



# Enabling High Performance Graphics Rendering on Optimus Systems

TB-05942-003v02 | July 2012

**Driver Technical Note for Notebooks**



# 03 ENABLING HIGH PERFORMANCE GRAPHICS RENDERING ON OPTIMUS SYSTEMS

A key feature of Optimus configurations is to support rendering applications using NVIDIA High Performance Graphics while displaying on monitors connected to the Integrated Graphics.

This document describes:

- ▶ [Methods That Enable NVIDIA High Performance Graphics Rendering on Optimus Systems](#)
- ▶ [Methods That Expose NVIDIA Graphics Processor Information on Optimus Systems](#)

## Methods That Enable NVIDIA High Performance Graphics Rendering on Optimus Systems

This feature is enabled using the following set of methods, listed in order from highest to lowest precedence:

- 1 [Forced Rendering Mode](#)
- 2 [Right-Click Context Menu](#)
- 3 [Application Profile Settings](#)
- 4 [Static Library Bindings](#)
- 5 [Global Variable NvOptimusEnablement \(new in Driver Release 302\)](#)
- 6 [Global Profile Settings](#)

The rendering device for a given application is determined when the application starts. Changes to any of the settings for a given application, or set of applications will NOT apply to any running instance of an application, and will take effect only upon subsequent application executions. The following sections describe each of the methods.

## Forced Rendering Mode

OEMs are provided mechanisms within the driver to dynamically override standard Optimus rendering behavior, and force graphics rendering for all applications to occur using either Integrated Graphics or High Performance Graphics. Some OEMs exploit this feature through the use of a physical switch, or a hot key. Others have linked this feature to the use of OS power plans. For example, an OEM may choose to make the system render using only Integrated Graphics when on battery mode.

The Forced Rendering Mode mechanism will override all other mechanisms described below. This mechanism is used to specify rendering the application using either the Integrated Graphics or the NVIDIA High Performance Graphics. It overrides the Global settings for all applications, whether or not they have an associated application profile.



**Note:** This mechanism does NOT override any application profile where the **Preferred graphics processor** setting in the NVIDIA Control Panel is set by NVIDIA as read-only (cannot be changed by the user).

## Right-Click Context Menu

The NVIDIA Optimus driver inserts an item in the mouse right-click context menu which allows the user to override application profile settings for a given application. This applies only to that particular application instance. The user can choose to render using either Integrated or High Performance Graphics.

## Application Profile Settings

Application profiles are accessible from within the NVIDIA Control Panel, under the *Manage 3D Settings->Program Settings* tab. A profile associates an application and one or more executables to a set of attributes that can affect the runtime behavior of the application.

One of the attributes for each profile is **Preferred graphics processor**. The value shown for this attribute describes the current value and a default value for a given profile. While the user can alter the current value based on personal preference, the user can also use *Restore Defaults* to return a profile back to its default settings, which restores each setting to the listed default value.

The **Preferred graphics processor** value will be either 1) Integrated, 2) High Performance, or 3) Use global setting. If the current value is set to *Use global setting*, then the default value is also defined within the parentheses. The default value is either 1) Integrated or 2) High Performance. The *Use global setting* value is a 'power' feature which provides a user the ability to override many profiles at a time. Refer to the [Global Profile Settings](#) section for more information.

The value for the **Preferred graphics processor** attribute may be read-only, which is used by NVIDIA to block the setting from being changed when application or driver constraints exist.

NVIDIA delivers its Optimus driver with an initial set of application profiles, for applications that have been tested and verified to run on an Optimus configuration. Users can also create new application profiles for applications which do not already have a profile. When creating a profile, the user can set the current value for **Preferred graphics processor**. The default value of the setting will be *Auto-Select:Integrated*.

## Static Library Bindings

For any application without an existing application profile, there is a set of libraries which, when statically linked to a given application executable, will direct the Optimus driver to render the application using High Performance Graphics. As of Release 302, the current list of libraries are `vcamp110.dll`, `vcamp110d.dll`, `nvapi.dll`, `nvapi64.dll`, `opencl.dll`, `nvcuda.dll`, and `cuda*.*`.

