Release Notes

SUPPLEMENT

DRIVER VERSION: 15.36.21.4222 & 15.36.21.64.4222

DATE: June 2, 2015

Additional Information on Dedicated Memory Reporting

The graphics driver now reports some amount of fictitious "dedicated" graphics memory to the operating system (128MB by default) to work around applications issues in a number of games. These games incorrectly look for some amount of "dedicated" graphics memory because they weren't coded with the unified memory architecture for processor graphics.

The amount reported can be modified or disabled if desired by changing the following registry value:

HKEY_LOCAL_MACHINE\Software\Intel\GMM, create a REG_DWORD value named "DedicatedSegmentSize". The value is interpreted as number of MB to report (0-512MB). A value of 0 disables reporting the dummy memory segment. If the registry key/value is not found, the driver will report the default 128MB.

Background: Intel processor graphics reserves a small amount of memory at BIOS boot for internal bookkeeping purposes. Any preallocated memory can only be used by graphics and not by the operating system or applications such as productivity tools, web browsers, etc. There is no performance benefit even for graphics with such preallocation since it is the same DRAM memory either way. As a result, the driver normally allocates memory for graphics resources via call to the operating system to dynamically obtain the pages of memory required. The graphics driver can dynamically allocate up slightly less than 50% of the physical memory on the system. Applications which want to check for available graphics memory should ask for how much "shared" or "total" graphics memory is available via one of the Microsoft specified APIs.

To offer users the most flexibility and user experience, especially on systems with small amounts of memory, Intel has been steadily reducing the pre-allocated memory over time, dropping from 128MB or more in 2010 to 64MB to down to 32MB in 2014. The driver only uses a portion of the pre-allocated memory, reporting the rest to the operating system as a memory segment that the operating system could then choose to place some graphics resources in. The OS reported this left over pre-allocated memory as "dedicated" graphics memory the Windows APIs listed above. CPU access to update data in that segment was slow

(especially when paging data in/out for standby) due to the need to use PCI Aperture access mechanism. As Intel reduced the preallocated size, the amount of residual left over got too small to be worth offering to the operating system. Consequently, when Intel reduced the pre-allocated to 32MB, a decision was made to no longer report any residual "dedicated" memory.

Unfortunately, there are a number of applications which incorrectly use the APIs listed above and read only the amount of "dedicated" memory and then make decisions based on that data – for example, refusing to run, limiting the resolution/texture detail, crashing, etc. For discrete graphics cards, the dedicated memory on the card is "fast" and the PCIE bus to access "shared" memory is slow. As such, application developers have been conditioned to think that the performance of their application will be determined by whether the game assets for a frame can fit in the dedicated memory. This logic is incorrect on processor graphics and unified memory architecture. Intel has tried to work with application developers to correct the misunderstanding and to get applications to apply appropriate detection logic. Unfortunately, not all application developers are responsive and many are uninterested in patching their games to make the detection logic work properly on Intel graphics.

Intel debated boosting the preallocated memory back up as a workaround for these application issues but ultimately decided that wasn't the right solution since it would potentially waste that memory in cases where the user needs the memory for operating system or applications instead of for graphics.

Instead, intel has implemented a workaround where the driver reports a dummy "dedicated" memory segment to the operating system which doesn't actually consume any physical memory (no backing storage) and instructs the operating system to never place any resources in that segment. This preserves all system memory for use by either graphics or applications while working around the application's incorrect graphics memory detection logic.

The list of applications which have been reported or observed to have incorrect detection logic which may run better with this workaround include:

- Grand Theft Auto IV
- Pro Evolution Soccer (various versions)
- Older games from the Total War series (e.g. Empire: Total War and earlier)
- Star Wars: The Old Republic
- Roblox
- Disney: Infinity
- Evolve (may be patched in latest version)