

Hi,

I recall a thread wondering about the fuzzyness of TheIDE (and others) when scaling the display to. e.g. 150% on Windows Vista or 7. Now I found an article <http://msdn.microsoft.com/en-us/library/dd464660.aspx> about Writing High-DPI Win32 Applications.

I added `::SetProcessDPIAware();` call in `<CtrlCore/Win32GuiA.h>` in `GUI_APP_MAIN` starting at line 36 as shown below, and got rid of the fuzzyness. Plus allCtrls I used in my application scaled nicely to 150 %. It is important to note that this call should not be done in a DLL, but instead only in the main application process to avoid race condition. Another way could be adding a manifest, but it seemed like a lot of work to me.

Additionally, this call must be done before reading dpi from Windows, since otherwise Windows reports constantly 96 dpi, and prevents the proper scaling of Ctrls, although fuzzyness may still disappear.

```
#define GUI_APP_MAIN \
\
void GuiMainFn_();\
\
int APIENTRY WinMain(HINSTANCE hInstance, HINSTANCE, LPSTR lpCmdLine, int
nCmdShow) \
{ \
    ::SetProcessDPIAware();\
    UPP::Ctrl::InitWin32(hInstance); \
    UPP::coreCmdLine__() = UPP::SplitCmdLine__(UPP::FromSystemCharset(lpCmdLine)); \
    UPP::AppInitEnvironment__(); \
    GuiMainFn_(); \
    UPP::Ctrl::CloseTopCtrls(); \
    UPP::UsrLog("----- About to delete this log..."); \
    UPP::DeleteUsrLog(); \
    UPP::Ctrl::ExitWin32(); \
    UPP::AppExit__(); \
    return UPP::GetExitCode(); \
} \
\
void GuiMainFn_()
```

I do not know, if this breaks anything in your applications, but it seems to me UPP handles this nicely.

Could this be merged?

Best regards,

Tom

---

Subject: Re: Writing High-DPI Win32 Applications  
Posted by [Tom1](#) on Wed, 14 Dec 2011 10:51:43 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

For some reason, TheIDE (ide/idewin.cpp) needs the following code instead of the previous solution to work:

```
static bool dpi_awareness_initializer=::SetProcessDPIAware();

#ifdef flagMAIN
GUI_APP_MAIN
#else
void AppMain____()
#endif
```

So I guess the universally proper place to put ::SetProcessDPIAware(); is not quite clear to me -- yet...

It needs to be before getting DPI in UPP but not called when we are building dlls.

Best regards,

Tom

---

Subject: Re: Writing High-DPI Win32 Applications  
Posted by [mirek](#) on Wed, 14 Dec 2011 13:24:30 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Tom1 wrote on Wed, 14 December 2011 05:51For some reason, TheIDE (ide/idewin.cpp) needs the following code instead of the previous solution to work:

```
static bool dpi_awareness_initializer=::SetProcessDPIAware();

#ifdef flagMAIN
```

```
GUI_APP_MAIN
#else
void AppMain____()
#endif
```

So I guess the universally proper place to put `::SetProcessDPIAware();` is not quite clear to me -- yet...

It needs to be before getting DPI in UPP but not called when we are building dlls.

Best regards,

Tom

Good find, thank you, adding to RM to investigate...

---

Subject: Re: Writing High-DPI Win32 Applications  
Posted by [Tom1](#) on Wed, 14 Dec 2011 13:47:52 GMT  
[View Forum Message](#) <> [Reply to Message](#)

Hi,

There is yet another related issue of small and large icon sizes covered in <http://msdn.microsoft.com/en-us/library/ms701681%28v=VS.85%29.aspx> .

So, if anybody ever wonders why icons do not scale well in upp applications, there's the answer. The `TopWindow::Icon()` should be called with properly sized icons depending on DPI to prevent icon raster rescaling.

Best regards,

Tom

---

Subject: Re: Writing High-DPI Win32 Applications  
Posted by [Tom1](#) on Fri, 16 Dec 2011 08:31:16 GMT  
[View Forum Message](#) <> [Reply to Message](#)

Further findings: `SetProcessDPIAware();` cannot be called when running in Windows XP, so this will require Vista or higher...

Will come back with further information.

// Tom

---

---

Subject: Re: Writing High-DPI Win32 Applications  
Posted by [Tom1](#) on Fri, 16 Dec 2011 09:38:11 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

OK, now I have tried to embed a manifest and it seems to work on both Windows XP x32 and Windows 7 x64:

Here's how: First I created a file in the main package directory called manifest.xml:

```
<assembly xmlns="urn:schemas-microsoft-com:asm.v1" manifestVersion="1.0"
xmlns:asmv3="urn:schemas-microsoft-com:asm.v3" >
  <asmv3:application>
    <asmv3:windowsSettings xmlns="http://schemas.microsoft.com/SMI/2005/WindowsSettings">
      <dpiAware>true</dpiAware>
    </asmv3:windowsSettings>
  </asmv3:application>
</assembly>
```

Then added a Custom build steps :: Commands:

```
mt.exe -manifest manifest.xml -outputresource:$(OUTPATH);1
```

// Tom

[EDIT]

Testing this solution was successful on Windows 2000 Professional, Windows XP Professional x32, Windows Vista Business x64 and Windows 7 Professional x64.

---

---

Subject: Re: Writing High-DPI Win32 Applications  
Posted by [mirek](#) on Wed, 22 May 2013 05:56:31 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

A little bit late, but... SetProcessDPIAware now used.

As for icon sizes, ide icon designer is already able to export any number of icon sizes into .ico file, so this should be already covered (you just have to provide more icons...).

Mirek

---

---

Subject: Re: Writing High-DPI Win32 Applications  
Posted by [Tom1](#) on Thu, 23 May 2013 07:42:23 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Better late than never! I'll take a look at it some time later; Now my hands are full of work.

Thanks,

Tom

---