## Global Variable NvOptimusEnablement (new in Driver Release 302)

Starting with the Release 302 drivers, application developers can direct the Optimus driver at runtime to use the High Performance Graphics to render any application—even those applications for which there is no existing application profile. They can do this by exporting a global variable named `NvOptimusEnablement`. The Optimus driver looks for the existence and value of the export. Only the LSB of the DWORD matters at this time. A value of `0x00000001` indicates that rendering should be performed using High Performance Graphics. A value of `0x00000000` indicates that this method should be ignored.

### Example Usage:

```
extern "C" {
    _declspec(dllexport) DWORD NvOptimusEnablement = 0x00000001;
}
```

## Global Profile Settings

Global profile settings are accessible from within the NVIDIA Control Panel, under the *Manage 3D Settings->Global Settings* tab. The Global Settings **Preferred graphics processor** attribute value can be either 1) *Integrated*, 2) *High Performance*, or 3) *Auto-select*. When the **Preferred graphics processor** attribute is set to *Integrated*, any application that has a profile which is set to *Use global setting*, or any application which does not have an associated profile will render using *Integrated Graphics*. Likewise, if set to *High Performance*, applications matching the same criteria will render using *High Performance Graphics*. If the attribute is set to *Auto-select*, then the global profile setting

is ignored, and applications with profiles will render using the default value for the given application, and any other application will render using Integrated Graphics.

## Methods That Expose NVIDIA Graphics Processor Information on Optimus Systems

Often, applications will check the system's graphics configuration at initialization to determine which hardware and corresponding optimal graphics settings to use during application execution. Optimus systems power off the High Performance Graphics processor when not in use.

However, if the application or the Optimus driver are configured to use the NVIDIA High Performance Graphics hardware through one of the methods listed in [Methods That Enable NVIDIA High Performance Graphics Rendering on Optimus Systems](#), then information about the NVIDIA High Performance Graphics hardware and its corresponding capabilities is made available to the application.

The NVIDIA driver monitors certain runtime APIs and returns the appropriate NVIDIA hardware information through calls made by the application to those APIs.

The following are the APIs that applications can use:

- ▶ **DirectX3 through DirectX9** – IDirect3D9::GetAdapterIdentifier
- ▶ **DirectX9 and above** – IDXGIAdapter::GetDesc
- ▶ **OpenGL** – glGetString(GL\_VENDOR) and glGetString(GL\_RENDERER)

## Notice

ALL NVIDIA DESIGN SPECIFICATIONS, REFERENCE BOARDS, FILES, DRAWINGS, DIAGNOSTICS, LISTS, AND OTHER DOCUMENTS (TOGETHER AND SEPARATELY, "MATERIALS") ARE BEING PROVIDED "AS IS." NVIDIA MAKES NO WARRANTIES, EXPRESSED, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE MATERIALS, AND EXPRESSLY DISCLAIMS ALL IMPLIED WARRANTIES OF NONINFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE.

Information furnished is believed to be accurate and reliable. However, NVIDIA Corporation assumes no responsibility for the consequences of use of such information or for any infringement of patents or other rights of third parties that may result from its use. No license is granted by implication of otherwise under any patent rights of NVIDIA Corporation. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all other information previously supplied. NVIDIA Corporation products are not authorized as critical components in life support devices or systems without express written approval of NVIDIA Corporation.

## HDMI

HDMI, the HDMI logo, and High-Definition Multimedia Interface are trademarks or registered trademarks of HDMI Licensing LLC.

## Macrovision Compliance Statement

NVIDIA Products that are Macrovision enabled can only be sold or distributed to buyers with a valid and existing authorization from Macrovision to purchase and incorporate the device into buyer's products.

Macrovision copy protection technology is protected by U.S. patent numbers 5,583,936; 6,516,132; 6,836,549; and 7,050,698 and other intellectual property rights. The use of Macrovision's copy protection technology in the device must be authorized by Macrovision and is intended for home and other limited pay-per-view uses only, unless otherwise authorized in writing by Macrovision. Reverse engineering or disassembly is prohibited.

## OpenCL

OpenCL is a trademark of Apple Inc. used under license to the Khronos Group Inc.

## Trademarks

NVIDIA and the NVIDIA logo are trademarks or registered trademarks of NVIDIA Corporation in the United States and other countries. Other company and product names may be trademarks of the respective companies with which they are associated.

## Copyright

© 2009, 2012 NVIDIA Corporation. All rights reserved.