Subject: Question to UPP developers: the issue with windows rendering (Windows XP)
Posted by porto on Tue, 12 Oct 2010 07:05:57 GMT
View Forum Message <> Reply to Message

There is a strange issue with UPP windows rendering: the window title bar becomes rounded shape (themed) not immediately, but first rectangle is drawn. Animated GIF File is attached. Native applications that use WinAPI and QT apps don't have this issue. It's a little bit annoying. Can you explain or fix it?

Sorry for bad English.

File Attachments
1) picasion.com_2c5f4df965320cb471c86a302088ca29.gif, downloaded 1154 times



Subject: Re: Question to UPP developers: the issue with windows rendering (Windows XP)
Posted by koldo on Wed, 13 Oct 2010 06:53:10 GMT
View Forum Message <> Reply to Message

Hello Porto

Excellent gif to explain it. However I cannot reproduce it, perhaps because window opening in my case is so fast that I cannot see the defect.

Subject: Re: Question to UPP developers: the issue with windows rendering (Windows XP)
Posted by porto on Wed, 13 Oct 2010 08:49:46 GMT
View Forum Message <> Reply to Message

Hello koldo!
Maybe your system is too fast. The best way to see this issue is to use an application that can record video screenshots, and then play the recorded video in slow motion mode or by frames.

I made another animation with this program:
http://www.wisdom-soft.com/downloads/setupautoscreenrecorder free.exe
If you will try this app, don't set "Object / Window" of "Active window" in "Record What" section. I dod't know why, but these options don't work properly with UPP windows for me.

Animation file is attached. (This animation is recorded on other system)

File Attachments

1) AutoScreenRecorder_08 Oct. 13 11.34.avi, downloaded 640 times

---

Subject: Re: Question to UPP developers: the issue with windows rendering (Windows XP)
Posted by koldo on Wed, 13 Oct 2010 10:24:13 GMT
View Forum Message <> Reply to Message

Hello porto

I have used your program to record the desktop but I did not get the problem.

I was wondering if the issue is not about how a new window is rendered, but about how an U++ application fills the upper borders of a window that is opened over it.

Subject: Re: Question to UPP developers: the issue with windows rendering (Windows XP)
Posted by porto on Wed, 13 Oct 2010 12:18:32 GMT
View Forum Message <> Reply to Message

You're right, I incorrectly described the issue - it's about how an U++ application fills the upper borders of a window.
Very strange. I'm getting this issue on 3 different systems with WindowsXP.
What operating system and version of UPP do you use? Can you upload your recorded video?

P.S. Can you ask the developers, maybe they would look at my post with a description of the issue? I really liked UPP, but a little bit annoying this issue.

P.P.S. I have just tried to record another video, and at first in slow view mode I dont't noticed the issue. But when I loaded the video to VirtualDub and watched it frame by frame - I noticed this issue.

Thanks.

Subject: Re: Question to UPP developers: the issue with windows rendering (Windows XP)
Posted by unodgs on Wed, 13 Oct 2010 13:07:51 GMT
View Forum Message <> Reply to Message

I think I saw this effect when parsing progress indicator dialog showed up. In other cases I was not able to see it. Open CtrlCore/Win32Wnd.cpp. There is InitWin32 method. Try to change values for wc.hbrBackground and/or wc.style (maybe CS_SAVEBITS or 0x20000 causes this, try to remove them). It's just an idea. Hard to say what's the source of the problem.

Subject: Re: Question to UPP developers: the issue with windows rendering (Windows XP)
Posted by porto on Wed, 13 Oct 2010 13:37:40 GMT
View Forum Message <> Reply to Message

Thanks, I tried to do as you wrote, but it didn't help. Maybe I did something wrong. I'd love to know the opinion of UPP creators about this issue, if it's of course possible...

Subject: Re: Question to UPP developers: the issue with windows rendering (Windows XP)
Posted by mirek on Wed, 13 Oct 2010 16:44:50 GMT
View Forum Message <> Reply to Message

porto wrote on Wed, 13 October 2010 09:37Thanks, I tried to do as you wrote, but it didn't help. Maybe I did something wrong. I'd love to know the opinion of UPP creators about this issue, if it's of course possible...

Well, the problem is we have little influence there: All windows borders (including these round corners) are in fact drawn by win32.

Does this happen if window gets opened over U++ app? In that case the issue is most likely caused by delayed rendering of underlying window.

Is it only delay, or you have to move the window to make it look right?

Mirek

Subject: Re: Question to UPP developers: the issue with windows rendering (Windows XP)
Posted by porto on Wed, 13 Oct 2010 18:01:33 GMT
View Forum Message <> Reply to Message

Quote:Does this happen if window gets opened over U++ app?

No, this happens in any case (for new UPP windows too) whith every UPP app, but sometimes it's hard to notice.

Quote:Is it only delay, or you have to move the window to make it look right?
It's only delay.

I'll say it again, the issue arises on an different computers running Windows XP. Native WinAPI apps or e.g. apps that use QT don't have this issue. Perhaps it is not necessary to pay your attention, but I would like to UPP application looked more natively in Windows.

Does not anyone notice this issue? I noticed this around 3 years ago when I first meet whith UPP. This is a wonderful framework!

Subject: Re: Question to UPP developers: the issue with windows rendering (Windows XP)
Posted by cbpporter on Wed, 13 Oct 2010 18:12:03 GMT
View Forum Message <> Reply to Message

No, never noticed it and I have an eye for things like this. Are your systems new or old? Do they have the same video card/driver?

And a question for Mirek: does U++ create the underlying WinAPI frame in one go? I am sure you are aware that for some properties, updating them is not enough and you have to "recreate" the HWND. I ran into this problem with my ancient WinAPI wrapper. This doesn't affect us that much because our widgets don't wrap native ones, but still maybe it is related to this?

Subject: Re: Question to UPP developers: the issue with windows rendering (Windows XP)
Posted by porto on Wed, 13 Oct 2010 18:23:21 GMT
View Forum Message <> Reply to Message

One of my systems is a netbook ASUS Eee PC 1001P, the second system is a modern desktop with discrete video (NVidia GSO 9600). My third system is 4 years old (Pentium 4 3Ghz, int. video). But I saw this issue on other systems. Every system I tried runs Windows XP SP3.
Now this issue is mystycal for me...

