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?
