Release Notes for the NVIDIA® OptiX™ ray tracing engine

Version 3.7 February 2015

Welcome to the latest release of the NVIDIA OptiX ray tracing engine and SDK, with support for all CUDA-capable GPUs. This package contains the libraries required to experience the latest technology for programmable GPU ray tracing, plus pre-compiled samples (with source code) demonstrating a broad range of ray tracing techniques and highlighting basic OptiX functionality.

End User License Agreement (EULA) Reminder

As of version 3.5, OptiX continues to be free to acquire and use within your own company or institution, and to include within applications you distribute but do no charge for, but commercial applications now require a Commercial License to redistribute OptiX 3.5 and later. NVIDIA is making this change to give commercial applications the higher level of support they require. The *OptiX Commercial* version also includes additional functionality and examples that are relevant to commercial products. Please email: *optix-commercial@nvidia.com* if you are interested in becoming a Commercial OptiX Developer and we will supply you with all the details. In the meantime, any development done with this OptiX version will be compatible with OptiX Commercial.

Support:

The normal path for OptiX support is on NVIDIA's Developer Zone at: https://devtalk.nvidia.com/default/board/90/

If you have any confidential concerns please send your issues directly to OptiX-Help@NVIDIA.com and we will do our best to address them. Emailed questions found not to be confidential will often be redirected to the forum so the community can benefit from the answers.

System Requirements (for running binaries referencing OptiX)

Graphics Hardware:

- CUDA capable devices of Compute Capability 2.0 ("Fermi") or higher are supported on **GeForce**, **Quadro**, or **Tesla** class products. Out-of-core ray tracing of large datasets exceeding GPU memory (AKA paging) is not supported on GeForce "GT" class GPUs.

Graphics Driver:

- The CUDA R343 or later driver is **required**. The latest available WHQL drivers are highly recommended. For the Mac, the driver extension module supplied with CUDA 5.0 or later will need to be installed. Note that the Linux and Mac drivers can only be obtained from the CUDA 6.5 download page at the moment.
- SLI is not required for OptiX to use multiple GPUs, and it interferes when OptiX uses either D3D or OpenGL interop. Disabling SLI will not degrade OptiX performance and will provide a more stable environment for OptiX applications to run. SLI is termed "Multi-GPU mode" in recent NVIDIA Control Panels, with the correct option being "Disable multi-GPU mode" to ensure OptiX is not encumbered by graphics overhead.

Operating System:

- Windows 7/8/8.1 32-bit or 64-bit; Linux RHEL 4.8 or Ubuntu 10.10 - 64-bit; OSX 10.8+.

Development Environment Requirements (for compiling with OptiX)

All Platforms (Windows, Linux, Mac OSX):

- **CUDA Toolkit 4.0, 4.1, 4.2, 5.0, 5.5, 6.0, 6.5.**
  OptiX 3.7 has been built with CUDA 6.5, but any specified toolkit should work when compiling PTX for OptiX. If an application links against both the OptiX library and the CUDA runtime on Mac and Linux, it is recommended to use the same version of CUDA as OptiX was built against.
- **C/C++ Compiler**
  Visual Studio 2008, 2010, 2012, or 2013 is required on Windows systems. gcc 4.4-4.8 have been tested on Linux. The 5.1 versions of Xcode development tools have been tested on Mac OSX 10.8.
- **GLUT**
  Most OptiX samples use the GLUT toolkit. Freeglut ships with the Windows OptiX distribution. GLUT is installed by default on Mac OSX. A GLUT installation is required to build samples on Linux.

Enhancements in OptiX 3.7 final

- Change several samples to default to Trbvh builder.
- Improve concurrency in OptiX Prime with multiple GPUs.
- Improve error messages for invalid children of Selector nodes.
- Bug fix with bindless texture IDs getting deleted and reused.
- Bug fix in samples. Some textures were not being loaded.

Enhancements in OptiX 3.7 beta 3

- Bug fix in both OptiX Prime and OptiX with Trbvh. The compiler in the R346 driver has a regression that caused kernel timeouts.
- Bug fix for OAC trace capture in OptiX Prime.
- Bug fix for st.local.v* instructions targeting a local pointer.
- Bug fix with Trbvh in locales that use “,” as the decimal point.
- Bug fix in OptiX Prime with chunked acceleration structure build on rtpModelCopy.
- Bug fix for missing inline decorator in optix_primepp.h.

Enhancements in OptiX 3.7 beta 2

- Support PTX copysign instruction.
- Don’t ship 32-bit OptiX Prime libraries, which was never a supported configuration.
- Additional enhancements in OpenGL library and glew linking.
- Bug fix for PROGRAM_ID exception check.
- Bug fix for OAC trace capture.
- Bug fix while parsing input programs for Fermi.
- Bug fix with callable programs in Win32 build.

- Bug fix with chunked Trbvh builds in OptiX.