For Mirek: May be a delay happens at the stage of initializing the window? Because when a window is moving, or when it is minimizing / maximizing issue doesn't occur.
And a silly assumption: all of my systems are running Russian edition of Windows XP. Can there be a reason for this issue?

Subject: Re: Question to UPP developers: the issue with windows rendering (Windows XP)

Posted by porto on Thu, 14 Oct 2010 08:45:23 GMT
View Forum Message <> Reply to Message

I found a better way to see this issue.
Download this app (it slows down the CPU):
http://www.cpukiller.com/cpukil305.zip
install it, set the "Slow Down Factor" (I set it on 95%), start slowing down CPU. Now, this issue will be visible very well.
I observed as native applications and applications written using QT. They did not have this defect.
If no one will confirm this issue, I will perform yet another experiment: I will install a clean copy of Windows XP SP3 Eng in VirtualBox and try to notice this issue there.

Subject: Re: Question to UPP developers: the issue with windows rendering (Windows XP)
Posted by mr_ped on Thu, 14 Oct 2010 09:13:12 GMT
View Forum Message <> Reply to Message

Don't you use some special theme or something like windows blinds?

I'm using "win classic" (like Win2000) theme, so I have sharp corners and no problems at all.

Subject: Re: Question to UPP developers: the issue with windows rendering (Windows XP)
Posted by porto on Thu, 14 Oct 2010 09:22:55 GMT
View Forum Message <> Reply to Message

For mr_ped: I am using standart WinXP Luna theme. When I'm switching theme to "classic", I don't see this issue too.
What is your OS? If it's Windows XP, can you try to use "Luna" theme and "Cpukiller3"?

Subject: Re: Question to UPP developers: the issue with windows rendering (Windows XP)
Posted by porto on Thu, 14 Oct 2010 16:48:36 GMT
View Forum Message <> Reply to Message

I have already started to doubt myself, because no one notices the issue. I performed an experiment with VirtualBox and clean copy of Windos XP SP3 Eng. Now I can confidently say: the issue is exist. Download the attached files, open them in VirtualDub and see the range of frames from the names of these files. The first file:

Subject: Re: Question to UPP developers: the issue with windows rendering (Windows XP)

Posted by dolik.rce on Thu, 14 Oct 2010 17:42:51 GMT
View Forum Message <> Reply to Message

From your videos, it looks to me like it only happens when there is a window behind the "wrong rendered" one and even only when the app is busy with something else then repainting. To be sure could you try to find out if this effect happens when you open some U++ window on top of any other application? IMHO the bottom window reacts slowly on the requests to repaint the area of rounded corners and you can notice it (that also explains why it takes longer if you keep the cpu busy with cpukill). I'm not an expert, but I am afraid there is most likely no solution if that's the case...

Honza

Subject: Re: Question to UPP developers: the issue with windows rendering (Windows XP)
Posted by porto on Thu, 14 Oct 2010 18:12:17 GMT
View Forum Message <> Reply to Message

dolik.rce wrote on Thu, 14 October 2010 20:42To be sure could you try to find out if this effect happens when you open some U++ window on top of any other application?
Yes, it happens too.
Thank you for your comment. Before, I sometimes noticed that the UPP applications are opened on a clean desktop had this issue.
I still hope that a solution must be, because in other frameworks I didn't notice anything like it.

Subject: Re: Question to UPP developers: the issue with windows rendering (Windows XP)
Posted by mirek on Fri, 15 Oct 2010 11:53:30 GMT
View Forum Message <> Reply to Message

In this very case, it is a Windows problem - that applet to remove apps is responsible repainting itself.

I would say if you would move the prompt somewhere, where would be whole black areas behind - because applet is busy and does not process WM_PAINT painting requests...

Subject: Re: Question to UPP developers: the issue with windows rendering (Windows XP)
Posted by mirek on Fri, 15 Oct 2010 11:55:04 GMT
View Forum Message <> Reply to Message

P.S.: Do not post .avi files into the forum. I guess if you need to post video, put it to the youtube and link it...

Subject: Re: Question to UPP developers: the issue with windows rendering (Windows XP)
Posted by porto on Fri, 15 Oct 2010 12:20:21 GMT
View Forum Message <> Reply to Message

But what about the cases described above? Is it possible to fix this issue?

P.S. Sorry for posting .avi files.

Subject: Re: Question to UPP developers: the issue with windows rendering (Windows XP)
Posted by mirek on Fri, 15 Oct 2010 12:31:30 GMT
View Forum Message <> Reply to Message

porto wrote on Fri, 15 October 2010 08:20I understand. But what about the cases described above? Is it possible to fix this issue?

P.S. Sorry for posting .avi files.

I am unable to replay the video, but with the screenshot, the problem is not with U++.

Maybe you have not noticed the same issue with other apps, but that is just coincidence. The problem is not with top window, but with window behind...

