
Subject: Re: What is the minimum OpenGL version required for GLCtrl?

Posted by [Tom1](#) on Fri, 30 Aug 2019 10:46:57 GMT

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

Hi Mirek,

I think I have solved the issue. Now MakeWGLContext() cleanly skips the second pass employing wglChoosePixelFormatARB (simultaneously disabling MSAA and possibly also double buffering) if an OpenGL version below 2.1 is detected. Not sure though, if this version limit is correct/optimal, but at least I have successfully used wglChoosePixelFormatARB with OpenGL version 2.1.2.

Here's the revised code:

```
void MakeWGLContext(int depthBits, int stencilBits, int samples)
{
    ONCELOCK {
        for(int pass = 0; pass < 2; pass++) {
            HWND hWnd = CreateWindow("UPP-CLASS-A", "Fake Window",
                                     WS_CAPTION|WS_SYSMENU|WS_CLIPSIBLINGS|WS_CLIPCHILDREN,
                                     0, 0, 1, 1, NULL, NULL,
                                     NULL, NULL);

            if(!hWnd)
                return;
            HDC hDC = ::GetDC(hWnd);
            if(!hDC)
                return;
            memset(&s_pfd, 0, sizeof(s_pfd));
            if(pass == 0) {
                s_pfd.nSize = sizeof(s_pfd);
                s_pfd.nVersion = 1;
                s_pfd.dwFlags =
PFD_DRAW_TO_WINDOW|PFD_SUPPORT_OPENGL|PFD_GENERIC_ACCELERATED|PFD_
GENERIC_FORMAT|PFD_DOUBLEBUFFER_DONTCARE;
                s_pfd.iPixelFormat = PFD_TYPE_RGBA;
                s_pfd.cColorBits = 32;
                s_pfd.cAlphaBits = 8;
                s_pfd.cDepthBits = 24;
                s_pfd.cStencilBits = 8;
                s_pfd.iLayerType = PFD_MAIN_PLANE;
                s_pixelFormatID = ChoosePixelFormat(hDC, &s_pfd);
            }
            else {
                Vector<int> attr;
                attr
                << WGL_DRAW_TO_WINDOW_ARB << GL_TRUE
                << WGL_SUPPORT_OPENGL_ARB << GL_TRUE
                << WGL_DOUBLE_BUFFER_ARB << GL_TRUE
                << WGL_PIXEL_TYPE_ARB << WGL_TYPE_RGBA_ARB
```

```

<< WGL_ACCELERATION_ARB << WGL_FULL_ACCELERATION_ARB
<< WGL_COLOR_BITS_ARB << 32
<< WGL_ALPHA_BITS_ARB << 8
<< WGL_DEPTH_BITS_ARB << depthBits
<< WGL_STENCIL_BITS_ARB << stencilBits
;
if(samples > 1)
    attr
        << WGL_SAMPLE_BUFFERS_ARB << GL_TRUE
        << WGL_SAMPLES_ARB << samples
    ;
    attr << 0;
    UINT numFormats;
    if(!wglChoosePixelFormatARB(hDC, attr, NULL, 1, &s_pixelFormatID, &numFormats))
        return;
}

```

```

DescribePixelFormat(hDC, s_pixelFormatID, sizeof(PIXELFORMATDESCRIPTOR), &s_pfd);
if(!SetPixelFormat(hDC, s_pixelFormatID, &s_pfd))
    return;

```

```

s_openGLContext = wglCreateContext(hDC);

```

```

bool enhanced_mode=false;

```

```

if(pass == 0) {
    HGLRC hRC = wglCreateContext(hDC);
    wglMakeCurrent(hDC, s_openGLContext);
    glewInit();

    if (glewIsSupported("GL_VERSION_2_1")) enhanced_mode=true;

    wglMakeCurrent(NULL, NULL);
}

```

```

    ReleaseDC(hWND, hDC);
    DestroyWindow(hWND);

```

```

if(!enhanced_mode) break; // In basic mode, this is it.
}
}
}

```

Reference/OpenGL now runs OK with Windows software OpenGL version 1.1.0 and also on HW accelerated OpenGL version 4.5. On 1.1.0 MSAA and DoubleBuffering are silently ignored even when they are enabled in GLCtrl.

Can you check if this is OK from your point of view and possibly commit?

Best regards,

Tom
