Subject: Appearance problem in TabCtrl
Posted by qapko on Tue, 24 May 2011 14:25:59 GMT
View Forum Message <> Reply to Message

Hello,
I have appearance problem in TabCtrl on one machine running Windows XP using Ultimate++ 3211. Problem appears after setting transparent (opaque) image to a tab using tab.GetItem(index).SetImage(...). Then after switching through the tabs it looks like this:

The mess in the last tab is certainly a distorted square from the first tab.
I can't reproduce the problem in a small test application but after some research I finally found something interesting. In file ImageWin32.cpp I've changed this one line in Image::Data::PaintImp method and it helped:

```
 if(GetKind() == IMAGE_OPAQUE && paintcount == 0 && sr == Rect(sz) && IsWinNT() &&
w.IsGui()) {
  LTIMING("Image Opaque direct set");
  SetSurface(w, x, y, sz.cx, sz.cy, buffer);
//  paintcount++;  /* removing this line helped */
  return;
 }
```

I still do not understand the issue completely but if you have any idea as how should I continue with the research I would appreciate it.

Thank you for your help and have a nice day

Gabi

File Attachments
1) tabctrl.png, downloaded 1137 times

Subject: Re: Appearance problem in TabCtrl
Posted by mirek on Tue, 24 May 2011 15:37:11 GMT
View Forum Message <> Reply to Message

qapko wrote on Tue, 24 May 2011 10:25Hello,
I have appearance problem in TabCtrl on one machine running Windows XP using Ultimate++ 3211. Problem appears after setting transparent (opaque)

Is not transparent oposite of opaque?!

Quote:

```
 if(GetKind() == IMAGE_OPAQUE && paintcount == 0 && sr == Rect(sz) && IsWinNT() &&
w.IsGui()) {
  LTIMING("Image Opaque direct set");
  SetSurface(w, x, y, sz.cx, sz.cy, buffer);
//  paintcount++;  /* removing this line helped */
  return;
 }
```

Well, this piece of code is sort of optimization - if particular Image is painted only once, it is not transfered to BITMAP and painted directly ('SetSurface'). If painted for the second time, it gets coverted to BITMAP and painted that way, as we supposed that BITMAP is faster for repeated paints.

By removing the line, you force U++ to paint it always directly.

So, the first estimate is that Image is misclassified. Where is Image coming from? Is it .iml file, or is it something synthetised?

All in all, testcase would be fine...

Mirek

Subject: Re: Appearance problem in TabCtrl
Posted by qapko on Tue, 24 May 2011 18:42:28 GMT
View Forum Message <> Reply to Message

Hello,
thanks for the response. I meant the image is transparent in the sense that it has some pixels set to be transparent in the IML editor (the pixels around the square, namely). In fact I was confused with the IMAGE_OPAQUE constant in the "if" statement and haven't read relevant help section at all (my fault, and my bad english of course). So the image class for the square should be IMAGE_MASK? And the question is, why it have not that class? The image is from .iml file. I have this file loaded more than once under different IMAGECLASSes. Can this be a problem?
I tried to make some testcase, but in vain. Nor was I able to see the problem on another 4 PCs. One more information: While testing I was able to fix the problem by disabling the LRUcache (by changing the template).
I will look at it tomorow. Unfortunately the "bad PC" is the desktop at my office

Thank you for the inspiration.

Gabi

Subject: Re: Appearance problem in TabCtrl
Posted by qapko on Wed, 25 May 2011 13:00:26 GMT
View Forum Message <> Reply to Message

Hello Mirek,
I've done some investigation and have some new facts:
1. The square in the first tab is classified as IMAGE_MASK. I can prove that it is really displayed in "if(GetKind() == IMAGE_MASK) {...}" statement in PaintImp method.
2. If I dump buffer of the messed image, then it is OK even in the case the image appears distorted on the screen. So the image cache is OK, I guess.
3. But when I dump dcMem of the messed image after the line with HBITMAP o = (HBITMAP)::SelectObject(dcMem, sd.hbmp) in PaintImp method, within if(GetKind() == IMAGE_OPAQUE) {...} statement, the dump is distorted. Can I guess from this that the sd.hbmp itself is distorted?