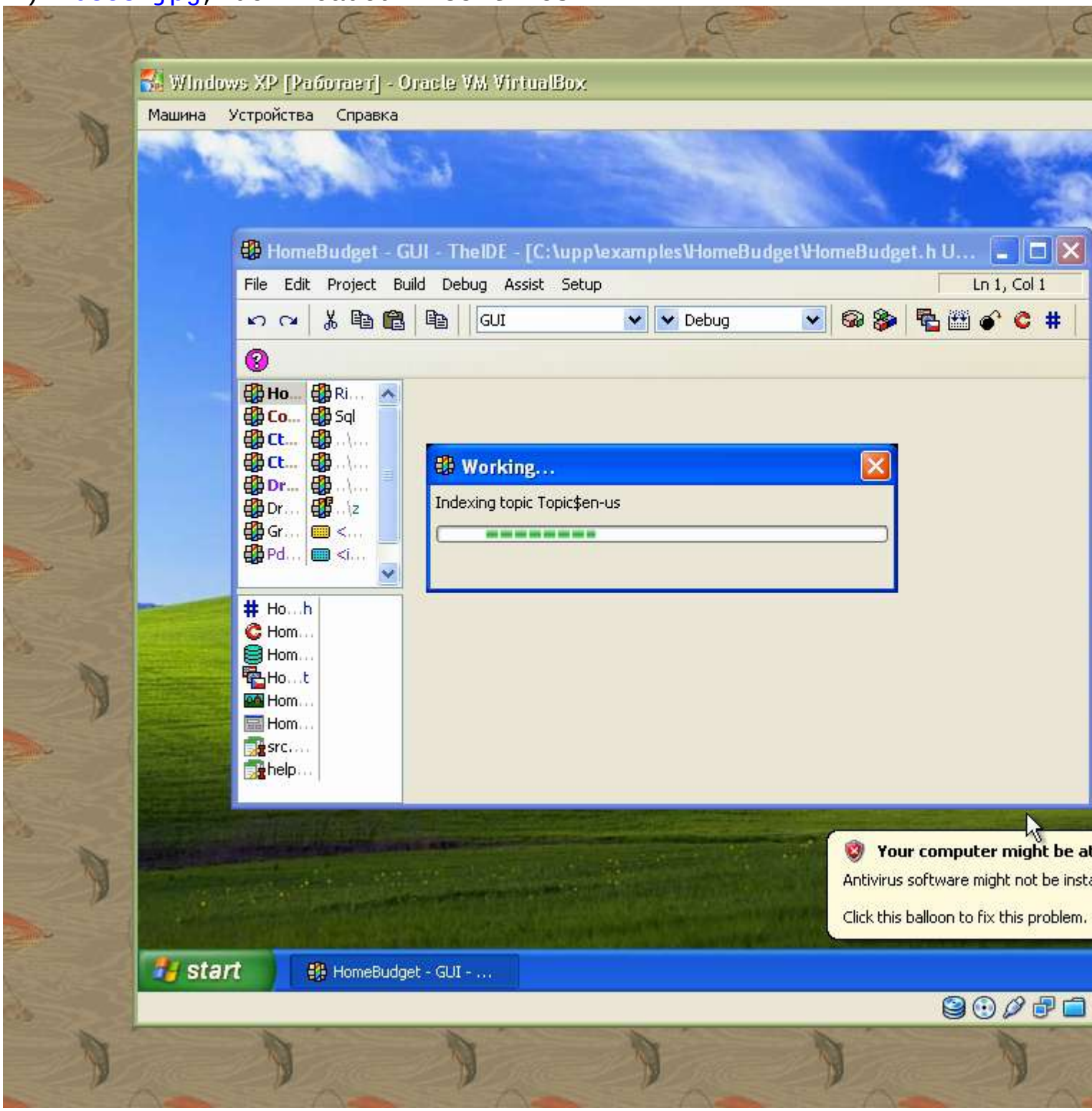
Subject: Re: Question to UPP developers: the issue with windows rendering (Windows XP)
Posted by porto on Fri, 15 Oct 2010 12:47:18 GMT
View Forum Message <> Reply to Message

I have examined in great detail other applications - native, QT apps, and I did not notice anything similar.
There is one screenshot from my video. (VirtualBox+clean WinXP)
In this case, even if the problem with window behind - it's UPP app's window too.

If you want to notice it, try:
Quote:I found a better way to see this issue.
Download this app (it slows down the CPU):
http://www.cpukiller.com/cpukil305.zip
install it, set the "Slow Down Factor" (I set it on 95%), start slowing down CPU. Now, this issue will be visible very well.

File Attachments

1) f0353.jpg, downloaded 1435 times



Subject: Re: Question to UPP developers: the issue with windows rendering (Windows XP)
Posted by porto on Fri, 15 Oct 2010 14:37:17 GMT

