

# Re: Layered Windows in VMWare and Terminal Services

---

*Source:*

<http://www.tech-archive.net/Archive/Development/microsoft.public.win32.programmer.gdi/2007-12/msg00038.html>

---

- *From:* "Fredo" <fredo@xxxxxxxxxxxx>
  - *Date:* Wed, 19 Dec 2007 09:29:25 -0600
- 

Ivan,

Thanks for the info...

We're using UpdateLayeredWindow.

The problem is noticeably worse, for example, on a machine with Terminal Services that's limited to 800x600x256. In that case, our splash screen and login dialogs (both of which use ULW) show up completely black.

I actually have my development environment set up in a VMWare VM, which is where I've personally experienced the issue the most. I've seen it in TS and it's a bit different, but I suspect the underlying problem is pretty similar.

MS has acknowledged that layered windows in terminal services is broken and said they have no plans to fix it (I no longer have a link to the issue where this was raised).

VMWare has their own SVGA driver which appears to be (or act) a bit higher-end than what I've seen in the TS drivers.

I know that the display change is causing a repaint (a couple or three, actually), but there's something else going on. After a display change, something is either being reset or initialized that allows ULW to continue to function in the app for the lifetime of the app.

It's this reset/initialization that I'd like to reproduce, if possible, from within the code. I suspect it's something happening at the driver level that I probably won't be able to reproduce without actually causing a mode change, however.

I used Spy++ to get all the messages sent to the window during a mode change and then I tried to internally send any that appeared they might have some affect, using the same parameters (where applicable) that happened during the mode change. None of these had any effect, however.

## Re: Layered Windows in VMWare and Terminal Services

What I'm seeing, specifically, is the areas where we have layered windows in use, the window itself is entirely transparent (invisible) with the exception of controls that are on the window. When I do the mode change, the windows display correctly.

Since our splash and login dialogs are both using ULW, they both show up transparent except for the controls on them. For our main app, the main frame window uses ULW and therefore there's a transparent border around the app that doesn't repaint right (that is, when it's first displayed, you see through the areas of the frame window and see what's behind them, but when you move the app window around, the frame window doesn't repaint, so you see whatever was behind the frame window when it originally displayed being dragged around.

Does that make sense?

Pete

"Ivan Brugiolo [MSFT]" <ivanbrug@xxxxxxxxxxxxxxxxxxxx> wrote in message [news:e1TOYPcQIHA.5160@xxxxxxxxxxxxxxxxxxxx](mailto:news:e1TOYPcQIHA.5160@xxxxxxxxxxxxxxxxxxxx)

Does this behavior reproes, for example, on the local machine, without WMWare or anything else, when you have installed the VGA display driver and you change, for example, from 800x600@32bpp to 800x600@16bpp, or from 800x600@32bpp to 1024x768@32bpp ?

In general, Layered Windows can be of two kind: ULW (from UpdateLayeredWindow) or SLWA (SetLayeredWindowAttribute). ULW window are always backed by a system memory redirection surface, while SLWA may have a video-memory copy, depending on what the display driver supports, and if you are running Win20009, WinXp, Win2003 or Vista.

You naturally understand that any display-driver realization is lost and possibly transfered between mode-changes, but, the system is designed in such a way that either the resource is transferred (the ULW case), or, if it is destroyed, the application is asked to repaint in order to recreate the resource underneath (this would be the SLWA case).

That said for the general theory of mode-changes, I seem to remember that most of the Visual Machine software do emulate some well-known ancient video-card (for example, an S3 from the late 90's), for which either VGA.sys is the display driver, or, some un-optimized compatible display driver is used.

Can you describe, at this point, a little bit better what the application is doing, what are the expected results, and what are the actual results ?

Re: Layered Windows in VMWare and Terminal Services

—  
This posting is provided "AS IS" with no warranties, and confers no rights.

Use of any included script samples are subject to the terms specified at <http://www.microsoft.com/info/copyright.htm>

"Fredo" <fredo@xxxxxxxxxx> wrote in message  
[news:t7CdnabLGasLs\\_XanZ2dnUVZ\\_tOtnZ2d@xxxxxxxxxxxxxxxx](news:t7CdnabLGasLs_XanZ2dnUVZ_tOtnZ2d@xxxxxxxxxxxxxxxx)

Our application uses layered windows in several places and, invariably, we have issues running the app inside Terminal Services and VMWare.

I've noticed that, at least in VMWare (I haven't tested this in terminal services yet), if I change the screen resolution while the app is running (even if I change it and then change it back to the original settings), the app then displays properly until I close the app. It's like something gets triggered by the resolution change that makes layered windows work. Does anyone have any idea what might be going on here? Maybe it's something I can reproduce programmatically (without actually changing the screen resolution).

Thanks