Gabi

Subject: Re: Appearance problem in TabCtrl
Posted by qapko on Thu, 26 May 2011 08:05:58 GMT
View Forum Message <> Reply to Message

Hello Mirek,
I think I have a simple fix for the problem.I've attached changed ImageWin32.cpp.
On line 251 I've changed
 sd.hbmp = ::CreateCompatibleBitmap(dc, sz.cx, sz.cy);
to
 sd.hbmp = ::CreateCompatibleBitmap(dcMem, sz.cx, sz.cy);

At http://msdn.microsoft.com/en-us/library/dd183488%28v=vs.85%2 9.aspx it is written that:
If a DIB section, which is a bitmap created by the CreateDIBSection function, is selected into the device context identified by the hdc parameter, CreateCompatibleBitmap creates a DIB section.

I have no skill with Windows API at all, so my question is: Can this really be the issue? And if it is, aren't there in the code of Ultimate++ any other places with the same problem?

Gabi

File Attachments
1) ImageWin32.cpp, downloaded 649 times

Subject: Re: Appearance problem in TabCtrl
Posted by mirek on Mon, 30 May 2011 09:54:20 GMT
View Forum Message <> Reply to Message

qapko wrote on Thu, 26 May 2011 04:05Hello Mirek,
I think I have a simple fix for the problem.I've attached changed ImageWin32.cpp.
On line 251 I've changed
 sd.hbmp = ::CreateCompatibleBitmap(dc, sz.cx, sz.cy);
to
 sd.hbmp = ::CreateCompatibleBitmap(dcMem, sz.cx, sz.cy);

At http://msdn.microsoft.com/en-us/library/dd183488%28v=vs.85%2 9.aspx it is written that:
If a DIB section, which is a bitmap created by the CreateDIBSection function, is selected into the device context identified by the hdc parameter, CreateCompatibleBitmap creates a DIB section.

I have no skill with Windows API at all, so my question is: Can this really be the issue? And if it is, aren't there in the code of Ultimate++ any other places with the same problem?

Gabi

Well, if I am not mistaken, such a change would mean "deoptimalisation". What we want to get there is not a DIB section. DIB section is only used to transfer the data, but in resulting bitmap, we definitely want to have device dependent bitmap here.

Would it be at least possible to post .iml file here?

What about color depth in affected computer? That could easily be the difference....

Mirek

Subject: Re: Appearance problem in TabCtrl
Posted by qapko on Mon, 30 May 2011 12:17:21 GMT
View Forum Message <> Reply to Message

Yes, it is about "deoptimalisation" - but we can remove most of the CreateHBMP to save some time .
On all my computers I have 32bit color depth, but I tried to change it to 16bit at "the bad" computer and the problem disappeared. Strange...
I've attached the .iml file from the real application, it's the "Exclamation" symbol that causes the problem.
In attached "ImageWin32.cpp" on line 309 I've put the only two operations for if(GetKind() == IMAGE_MASK) statement to have the problem still appearing. If I remove the ::Delete operation (keeping only the ::CreateCompatibleBitmap), the problem disappears. That is the clue that brought me to the "deoptimalisation" fix.
In "ImageWin32_2.cpp" the problem is fixed by creating a bitmap that is not used at all but supposedly grabs the distortion to itself.

Gabi

File Attachments
1) att1.zip, downloaded 468 times

Subject: Re: Appearance problem in TabCtrl
Posted by mirek on Wed, 01 Jun 2011 18:42:24 GMT
View Forum Message <> Reply to Message

qapko wrote on Mon, 30 May 2011 08:17Yes, it is about "deoptimalisation" - but we can remove most of the CreateHBMP to save some time .
On all my computers I have 32bit color depth, but I tried to change it to 16bit at "the bad" computer and the problem disappeared. Strange...
I've attached the .iml file from the real application, it's the "Exclamation" symbol that causes the problem.
In attached "ImageWin32.cpp" on line 309 I've put the only two operations for if(GetKind() == IMAGE_MASK) statement to have the problem still appearing. If I remove the ::Delete operation (keeping only the ::CreateCompatibleBitmap), the problem disappears.

Really weird as you have basically removed painting for masked images...

Well, this code exists mostly unchanged for more than 7 years - and I see nothing wrong with it. You also seem to be unable to 'isolate' the issue in testcase. Given these fact I would say that there might be some nasty bug in your app that somehow generates the problem... Or maybe the computer is broken, even such things happen...

Mirek

Subject: Re: Appearance problem in TabCtrl
Posted by qapko on Thu, 02 Jun 2011 12:23:08 GMT
View Forum Message <> Reply to Message

Hello Mirek, thank you for your attention.
Now I too suppose that the problem is at some other place, not in the Ultimate++ library. I will post a massage if I would get some definitive result some day

Gabi

PS: I'm attaching the screenshot for the situation described in my last post - it's just interesting and I forgot to attach it before.