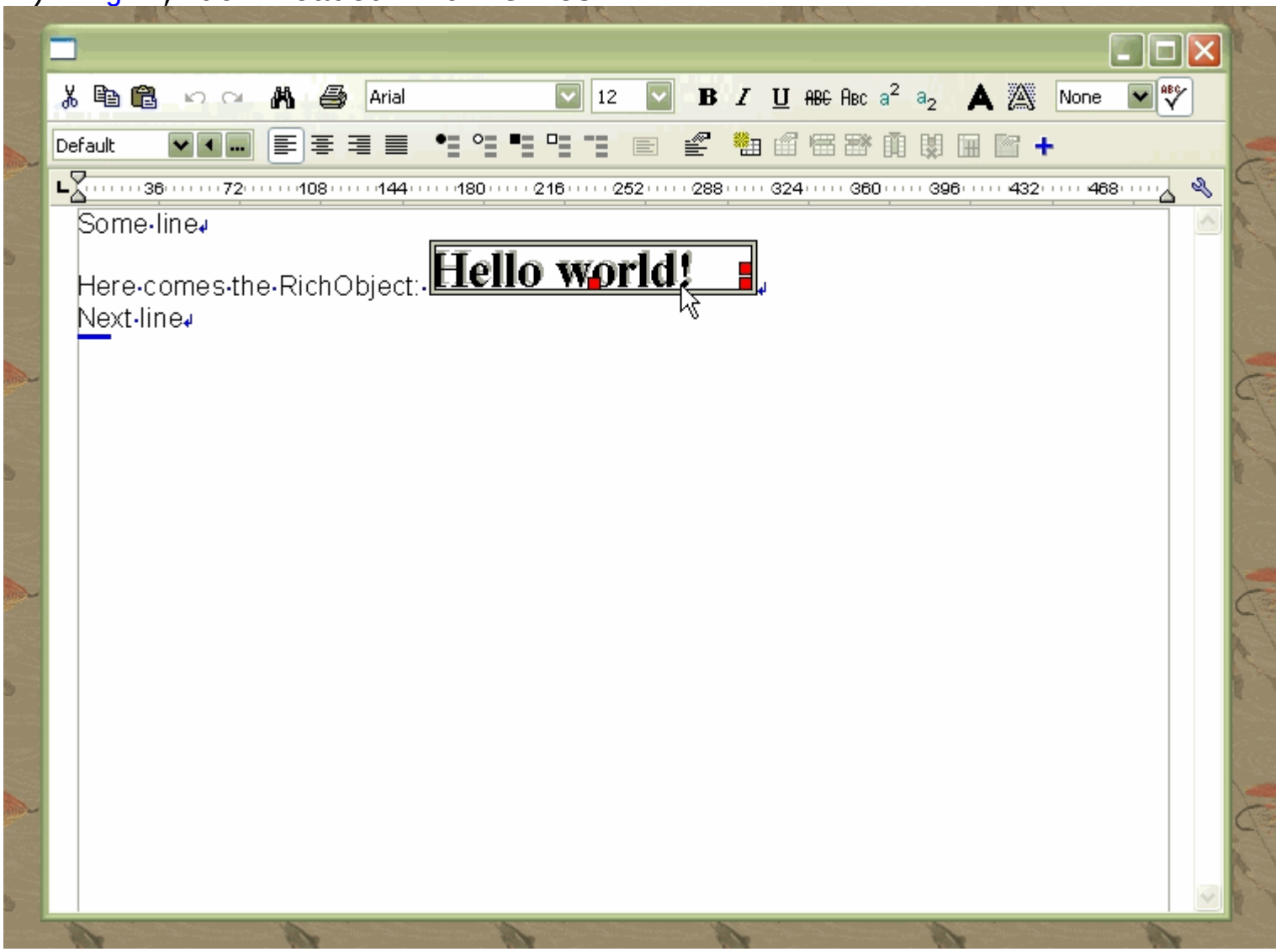
View Forum Message <> Reply to Message

I decided to make another animation to explain the issue:

This is an example application "RichTextObject" from UPP.
But if you think this is not a problem, never mind.

File Attachments

1) 1.gif, downloaded 1492 times



Subject: Re: Question to UPP developers: the issue with windows rendering (Windows XP)
Posted by mirek on Sat, 16 Oct 2010 06:24:12 GMT
View Forum Message <> Reply to Message

I know what the issue is, but I am not sure how to fix it. Win32 has control about it.

Subject: Re: Question to UPP developers: the issue with windows rendering (Windows XP)
Posted by porto on Sat, 16 Oct 2010 06:38:26 GMT
View Forum Message <> Reply to Message

And the last question. Why I can't notice something like this issue with other apps (WinAPI native, Qt etc)? This is an animation of cold start UPP application:

P.S. Thanks for your replies. Sorry for my fault-finding and taken away your valuable time.

File Attachments
1) nnnnnnnnn.gif, downloaded 1429 times



Subject: Re: Question to UPP developers: the issue with windows rendering (Windows XP)
Posted by mirek on Sat, 16 Oct 2010 08:01:48 GMT
View Forum Message <> Reply to Message

porto wrote on Sat, 16 October 2010 02:38And the last question. Why I can't notice something like this issue with other apps?

My guess: Because you are not trying hard enough...

Quote:
This is an animation of cold start UPP application:

P.S. Thanks for your replies. Sorry for my fault-finding and taken away your valuable time.

Well, to be absolutely sure, I have tried to reproduce the problem on slowest computer I have around (Asus EEE with 1.6Ghz Atom, WinXP), found nothing.

My guess is that the effect is caused by particular setup of your machine.

Actually, within WinXP, this effect is normal - but with normal PC, it is so fast that black areas go away before they are send to the monitor.

For some reason, on your machine it is slower. Well, it CAN becaused by some particular set of flags we are using in call to CreateWindow or something like that, however we are not doing anything "illegal", just using normal Win32 API calls.

I would even tried to mingle with these flags to see if I can solve the problem for you, the problem is I cannot even reproduce the damned thing.

You might also consider upgrading your video driver...

Subject: Re: Question to UPP developers: the issue with windows rendering (Windows XP)
Posted by mirek on Sat, 16 Oct 2010 08:24:29 GMT
View Forum Message <> Reply to Message

OK, I did some test using CpuKiller (btw, useful utility, thanks), setting it at 98%.

It is true that U++ based apps show black corners. The same thing is true for following well known software:

Adobe Reader
IrfanView
Total Commander
DVD Shrink

(stopped testing there).

If Adobe does not see this as problem, neither do I.

Just technical sidenote, to me it looks like technical issue behind the problem is that it depends on the way how the main window is composed. If it has a lot of child widgets, Win32 split painting widget by widget to multiple messages and this way some messages in multitasking system reach

other windows. U++ is highly optimized there and paints the window in single pass.

You can even notice that if you slow down things, those windows that do not show the black corners generally paint by parts and for quite a longer time...

I guess there is little I can add here - I did not want to dismiss the issue, but this really is non-issue...

Subject: Re: Question to UPP developers: the issue with windows rendering (Windows XP)
Posted by porto on Sat, 16 Oct 2010 08:42:08 GMT
View Forum Message <> Reply to Message

Thanks, you gave a convincing explanation. Once again sorry for the time taken away from you and I wish you success in U++ development, it's a wonderful framework.

Subject: Re: Question to UPP developers: the issue with windows rendering (Windows XP)
Posted by porto on Mon, 25 Oct 2010 14:11:46 GMT
View Forum Message <> Reply to Message

Delete this message, please

Subject: Re: Question to UPP developers: the issue with windows rendering (Windows XP)
Posted by porto on Tue, 26 Oct 2010 07:56:53 GMT
View Forum Message <> Reply to Message

I do not know why, but this dirty hack COMPLETELY eliminates this "issue":

```
void Ctrl::Create0(Ctrl::CreateBox *cr)
{
 GuiLock __;
 cr->exstyle = cr->exstyle | WS_EX_COMPOSITED;
```

Subject: Re: Question to UPP developers: the issue with windows rendering (Windows XP)
Posted by porto on Tue, 26 Oct 2010 09:04:45 GMT
View Forum Message <> Reply to Message

But this is not the solution. Now it reduces a performance when I dragging any windows on top of

UPP applications.
Partially solves my problem is forced set for each window CS_SAVEBITS style. But in this case, sometimes the "issue" is visible, for example, when TheIDE parses the source files when I open any project ...
I will dig further.

