Re: Improving Application Performance by DISABLING hardware acceleration?

Source:
http://www.tech−archive.net/Archive/Development/microsoft.public.win32.programmer.gdi/2005−08/msg00150.html

- *From*: "gregory_may" <None>
- *Date*: Thu, 11 Aug 2005 12:46:27 −0700

Thanks Severian for responding to this post.

As a background note, I have been using a Nvidia Go5200 and ATI 9600 as my main test machines.

I found a similar issue when using Windows Media Encoder for screen capture (Disabling Hardware Acceleration improving performance):
http://www.microsoft.com/windows/windowsmedia/9series/encoder/faq.aspx
Look over Section 3.2 −What can I do to improve the quality of my screen captures?

Another issue that Disabling Hardware Acceleration seems to solve, is DirectX overlays become accessable to my code below for a screen capture.

Currently by Disabling Hardware acceleration, I gain about 40−50MS of speed per screen capture!

>From the look of your post, DIBs are worth investigating! It sounds like there is a much better way to do this code, than what I have. I might need a few pointers on where to start on using DIB's.

I will post the code I am using below:
−−−−−−−−−−−−−−−−−−−−−−−−−

```
Public oBackground As System.Drawing.Bitmap

Public Sub CaptureScreen()

Dim hSDC, hMDC, hMDC2 As Integer

Dim hBMP, hBMPOld As Integer

Dim CurInf As CursorInfo

Dim CurPos As PointAPI

Dim CursorIcon As New ICONINFO
```

```
Dim FW As Integer 'Width

Dim FH As Integer 'Height

Try

'Source DC

hSDC = CreateDC("DISPLAY", "", "", "")

'Destination DC with transpairent windows

hMDC = CreateCompatibleDC(hSDC)

'Screen Width

FW = GetDeviceCaps(hSDC, HORIZRES)

'Screen height

FH = GetDeviceCaps(hSDC, VERTRES)

hBMP = CreateCompatibleBitmap(hSDC, FW, FH)

hBMPOld = SelectObject(hMDC, hBMP)

Call BitBlt(hMDC, 0, 0, FW, FH, hSDC, 0, 0, RasterOperations.SRCCOPY +
RasterOperations.CAPTUREBLT)

Call DeleteDC(hSDC)

'get cursor information

CurInf.cbSize = Len(CurInf)

Call GetCursorInfo(CurInf)


If (CurInf.flags And CURSOR_SHOWING) Then

'Get Hot Spot Info

GetIconInfo(CurInf.hCursor, CursorIcon)

Call DrawIconEx(hMDC, CurInf.ptScreenPos.X − CursorIcon.xHotspot, _

CurInf.ptScreenPos.Y − CursorIcon.yHotspot, _

CurInf.hCursor, 0, 0, 0, False, DI_NORMAL)
```

End If

Call SelectObject(hMDC, hBMPOld)

Call DeleteDC(hMDC)

oBackground = System.Drawing.Image.FromHbitmap(New IntPtr(hBMP))

DeleteObject(hBMP)

DeleteObject(hBMPOld)

Catch ex As Exception

Debug.WriteLine("CaptureScreen: General GDI ERROR: " & ex.Message.ToString)

End Try

End Sub

"Severian [MVP]" <severian@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx> wrote in message
news:ekjjf15om69ici5125n0jkth6d40ovi5lk@xxxxxxxxxx
> On Tue, 9 Aug 2005 11:36:35 −0700, "gregory_may" <None> wrote:
>
>>My understanding, is there are two paths to the graphics hardware. One is
>>Legacy GDI and the other is DirectX:
>>http://msdn.microsoft.com/library/default.asp?url=/library/en−us/directx9_c/directx/graphics/programmingguide/ge
>>
>>It seems that disabling graphics hardware acceleration can dramatically
>>improve performance of GDI applications and system resource usage on
>>Windows
>>2000/XP.
>
> I would think this would only be the case with broken, or improperly
> optimized, display adapters/drivers.
>
> A specific difference I recall is that disabling hardware acceleration
> may force DDBs to be allocated in system (rather than video) memory;
> if you have a fast processor but a crappy display adapter, this may
> improve performance when operating on in−memory DDBs.
>
>>Anyone know of a programmatic way to disable hardware acceleration?
>
> Modern display adapters work differently than old ones. Using GDI, old
> ones would usually perform better with DDBs rather than DIBs, while
> the reverse seems to sometimes be the case today.
>
> If you're using memory DCs with DDBs (compatible bitmaps), you may

> want to try switching to DIBs; these are not as memory−limited as DDBs
> and may solve your problem without having to reconfigure the display.
>
> Otherwise, let us know what display device you're using now, and what
> kind of drawing operations you're doing (and whether you're doing them
> directly to the screen or using a memory DC).
>
> −−
> Phillip Crews aka Severian
> Microsoft MVP, Windows SDK
> Posting email address is real, but please post replies on the newsgroup.

.

- ***Follow−Ups***:
  - ♦ ***Re: Improving Application Performance by DISABLING hardware acceleration?***
    - ◊ *From:* Severian [MVP]

- ***References***:
  - ♦ ***Improving Application Performance by DISABLING hardware acceleration?***
    - ◊ *From:* gregory_may
  - ♦ ***Re: Improving Application Performance by DISABLING hardware acceleration?***
    - ◊ *From:* Severian [MVP]

- Prev by Date: ***Re: Reading a DIB's palette***
- Next by Date: ***Re: Getting AddFontMemResourceEx to load multiple fonts***
- Previous by thread: ***Re: Improving Application Performance by DISABLING hardware acceleration?***
- Next by thread: ***Re: Improving Application Performance by DISABLING hardware acceleration?***
- Index(es):
  - ♦ ***Date***
  - ♦ ***Thread***