

## Re: Double Buffering and Initial Paint

**Source:**

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework.windowsforms/2004-04/0888.html>

---

**From:** Bob Powell [MVP] ([bob\\_at\\_spamkiller\\_bobpowell.net](mailto:bob_at_spamkiller_bobpowell.net))

**Date:** 04/16/04

Date: Fri, 16 Apr 2004 23:04:25 +0200

AllPaintingInWMPaint prevents the OnEraseBackground event from firing so you are responsible for clearing the control's surface.

Use e.Graphics.Clear(<somecolour>) in the OnPaint handler.

--

Bob Powell [MVP]

Visual C#, System.Drawing

All you ever wanted to know about ListView custom drawing is in Well Formed.

<http://www.bobpowell.net/currentissue.htm>

Answer those GDI+ questions with the GDI+ FAQ

[http://www.bobpowell.net/gdiplus\\_faq.htm](http://www.bobpowell.net/gdiplus_faq.htm)

Read my Blog at <http://bobpowelldotnet.blogspot.com>

"msnews.microsoft.com" <[bkotch@liquidnet.com](mailto:bkotch@liquidnet.com)> wrote in message

news:e9Chcm%23IEHA.3712@TK2MSFTNGP09.phx.gbl...

> when I turn on double buffering ... I get the undesirable side effect that

> the control which is being double buffered momentarily has the same

> background as whatever the window popped up on top of. So for example, if

> the background is the desktop, the background of the control will

> momentarily be blue. (This is especially a problem in my app, b/c not all

> controls have double buffering turned on, so some come up white, and

others

> blue, blah!)

>

> After much experimentation ... this seems to be directly related to the flag

>

> this.SetStyle(ControlStyles.AllPaintingInWmPaint, true);

> When this is set to false, flickering once again occurs, but the

background

> doesn't exhibit this undesirable initial transparency effect.

>

> So the question is ... is there a way NOT to have this background bleed into

> my app, but STILL not have any flicker.

>

>

>

>