Subject: Re: Question to UPP developers: the issue with windows rendering (Windows XP)
Posted by porto on Wed, 27 Oct 2010 12:56:16 GMT
View Forum Message <> Reply to Message

It seems a simple solution was found. In a method:
void Ctrl:: Create0 (Ctrl:: CreateBox * cr) (Win32Wnd.cpp)
Before showing the window, we add an extended style:
cr->exstyle = cr->exstyle | WS_EX_COMPOSITED;
After the window is shown, remove it:
::SetWindowLong(top->hwnd, GWL_EXSTYLE, cr->exstyle ^ WS_EX_COMPOSITED);
It would be nice before to check which version of Windows we have - do it if we have Windows XP or higher.

These simple changes fixed the problem, and the performance still the same.

Can I hope that someone of the developers will make these (or more correct than mine) changes to one of the following buids of UPP?

Subject: Re: Question to UPP developers: the issue with windows rendering (Windows XP)
Posted by mirek on Wed, 27 Oct 2010 17:37:04 GMT
View Forum Message <> Reply to Message

Hey, nice digging It is sort of extreme workaround using some sideffect, but why not.

Do you have it as complete patch? I mean, where have you put

::SetWindowLong(top->hwnd, GWL_EXSTYLE, cr->exstyle ^ WS_EX_COMPOSITED);

?

Subject: Re: Question to UPP developers: the issue with windows rendering (Windows XP)
Posted by porto on Wed, 27 Oct 2010 19:44:23 GMT
View Forum Message <> Reply to Message

I have put it after this line:

::ShowWindow(top->hwnd, visible ? cr->show : SW_HIDE);

Subject: Re: Question to UPP developers: the issue with windows rendering (Windows XP)
Posted by porto on Sat, 30 Oct 2010 11:45:30 GMT
View Forum Message <> Reply to Message

There was found, I think, a good way to get away of this "issue" (the temporary addition to an extended style of windows a flag WS_EX_KOMPOSITED *) - but I want to find the reason of it.
Do not judge strictly an amateur in programming.
After a series of experiments with MS Spy++ utility I made some conclusions, this "issue" has two causes:
1. Child windows of traditional Windows applications have a flag of style CS_SAVEBITS. Child windows of UPP apps, this style does not have (or have, but not all).
After the forced set of this flag to all windows, "issue" only occurs if UPP application is busy, for example when I opening a package in TheIDE or doing "Rescan Code".
2. If UPP app does not busy (idle) and it creates the child window, the parent window is not repainting. But if UPP app is busy and it creates the child window, in this case, for some reason is starting repainting of the parent window or region of it, and CS_SAVEBITS is no longer helps.
If in this case the parent window or region of it does not repaint and all the child windows would have a flag CS_SAVEBITS (like in traditional applications), then the "issue" would not appear (IMHO).
This was only my best guess, do not take very seriously.

* With WS_EX_COMPOSITED set, all descendants of a window get bottom-to-top painting order using double-buffering. Bottom-to-top painting order allows a descendent window to have translucency (alpha) and transparency (color-key) effects, but only if the descendent window also has the WS_EX_TRANSPARENT bit set. Double-buffering allows the window and its descendents to be painted without flicker.

Subject: Re: Question to UPP developers: the issue with windows rendering (Windows XP)
Posted by mirek on Mon, 01 Nov 2010 08:56:02 GMT
View Forum Message <> Reply to Message

porto wrote on Wed, 27 October 2010 15:44I have put it after this line:
::ShowWindow(top->hwnd, visible ? cr->show : SW_HIDE);

Lines have numbers. Files have names.

Is it so hard to cite the filename and the number of line?

Eventually quoting couple of sourounding lines here (or the whole function, if that is not too big)?

Subject: Re: Question to UPP developers: the issue with windows rendering (Windows XP)
Posted by mirek on Mon, 01 Nov 2010 09:03:05 GMT
View Forum Message <> Reply to Message

porto wrote on Sat, 30 October 2010 07:45There was found, I think, a good way to get away of this "issue" (the temporary addition to an extended style of windows a flag WS_EX_KOMPOSITED *) - but I want to find the reason of it.
Do not judge strictly an amateur in programming.
After a series of experiments with MS Spy++ utility I made some conclusions, this "issue" has two causes:
1. Child windows of traditional Windows applications have a flag of style CS_SAVEBITS. Child windows of UPP apps, this style does not have (or have, but not all).

I guess you have got the term "child window" wrong. CS_SAVEBITS is intended for popups.

Quote:
After the forced set of this flag to all windows, "issue" only occurs if UPP application is busy, for example when I opening a package in TheIDE or doing "Rescan Code".

For one thing, there are no child windows in U++... (Except DHCtrl descendants, but that is a special case).

My guess is that WS_EX_COMPOSITED only helps accidentally - it changes repainting process of windows and as side-effect, the effect is gone. In the same time, it is not unlikely that this will still depend on actual windows version, graphics driver etc...

Subject: Re: Question to UPP developers: the issue with windows rendering (Windows XP)
Posted by porto on Mon, 01 Nov 2010 09:21:19 GMT
View Forum Message <> Reply to Message

I made changes in file Wnd32Wnd.cpp, in method:
void Ctrl::Create0(Ctrl::CreateBox *cr) (begin at line 459)

Added lines are marked in bold:

```
void Ctrl::Create0(Ctrl::CreateBox *cr)
{
 GuiLock __;
       cr->exstyle = cr->exstyle | WS_EX_COMPOSITED;
       ... some code ...

 ::ShowWindow(top->hwnd, visible ? cr->show : SW_HIDE);
```

```
	::SetWindowLong(top->hwnd, GWL_EXSTYLE, cr->exstyle ^
WS_EX_COMPOSITED);
	... some code ...
}
```

I really want to find out the cause of this effect. So I want to ask you, if UPP apps windows need repainting some area, they are fully updated or update only the necessary area?