File Attachments
1) tabctrl2.PNG, downloaded 1148 times

Subject: Re: Appearance problem in TabCtrl
Posted by mirek on Fri, 03 Jun 2011 06:21:19 GMT
View Forum Message <> Reply to Message

Well, now looking at it... What OS is that? Win2k?

If it is WinXP (or later), it might be interesting to switch it to "XP" visual mode.

Mirek

Subject: Re: Appearance problem in TabCtrl
Posted by qapko on Fri, 03 Jun 2011 20:24:47 GMT
View Forum Message <> Reply to Message

Good question It is Windows XP and in XP visual mode everything is OK. At the beggining of the story after some study of the chamelon technology sources I've tried to swich to XP visual mode on "the bad" machine and the problem disapeared. I'm sorry I didn't mention it right in the first post. Anyway, on all 5 tested computers I have Classic visual mode set as default.
Meanwhile, I've started to isolate "the bug" in the sources of the application with some preliminary results but the way to the testcase is still long

Have a nice day

Gabi

Subject: Re: Appearance problem in TabCtrl
Posted by qapko on Mon, 06 Jun 2011 09:57:58 GMT
View Forum Message <> Reply to Message

Hello Mirek
I've prepared a testcase by cutting down the parts of the application. Please look at it.

Gabi

File Attachments
1) kmedis_testcase.zip, downloaded 467 times

Subject: Re: Appearance problem in TabCtrl
Posted by mirek on Tue, 07 Jun 2011 21:14:35 GMT
View Forum Message <> Reply to Message

Thanks. I see nothing wrong with testcase.

It is quite weird that NoBackground for TreeCtrl should have any effect, all it does is affecting color and painting color on TreeCtrl background - nothing to do with Images.

Hm, tried upgrading video driver?

Mirek

Subject: Re: Appearance problem in TabCtrl
Posted by qapko on Wed, 08 Jun 2011 08:18:36 GMT
View Forum Message <> Reply to Message

I have the newest available video driver on the PC
Now I see that the problem has nothing to do with TreeCtrl. I was able to reproduce it using StaticText at the place of TreeCtrl (see attachment). The problem is probably in video card or its driver. The weird thing is that no other application on the PC has such displaying issues. So that goes...
Anyway, thanks for your help and inspiration. For me it was a good lesson to hunt "the bug" through the sources of Ultimate++!

Gabi

File Attachments
1) kmedis_testcase2.zip, downloaded 482 times

Subject: Re: Appearance problem in TabCtrl
Posted by mirek on Wed, 08 Jun 2011 17:55:24 GMT
View Forum Message <> Reply to Message

qapko wrote on Wed, 08 June 2011 04:18I have the newest available video driver on the PC
Now I see that the problem has nothing to do with TreeCtrl. I was able to reproduce it using StaticText at the place of TreeCtrl (see attachment). The problem is probably in video card or its driver. The weird thing is that no other application on the PC has such displaying issues. So that goes...
Anyway, thanks for your help and inspiration. For me it was a good lesson to hunt "the bug" through the sources of Ultimate++!

Gabi

Well, I still do not want to deny possible responsibility....

I guess there are two areas that could produce this problem. One of them is the area you have already tested, but the code there is quite long and never caused anything like this.

Another possible cause is the fact that the wrong bitmap is a result of CH operation that uses caching via ImageMaker (ChImageMaker). Something might be wrong with that...

Subject: Re: Appearance problem in TabCtrl
Posted by qapko on Thu, 09 Jun 2011 09:45:59 GMT
View Forum Message <> Reply to Message

In message #32575 (posted on 25th May 2011) I've written about the results of dumping the distorted image to the file. My opinion is that the cache itself is OK. I'm able to reproduce the dumping test even with the testcase I've posted here recently.

Gabi

Subject: Re: Appearance problem in TabCtrl
Posted by qapko on Mon, 13 Jun 2011 07:52:46 GMT
View Forum Message <> Reply to Message

Hello,
after some private messaging with Sender Ghost I decided to apply one useful hint from him and added Ctrl::GlobalBackPaintHint() to the GUI_APP_MAIN of the application. It works fine now.

Gabi

Subject: Re: Appearance problem in TabCtrl
Posted by mirek on Mon, 13 Jun 2011 08:23:12 GMT
View Forum Message <> Reply to Message

qapko wrote on Mon, 13 June 2011 03:52Hello,
after some private messaging with Sender Ghost I decided to apply one useful hint from him and added Ctrl::GlobalBackPaintHint() to the GUI_APP_MAIN of the application. It works fine now.

Gabi

Interesting, that would mean the problem is rather in reresh/paint algorithm than in the Image...
