
Subject: Re: Native DPI

Posted by [mirek](#) on Fri, 29 May 2015 08:16:25 GMT

[View Forum Message](#) <> [Reply to Message](#)

Tom1 wrote on Thu, 28 May 2015 09:47Hi,

This is what I tried.

Added file manifest.xml to the ide package to enable per-monitor DPI awareness:

```
<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/PM</dpiAware>
    </asmv3:windowsSettings>
  </asmv3:application>
</assembly>
```

This also needs a Custom build step to be added for WIN32 - post-link to be effective:

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

Then added two Size variables to TopWindow class in TopWindow.h:

```
Size currentdpi;
Size primarydpi;
```

Finally added two new cases to TopWindow::WindowProc in TopWin32.cpp:

```
case WM_CREATE:{
  HDC hdc=GetDC(hwnd);
  primarydpi=Size(GetDeviceCaps(hdc,LOGPIXELSX),GetDeviceCaps(hdc,LOGPIXELSY)); //
Returns primary display DPI
  ReleaseDC(hwnd,hdc);
  // RLOG(Format("WM_CREATE: primary DPI(%d,%d)",primarydpi.cx,primarydpi.cy));
  break;
}
case 0x02E0: // WM_DPICHANGED -- e.g. moved window to another monitor with different DPI
  // WndSetPos(Rect(*(RECT*)lParam)); // LPARAM: RECT* of suggested new window pos --
non-optimal
```

```

currentdpi=Size(LOWORD(wParam),HIWORD(wParam)); // Get DPI setting for current display
monitor
// RLOG(Format("WM_DPICHANGED: current DPI(%d,%d)",currentdpi.cx,currentdpi.cy));
if(primarydpi.cy&&currentdpi.cy){
    String face;
    int height;
    GetStdFontSys(face,height);
    SetStdFont(GetStdFont().Height(height*currentdpi.cy/primarydpi.cy));
    GUI_HiDPI_Write(currentdpi.cx>150?1:0); // This does not work optimally here.
}
break;

```

Possible DPI values returned by Windows depending on scaling set in Control Panel / Display:

```

96 DPI = 100% scaling
120 DPI = 125% scaling
144 DPI = 150% scaling
192 DPI = 200% scaling

```

... but there may be even more available. I think I saw 250% somewhere already.(?)

The result is that Font size scales properly when moving the window between the two monitors with different DPI -- 192 and 96 in my case.

There are problems though:

- Line spacing in lists does not adjust according to the StdFontSize while the actual fonts in the lists do.
- Check boxes, radio buttons, scroll bars and spin control arrows in dialogs remain in the HI-DPI mode regardless the change in the mode.
- Manually set font sizes in TheIDE do not scale according to the changes in StdFontSize. In fact, it would be very useful to just declare a scaling percentage e.g. 25... 400 in comparison to the StdFontSize for these fonts instead of their actual sizes. This would keep them proportionally sized even in single monitor scenarios where display scaling is changed in Windows Control Panel.

Further on I think all controls should be scaled and drawn based directly on a per TopWindow defined StdFontSize. This becomes evident when opening a new dialog on a different monitor or moving it onto another monitor. Then scaling really gets out of control.

Best regards,

Tom

I am afraid that per-window scaling is a massive change. We will do it eventually, but for the next release, I would concentrate on 'Vista style' DPI awareness (means primary UHD display scaled by us, secondary non-UHD display scaled down by windows).

(Note that HiDPI is not the primary focus of next release; new Assist++ parser, better support for C++11 are main topics).

Mirek