I have achieved the same effect in WinAPI application by locking update in underlying window by LockWindowUpdate():

```
  HWND hWnd, hWnd2;

  hInst = hInstance;

  hWnd = CreateWindow(szWindowClass, szTitle, WS_OVERLAPPEDWINDOW,
    CW_USEDEFAULT, 0, CW_USEDEFAULT, 0, NULL, NULL, hInstance, NULL);
  hWnd2 = CreateWindow(szWindowClass, szTitle, WS_OVERLAPPEDWINDOW,
    CW_USEDEFAULT, 0, CW_USEDEFAULT, 0, NULL, NULL, hInstance, NULL);

  ShowWindow(hWnd, SW_MAXIMAZED);
  LockWindowUpdate(hWnd);
  ShowWindow(hWnd2, SW_NORMAL);
  Sleep(1000);
```

Without LockWindowUpdate() in this code the effect is gone.
My another suggestion (possibly incorrect again): in UPP apps for some reason MS Windows are not immediately sends WM_PAIN command to underlying windows.

I also tried to make the following changes: after the new window is shown, I just completely update the parent window. This is also a solution to the problem, but why Windows does not update a necessary area of parent window automatically not clear to me...

Subject: Re: Question to UPP developers: the issue with windows rendering (Windows XP)
Posted by porto on Wed, 03 Nov 2010 16:07:45 GMT
View Forum Message <> Reply to Message

As I understood from the description in MSDN the flag of extended style "WS_EX_COMPOSITED" activates the double-buffering at the OS level (it's supported minimum in WinXP), so unlikely it will depend on the graphics card drivers. About making changes in UPP if you agree, I think the better off the flag after the full appearance of the window. About changes in UPP, if you agree, I think it's better to set off the flag after full window rendering instead after non

client area rendering, it will give a smoother draw them.
You can make changes like that e.g., or more correctly: (added lines are marked in bold):
1. In file Wnd32Wnd.cpp, at begin of the method (line 459):
void Ctrl::Create0(Ctrl::CreateBox *cr)
{
cr->exstyle = cr->exstyle | WS_EX_COMPOSITED;
...some code...
}
2. In file TopWin32.cpp, at end of the method (line 173):
void TopWindow::Open(HWND hwnd)
{
...some code...
::SetWindowLong(hwnd, GWL_EXSTYLE, ::GetWindowLong(hwnd, GWL_EXSTYLE) ^
WS_EX_COMPOSITED);
}

Subject: Re: Question to UPP developers: the issue with windows rendering (Windows XP)
Posted by mirek on Thu, 04 Nov 2010 13:27:43 GMT
View Forum Message <> Reply to Message

porto wrote on Mon, 01 November 2010 05:21
I also tried to make the following changes: after the new window is shown, I just completely update the parent window. This is also a solution to the problem, but why Windows does not update a necessary area of parent window automatically not clear to me...

It does. Please note that everythings gets updated in the end, even when the effect is visible. What so much bothers you is the pause.

But all these things are affected by timing. Win32 simply sends some messages first, some others later, that is all. Accidentally, for U++ or IrfanView, with certain HW and system config, it means that messages required to repaint nonclient area of window are sent later.

Subject: Re: Question to UPP developers: the issue with windows rendering (Windows XP)
Posted by mirek on Thu, 04 Nov 2010 13:31:09 GMT
View Forum Message <> Reply to Message

porto wrote on Wed, 03 November 2010 12:07As I understood from the description in MSDN the flag of extended style "WS_EX_COMPOSITED" activates the double-buffering at the OS level (it's supported minimum in WinXP), so unlikely it will depend on the graphics card drivers. About making changes in UPP if you agree, I think the better off the flag after the full appearance of the window. About changes in UPP, if you agree, I

Frankly, I am undecided. Double-buffering would take toll on total performance (it eats quite some memory), is it worth to introduce this to the library just to fix repainting lag on your specific machine? Besides, it is low-level machines we are trying to fix here...

I guess if there would be some intelligent fix I might be less undecided, but COMPOSITED is quite heave weaponry for the task.

Subject: Re: Question to UPP developers: the issue with windows rendering (Windows XP)
Posted by porto on Thu, 04 Nov 2010 14:18:46 GMT
View Forum Message <> Reply to Message

I have seen this effect on three different machines. The most powerful of them has a modern dual core 3Ghz processor and nVidia GSO 9600 graphics card, and even there the effect is sometimes seen.
But you're right. Maybe in the future I will find a clever solution. Thanks for your answers.

Subject: Re: Question to UPP developers: the issue with windows rendering (Windows XP)
Posted by porto on Thu, 11 Nov 2010 13:34:59 GMT
View Forum Message <> Reply to Message

