

# Re: Dual core and the Sample Framework

---

*Source:*

<http://www.tech-archive.net/Archive/Development/microsoft.public.win32.programmer.directx.managed/2006-02/ms>

---

- *From:* "ZMan" <[zman@xxxxxxxxxxxxxxxx](mailto:zman@xxxxxxxxxxxxxxxx)>
  - *Date:* Tue, 28 Feb 2006 11:46:31 -0800
- 

This may help

[http://msdn.microsoft.com/library/default.asp?url=/library/en-us/directx9\\_c/Game\\_Timing\\_and\\_Multicore\\_Processors](http://msdn.microsoft.com/library/default.asp?url=/library/en-us/directx9_c/Game_Timing_and_Multicore_Processors)

However it sounds like you are talking more about issues with the sample framework – as far as I know (and I just searched the sample framework for 'thread') the sample code is not actually multithreaded – it certainly doesn't turn on the multithreaded flag for D3D. So what you are seeing is (I assume) your application being switched across the processors by windows as part of its context switching (I had no clue that it could do that). From the above article though setting processor affinity is the correct way to fix this.

—

Zman

<http://www.thezbuffer.com> – News and information for Managed DirectX

"Peter C." <[PeterC@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx](mailto:PeterC@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx)> wrote in message <news:D07E909C-556F-4C65-8478-B4632400D675@xxxxxxxxxxxxxxxxxxxx>

If this is an old topic or if it's already solved, I'm quite sorry.

I'm using the February 2006 SDK to make an MDX2.0 app. On my dual core machine, the timing from the Sample Framework (totalTime, elapsedTime) gets screwed up. I found the cause, which is that the render code is run in threads which run on both cores. Since the cores have slightly different clock counters and the time is based on them, the current time jumps backward and forward small amounts and makes the animation terribly jumpy. Setting the current Process's affinity to either core solves the problem.

Is there a way around this issue without limiting the entire app to one core? I've thought about workarounds but I haven't come up with anything very bright yet.