Enhancements in OptiX 3.7 beta 1

- **Instancing** support in OptiX Prime. The implementation is very fast. The typical performance hit vs. non-instanced geometry is only 5%. See a demo in this post: http://blog.mentalray.com/.
- **Tesla K80 support** with SM 3.7. This is the highest performing GPU for ray tracing.
- **Visual Studio 2013 support.**
- **CUDA 6.5 support.**
- **OpenGL** libraries are now dynamically loaded, allowing easier linking on Linux, especially for HPC installations.
- **Samples** have updated model loading code.
- **OptiX API Capture** bug fixes for OptiX Prime.
- **rtpQuerySetCudaStream** API added to OptiX Prime to for increased performance on asynchronous queries.
- **RTP_BUFFER_FORMAT_INDICES_INT3_MASK_INT** is an additional way to suppress intersections without rebuilding the Model's acceleration structure.
- **Bug fix** for when RT_EXCEPTION_PROGRAM_ID_INVALID is used with callable programs.
- **Bug fixes** to Trbvh.
- **Bug fix** Only dirty device-side storage on rtBufferGetDevicePtr.
- **Bug fix** Prevent Collision sample window resizing when using the fake rendered background, since it doesn't scale.
- **Bug fix** Add more detail to the "Unable to set the CUDA device" error message.
- **Documentation improvements** in OptiX Prime API reference and Buffer of Buffer IDs in Programming Guide, and RTtimeoutcallback in OptiX API reference.

Limitations in OptiX 3.7

- The 32-bit Windows distribution has been removed, but the 32-bit Windows libraries are found in the 64-bit Windows distribution.
- The ability to use 32-bit PTX within a 64-bit application has been **removed**, but 32-bit PTX with 32-bit applications still works on Windows.
- **32-bit OSX** builds have been removed.
- **ALL 32-bit support will be removed in an upcoming version.**
- **SM 1.X GPU support has been removed**. This includes G80 through GT2XX-based GPUs.
- **DirectX 9 Interop support will be removed** in upcoming drivers. DirectX 9 Ex will continue to be supported. We recommend porting the application to DirectX 11.
- OptiX supports running with NVIDIA Parallel Nsight but does not currently support kernel debugging in Nsight. In addition, it is not recommended to compile PTX code using any -G (debug) flags to nvcc.

- All GPUs used by OptiX must be of the same MAJOR compute capability, such as compute capability 3.x or 5.x, and may only differ in minor compute capability by 0.2. OptiX will automatically select the set of GPUs of the highest major compute capability and only use those. For example, in a system with a GeForce GTX 460 (compute 2.1) and a GeForce GTX 480 (compute 2.0), both will be used, but in a system with a Quadro 6000 (compute 2.0) and a Quadro M6000 (compute 5.2) only the compute 5.2 device would be selected. Applications may explicitly choose which GPUs to run, as is done in the progressive photon mapper sample, ppm.cpp, at the start of initScene(), but if the application requests a set of devices of different major compute capability an error will be returned.
- malloc() and free() do not work in device code.
- Linux only: due to a bug in GLUT on many Linux distributions, the SDK samples will not restore the original window size correctly after returning from full-screen mode. A newer version of freeglut may avoid this limitation.
- Several OptiX SDK samples are intended to run on a single GPU and will perform more slowly when run on multiple GPUs. The samples: Cook, Hybrid Shadows, isgReflections, isgShadows, and SimpleAnimation will only run at full speed on a single GPU. The compiled versions of these samples can be constrained to a single GPU by setting the environment variable CUDA_VISIBLE_DEVICES = N, where "N" is the alignumber of the GPU you wish to utilize. The OptiX sample3 is a convenient means to quickly identify your device ID's.

Performance Notes:

- See the Performance Guidelines chapter of the OptiX Programming Guide.
- OptiX performance tracks very closely to a GPU's CUDA core count and core clock speed for a given GPU generation. The performance "value" of a CUDA core varies greatly between GPU generations (e.g., Fermi, Kepler, Maxwell) but can be compared directly within the same GPU generation.
- Different techniques may find varying performance results on different GPU generations. For example, path tracing benefited more from the Fermi architecture than did simpler techniques as compared to how they performed on GT200.
- OptiX takes advantage of multiple GPUs without using SLI. Multi-GPU scalability will vary with the workload being done, with longer and complex rendering (e.g., path tracing) scaling quite well while simple and fast rendering (e.g. Whitted) may scale far less.
- Performance of OpenGL and DirectX interop is unpredictable when SLI is enabled. As noted previously, SLI is not required to achieve scaling across multiple GPUs and is not recommended when using OptiX. Mixing board types will reduce the memory size available to OptiX to that of the smallest GPU.
- Performance will be better when the entire scene fits within a single GPU’s memory. Adding additional GPUs increases performance, but does not increase the available memory beyond that of the smallest board. The entire scene must fit within GPU memory when OptiX Paging is disabled or not available.
- For compute-intensive rendering, performance is usually fairly linear to the number of pixels rendered. Reducing resolution can make development on entry level boards or laptops more practical.

Other Notes:

- **CMake 2.8.12** http://www.cmake.org/cmake/resources/software.html
  The executable installer http://www.cmake.org/files/v2.8/cmake-2.8.12-win32-x86.exe is recommended for Windows systems.