Hello, Mirek!
I think finally I found the cause of this issue. I understand that perhaps no one, does not bother about this, but I ask a little more of your attention, if it possible.
When a little digging into the source code of apps that are deprived of this problem, I discovered that these apps after the window creation (but before it is displayed when the window is still invisible) change the size and/or position (perhaps when it placed the child elements, etc.) I don't know why, but in this case redrawing a non-client area of the window does not by underlying window, but it does by Windows itself. When UPP application creates a window, it immediately set the final size and position.
I've made small changes to the source code, which instead of the double buffering does not affect the performance and don't eat more memory then original code. The idea is simple: When we create a window (it is not yet visible), we set to zero (or another value that is different from the original) its size and position. And then, before the window is displaying on the screen, we change size and position to the original value by MoveWindow() function.
Below I post the complete changed method. It is in the file "Win32Wnd.cpp". The method begin at line 459 (added lines are marked whith "// added" comment, changed lines are marked whith "// changed" comment):

```
void Ctrl::Create0(Ctrl::CreateBox *cr)
{
 GuiLock __;
 if(cr->style & WS_VISIBLE)
```

```
  cr->style ^= WS_VISIBLE; //added
 ASSERT(IsMainThread());
 LLOG("Ctrl::Create(parent = " << (void *)parent << ") in " <<UPP::Name(this) << BeginIndent);
 ASSERT(!IsChild() && !IsOpen());
 Rect r = GetRect();
 AdjustWindowRectEx(r, cr->style, FALSE, cr->exstyle);
 isopen = true;
 top = new Top;
 ASSERT(!cr->parent || IsWindow(cr->parent));
 if(!IsWinXP())
  cr->dropshadow = false;
#ifdef PLATFORM_WINCE
  if(parent)
   top->hwnd = CreateWindowExW(cr->exstyle,
                               cr->savebits ? cr->dropshadow ? L"UPP-CLASS-SB-DS-W" :
L"UPP-CLASS-SB-W"
                                            : cr->dropshadow ? L"UPP-CLASS-DS-W"    : L"UPP-CLASS-W",
                               L"", cr->style, 0, 0, 0, 0,
                               cr->parent, NULL, hInstance, this); // changed
  else
   top->hwnd = CreateWindowW(L"UPP-CLASS-W",
                             L"", WS_VISIBLE, CW_USEDEFAULT, CW_USEDEFAULT,
CW_USEDEFAULT, CW_USEDEFAULT,
                             cr->parent, NULL, hInstance, this);
#else
 if(IsWinNT() && (!cr->parent || IsWindowUnicode(cr->parent)))
  top->hwnd = CreateWindowExW(cr->exstyle,
                              cr->savebits ? cr->dropshadow ? L"UPP-CLASS-SB-DS-W" :
L"UPP-CLASS-SB-W"
                                           : cr->dropshadow ? L"UPP-CLASS-DS-W"    : L"UPP-CLASS-W",
                              L"", cr->style, 0, 0, 0, 0,
                              cr->parent, NULL, hInstance, this); // changed
 else
  top->hwnd = CreateWindowEx(cr->exstyle,
                             cr->savebits ? cr->dropshadow ? "UPP-CLASS-SB-DS-A" :
"UPP-CLASS-SB-A"
                                          : cr->dropshadow ? "UPP-CLASS-DS-A"    : "UPP-CLASS-A",
                             "", cr->style, 0, 0, 0, 0,
                             cr->parent, NULL, hInstance, this); // changed
#endif

 inloop = false;

 ASSERT(top->hwnd);

 ::MoveWindow(top->hwnd, r.left, r.top, r.Width(), r.Height(), false); //added
 ::ShowWindow(top->hwnd, visible ? cr->show : SW_HIDE);
// ::UpdateWindow(hwnd);
```

```
 StateH(OPEN);
 LLOG(EndIndent << "//Ctrl::Create in " <<UPP::Name(this));
 RegisterDragDrop(top->hwnd, (LPDROPTARGET) (top->dndtgt = NewUDropTarget(this)));
 CancelMode();
 RefreshLayoutDeep();
}
```

I also want to explain this code:

```
if(cr->style & WS_VISIBLE)
  cr->style ^= WS_VISIBLE;
```

When a window is displayed, OS sets a WS_VISIBLE flag to its style. In file "TopWin32.cpp", in method void TopWindow::SyncCaption0() we have this code:

```
if(hwnd) {
  style = ::GetWindowLong(hwnd, GWL_STYLE);
  exstyle = ::GetWindowLong(hwnd, GWL_EXSTYLE);
 }
```

After the window is displayed and the flag has been set, we initialize the "style" variable whith a style info was readed from the window, but the window has a WS_VISIBLE flag! If the window is destroyed, and its class instance is still exists whith set WS_VISIBLE style flag. If now we again call for that window method CreateBox() function CreateWindowEx(). This function is show a window during creation is WS_VISIBLE flag was set, and ShowWindow() function a second time will show already shown window! Maybe you can fix it on other level, I have no experience to do this.

Is it possible to make changes similar to mine in the official build?

P.S. Sorry for bad English.

Subject: Re: Question to UPP developers: the issue with windows rendering (Windows XP)
Posted by mirek on Wed, 17 Nov 2010 09:19:01 GMT
View Forum Message <> Reply to Message

porto wrote on Thu, 11 November 2010 08:34Hello, Mirek!
I think finally I found the cause of this issue. I understand that perhaps no one, does not bother about this, but I ask a little more of your attention, if it possible.
When a little digging into the source code of apps that are deprived of this problem, I discovered that these apps after the window creation (but before it is displayed when the window is still invisible) change the size and/or position (perhaps when it placed the child elements, etc.) I don't know why, but in this case redrawing a non-client area of the window does not by underlying window, but it does by Windows itself. When UPP application creates a window, it immediately set the final size and position.
I've made small changes to the source code, which instead of the double buffering does not affect the performance and don't eat more memory then original code. The idea is simple: When we create a window (it is not yet visible), we set to zero (or another value that is different from the original) its size and position. And then, before the window is displaying on the screen, we change

