

# Re: managed pool question

---

*Source:*

<http://www.tech-archive.net/Archive/Development/microsoft.public.win32.programmer.directx.graphics/2005-08/ms>

---

- *From:* "Wessam Bahnassi" <[wbahnassi@xx](mailto:wbahnassi@xx)>
  - *Date:* Fri, 19 Aug 2005 09:54:39 +0300
- 

You might get out-of-memory errors (based on HW video memory and AGP size).  
Or in the best case, you'll face some stutters when requesting managed but unloaded resources.

Wessam Bahnassi  
Microsoft DirectX MVP,  
Lead Programmer  
In|Framez

"john" <[jdlee12@xxxxxxxx](mailto:jdlee12@xxxxxxxx)> wrote in message  
[news:Oxw%23QoFpFHA.1044@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx](mailto:news:Oxw%23QoFpFHA.1044@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx)  
> I want to load very huge amount of textures and vertices  
> into memory using CreateTextureFromFileEx and CreateVertexBuffer .  
>  
> I will use managed mode setting .  
> and I will load it all at once in loading time.  
>  
> the amount of resources that will be loaded all at once is  
>  
> vertices = about 100 MBytes.  
> textures = dds format dxt1 or dxt3 about 100 MBytes.  
>  
> I will load total 300 MBytes in managed memory pool.  
>  
> and I will render objects [most of them are object meshes]  
> only visible things using frustum culling.  
>  
> so , the amount of drawing resources at each frame is not too large.  
>  
>  
> my question is  
> if I do it like above, what problem will be happened ?  
>  
> no problem happen ?  
>  
> or some serious problem that I don't know happen ?  
>  
> % my target PC spec is 256 M RAM, 1.2 GHz CPU, Geforce 4 MX 400 AGP 8x 64

Re: managed pool question

- > M
- > Vid Mem
- > hard disk storage 40Gb.
- >
- >
- >
- >
- >
- >
- >

.

---

• **References:**

- ◆ **managed pool question**
  - ◇ *From: john*
- Prev by Date: **backface culling transparent objects**
- Next by Date: **Re: backface culling transparent objects**
- Previous by thread: **managed pool question**
- Next by thread: **Re: managed pool question**
- Index(es):
  - ◆ **Date**
  - ◆ **Thread**