it gets repainted)

Assertion failed in C:\upp\uppsrc\CtrlCore\Win32Proc.cpp, line 77
IsNull(sPainting)

I attached a very simple testcase.

File Attachments
1) ImgBufferTest.zip, downloaded 443 times

Subject: Re: ImageBuffer Paint problem
Posted by mr_ped on Wed, 24 Sep 2008 16:46:43 GMT
View Forum Message <> Reply to Message

You are very likely hitting this issue:
http://www.ultimatepp.org/forum/index.php?t=msg&th=3262& amp; amp; amp;start=0&
edit: (I didn't even bother to check your source, sorry ... if it's not related, I will feel ashamed for at least 15minutes, ok? Than again, you got reply in 4minutes, I'm like Pentium, maybe not right, but fast . )

Subject: Re: ImageBuffer Paint problem
Posted by kodos on Wed, 24 Sep 2008 17:11:50 GMT
View Forum Message <> Reply to Message

Thank you for the fast answer

It seems that this is the problem of the testcase but not of my real problem.

But why can't I use an ImageBuffer as a member variable of a class? I still don't understand the actual problem.

I need a pointer to the image data so that i can manipulate it with a pointer. So I need an Image object and a pointer that that use the same area of memory. Is there another possibility if I can't use ImageBuffer?

Subject: Re: ImageBuffer Paint problem
Posted by kodos on Wed, 24 Sep 2008 19:02:03 GMT
View Forum Message <> Reply to Message

Hm, ok I think I found a solution even if it is not really nice imho.

It works like I want it if I use an ImageBuffer but for the drawing I use the following code:

```
Image i = buffer;
draw.DrawImage(0, 0, i);
buffer = i;
```

Subject: Re: ImageBuffer Paint problem
Posted by mirek on Wed, 24 Sep 2008 19:52:24 GMT
View Forum Message <> Reply to Message

I think this is definitely quite confusing...

Anyway, let us say this strange requirement has its root in performance and implementation issues.

The main problem is that Image is sort of "immutable", which allows it to cache host platform resource handle. That is not possible with ImageBuffer, whose content can change at any moment.

If it would be supposed to work in "non-ugly" way, ImageBuffer <-> Image conversions would require the full copy of ImageBuffer pixels.

In practice, you should use ImageBuffer only as local variable, "Image content accesor". Or be very very careful...

Mirek