size and position to the original value by MoveWindow() function.
Below I post the complete changed method. It is in the file "Win32Wnd.cpp". The method begin at line 459 (added lines are marked whith "// added" comment, changed lines are marked whith "// changed" comment):

```
void Ctrl::Create0(Ctrl::CreateBox *cr)
{
 GuiLock __;
 if(cr->style & WS_VISIBLE)
  cr->style ^= WS_VISIBLE; //added
 ASSERT(IsMainThread());
 LLOG("Ctrl::Create(parent = " << (void *)parent << ") in " <<UPP::Name(this) << BeginIndent);
 ASSERT(!IsChild() && !IsOpen());
 Rect r = GetRect();
 AdjustWindowRectEx(r, cr->style, FALSE, cr->exstyle);
 isopen = true;
 top = new Top;
 ASSERT(!cr->parent || IsWindow(cr->parent));
 if(!IsWinXP())
  cr->dropshadow = false;
#ifdef PLATFORM_WINCE
  if(parent)
   top->hwnd = CreateWindowExW(cr->exstyle,
                               cr->savebits ? cr->dropshadow ? L"UPP-CLASS-SB-DS-W" :
L"UPP-CLASS-SB-W"
                                            : cr->dropshadow ? L"UPP-CLASS-DS-W"    : L"UPP-CLASS-W",
                               L"", cr->style, 0, 0, 0, 0,
                               cr->parent, NULL, hInstance, this); // changed
  else
   top->hwnd = CreateWindowW(L"UPP-CLASS-W",
                             L"", WS_VISIBLE, CW_USEDEFAULT, CW_USEDEFAULT,
CW_USEDEFAULT, CW_USEDEFAULT,
                             cr->parent, NULL, hInstance, this);
#else
 if(IsWinNT() && (!cr->parent || IsWindowUnicode(cr->parent)))
  top->hwnd = CreateWindowExW(cr->exstyle,
                              cr->savebits ? cr->dropshadow ? L"UPP-CLASS-SB-DS-W" :
L"UPP-CLASS-SB-W"
                                           : cr->dropshadow ? L"UPP-CLASS-DS-W"    : L"UPP-CLASS-W",
                              L"", cr->style, 0, 0, 0, 0,
                              cr->parent, NULL, hInstance, this); // changed
 else
  top->hwnd = CreateWindowEx(cr->exstyle,
                             cr->savebits ? cr->dropshadow ? "UPP-CLASS-SB-DS-A" :
"UPP-CLASS-SB-A"
                                          : cr->dropshadow ? "UPP-CLASS-DS-A"    : "UPP-CLASS-A",
                             "", cr->style, 0, 0, 0, 0,
                             cr->parent, NULL, hInstance, this); // changed
#endif
```

```
 inloop = false;

 ASSERT(top->hwnd);

 ::MoveWindow(top->hwnd, r.left, r.top, r.Width(), r.Height(), false); //added
 ::ShowWindow(top->hwnd, visible ? cr->show : SW_HIDE);
// ::UpdateWindow(hwnd);
 StateH(OPEN);
 LLOG(EndIndent << "//Ctrl::Create in " <<UPP::Name(this));
 RegisterDragDrop(top->hwnd, (LPDROPTARGET) (top->dndtgt = NewUDropTarget(this)));
 CancelMode();
 RefreshLayoutDeep();
}
```

I also want to explain this code:

```
if(cr->style & WS_VISIBLE)
  cr->style ^= WS_VISIBLE;
```

When a window is displayed, OS sets a WS_VISIBLE flag to its style. In file "TopWin32.cpp", in method void TopWindow::SyncCaption0() we have this code:

```
if(hwnd) {
  style = ::GetWindowLong(hwnd, GWL_STYLE);
  exstyle = ::GetWindowLong(hwnd, GWL_EXSTYLE);
 }
```

After the window is displayed and the flag has been set, we initialize the "style" variable whith a style info was readed from the window, but the window has a WS_VISIBLE flag! If the window is destroyed, and its class instance is still exists whith set WS_VISIBLE style flag. If now we again call for that window method CreateBox() function
Is it possible to make changes similar to mine in the official build?

P.S. Sorry for bad English.

This patch seems to be reasonable way to fix the problem.

However, I am not sure about fixing WS_VISIBLE issue... If the only effect to fix is:

[b]CreateWindowEx()[/b]. This function is show a window during creation is WS_VISIBLE flag was set, and [b]ShowWindow()[/b] function a second time will show already shown window! Maybe you can fix it on other level, I have no experience to do this.

then I really have to ask:

What is the problem with showing already shown window?

Subject: Re: Question to UPP developers: the issue with windows rendering (Windows XP)
Posted by porto on Wed, 17 Nov 2010 10:22:38 GMT
View Forum Message <> Reply to Message

Quote:then I really have to ask:

What is the problem with showing already shown window?
There is no problem. The fix works fine without that line. This is just for reassurance. I'm not an expert in WinAPI and am not sure that a window with zero size and position can not be seen on the screen under certain conditions. But if you are sure about it - that line is not needed.

Subject: Re: Question to UPP developers: the issue with windows rendering (Windows XP)
Posted by porto on Wed, 17 Nov 2010 10:36:24 GMT
View Forum Message <> Reply to Message

I just remembered, that line solves the problem that I created myself. Therefore, it is not needed.

Subject: Re: Question to UPP developers: the issue with windows rendering (Windows XP)
Posted by mirek on Sat, 20 Nov 2010 17:03:57 GMT
View Forum Message <> Reply to Message

Path applied.

Thank you. I guess with fix like this one, it is worth to be better than Adobe

Subject: Re: Question to UPP developers: the issue with windows rendering (Windows XP)
Posted by porto on Sat, 20 Nov 2010 20:10:26 GMT
View Forum Message <> Reply to Message

Quote:Path applied.
I'm very pleased. Thanks for that.
Quote:Thank you. I guess with fix like this one, it is worth to be better than Adobe

Subject: Re: Question to UPP developers: the issue with windows rendering (Windows XP)

Posted by mirek on Mon, 22 Nov 2010 18:07:39 GMT
View Forum Message <> Reply to Message

Bad news: While it might have fixed the issue of black corners 'flash', it introduced pretty bad artifact with Terminal Server sessions (some windows are now missing frame completely).

I am afraid I have to rollback it back for now.

Mirek

Subject: Re: Question to UPP developers: the issue with windows rendering (Windows XP)
Posted by porto on Mon, 22 Nov 2010 20:00:44 GMT
View Forum Message <> Reply to Message

I guess that adding in the beginning of the method of this line:
Quote:if(cr->style & WS_VISIBLE)
  cr->style ^= WS_VISIBLE;
will solve the problem.

Subject: Re: Question to UPP developers: the issue with windows rendering (Windows XP)
Posted by mirek on Wed, 24 Nov 2010 13:58:42 GMT
View Forum Message <> Reply to Message

porto wrote on Mon, 22 November 2010 15:00I guess that adding in the beginning of the method of this line:
Quote:if(cr->style & WS_VISIBLE)
  cr->style ^= WS_VISIBLE;
will solve the problem.

OK, with

cr->style &= ~WS_VISIBLE;

(the same fix, actually)

it seems to work fine. Will keep that - hopefully this is the real end of this issue

Mirek

Subject: Re: Question to UPP developers: the issue with windows rendering (Windows XP)
Posted by porto on Thu, 25 Nov 2010 07:12:44 GMT
View Forum Message <> Reply to Message

I hope so