



NVAPI Open Source Interface for Driver Release 545

Release Notes

Table of Contents

- NVAPI Release Notes..... 3
 - Introduction..... 3
 - NVAPI Runtime..... 3
 - Changes in NVAPI for Driver Release 545..... 4
 - New Functions 4
 - New/Updated Structures..... 4
 - New/Updated Enums..... 5
 - New/Updated Unions 5
 - New Macros 5
 - New Errors..... 6
 - TCC Support 6
 - Deprecated NVAPI Functions 7
 - Deprecated Enums..... 7
 - NVAPIDriverSettings Additions/Removals 7
 - NVAPI Security Information..... 7

NVAPI Release Notes

Introduction

NVAPI is NVIDIA Corporation's core software development kit that allows access to NVIDIA GPUs and drivers on all Windows platforms. NVAPI provides support for categories of operations that range beyond the scope of those found in familiar graphics APIs such as DirectX and OpenGL.

This release contains a version of `nvapi.h`, provided under MIT license, to enable open-source re-implementations of NVAPI for Windows emulation environments.

For those interested in developing applications using the NVAPI Developer SDK on Windows, NVIDIA recommends using the NVAPI Developer SDK available at <https://developer.nvidia.com/nvapi>.

The following files are provided by NVIDIA:

- > `nvapi.h`

- This file contains the interface constants, structure definitions and function prototypes for NVAPI interface.

- > `nvapi_interface.h`

- This file is a reference for mapping the NVAPI identifiers to functions.

- > `NvApiDriverSettings.h`

- > `NvApiDriverSettings.c`

- These files are used to query and change driver settings. For more details, please see:

- http://developer.download.nvidia.com/NVAPI/PG-5116-001_v01_public.pdf

These release notes describe the changes made in the NVAPI Open Source Interface for this release.

NVAPI Runtime

The NVAPI runtime, `nvapi.dll`, provides the following key functions:

> nvapi_QueryInterface():

Maps a 32-bit identifier to a function pointer.

```
void *nvapi_QueryInterface(NvU32 id);
```

The NVAPI application will call nvapi_QueryInterface() to get individual NVAPI function pointers from nvapi.dll.

Please consult nvapi_interface.h for a mapping of identifiers to NVAPI function names.

> NvAPI_Initialize():

```
NvAPI_Status __cdecl NvAPI_Initialize()  
{  
    return NVAPI_OK;  
}
```

> NvAPI_Unload():

```
NvAPI_Status __cdecl NvAPI_Unload()  
{  
    return NVAPI_OK;  
}
```

Changes in NVAPI for Driver Release 545

New Functions

- > Added NvAPI_GPU_NVLINK_GetCaps
- > Added NvAPI_GPU_NVLINK_GetStatus
- > Added NvAPI_GPU_GetGPUInfo
- > Added NvAPI_DISP_GetEdidData
- > Added NvAPI_D3D12_EnableRaytracingValidation
- > Added NvAPI_D3D12_RegisterRaytracingValidationMessageCallback
- > Added NvAPI_D3D12_UnregisterRaytracingValidationMessageCallback
- > Added NvAPI_D3D12_FlushRaytracingValidationMessages

New/Updated Structures

- > Added NVLINK_GET_CAPS_V1
- > Added NVLINK_DEVICE_INFO_V1
- > Added NVLINK_LINK_STATUS_INFO_V1
- > Added NVLINK_LINK_STATUS_INFO_V2
- > Added NVLINK_GET_STATUS_V1
- > Added NVLINK_GET_STATUS_V2

- > Added NV_GPU_INFO_V1
- > Added NV_GPU_INFO_V2
- > Added NV_EDID_DATA_V1
- > Added NV_EDID_DATA_V2

New/Updated Enums

- > Added NV_EDID_FLAG
- > Added NVAPI_NVLINK_DEVICE_INFO_DEVICE_TYPE
- > Added NVAPI_NVLINK_STATUS_LINK_STATE
- > Added NVAPI_NVLINK_STATUS_SUBLINK_RX_STATE
- > Added NVAPI_NVLINK_STATUS_SUBLINK_TX_STATE
- > Added NVAPI_D3D12_RAYTRACING_VALIDATION_FLAGS
- > Added NVAPI_D3D12_RAYTRACING_VALIDATION_MESSAGE_SEVERITY

New/Updated Unions

- > None

New Macros

- > Added NV_EDID_DATA_SIZE_MAX
- > Added NVAPI_NVLINK_COUNTER_MAX_TYPES
- > Added NVAPI_NVLINK_MAX_LINKS
- > Added NVAPI_NVLINK_CAPS_SUPPORTED
- > Added NVAPI_NVLINK_CAPS_P2P_SUPPORTED
- > Added NVAPI_NVLINK_CAPS_SYSTEMEM_ACCESS
- > Added NVAPI_NVLINK_CAPS_P2P_ATOMICS
- > Added NVAPI_NVLINK_CAPS_SYSTEMEM_ATOMICS
- > Added NVAPI_NVLINK_CAPS_PEX_TUNNELING
- > Added NVAPI_NVLINK_CAPS_SLI_BRIDGE
- > Added NVAPI_NVLINK_CAPS_SLI_BRIDGE_SENSABLE
- > Added NVAPI_NVLINK_CAPS_POWER_STATE_L0
- > Added NVAPI_NVLINK_CAPS_POWER_STATE_L1
- > Added NVAPI_NVLINK_CAPS_POWER_STATE_L2
- > Added NVAPI_NVLINK_CAPS_POWER_STATE_L3
- > Added NVAPI_NVLINK_CAPS_VALID
- > Added NVAPI_NVLINK_CAPS_NVLINK_VERSION_INVALID
- > Added NVAPI_NVLINK_CAPS_NVLINK_VERSION_1_0

- > Added NVAPI_NVLINK_CAPS_NVLINK_VERSION_2_0
- > Added NVAPI_NVLINK_CAPS_NCI_VERSION_INVALID
- > Added NVAPI_NVLINK_CAPS_NCI_VERSION_1_0
- > Added NVAPI_NVLINK_CAPS_NCI_VERSION_2_0
- > Added NVAPI_NVLINK_DEVICE_INFO_DEVICE_ID_FLAGS_NONE
- > Added NVAPI_NVLINK_DEVICE_INFO_DEVICE_ID_FLAGS_PCI
- > Added NVAPI_NVLINK_DEVICE_INFO_DEVICE_ID_FLAGS_UUID
- > Added NVAPI_NVLINK_STATUS_PHY_NVHS
- > Added NVAPI_NVLINK_STATUS_PHY_GRS
- > Added NVAPI_NVLINK_STATUS_PHY_INVALID
- > Added NVAPI_NVLINK_STATUS_NVLINK_VERSION_1_0
- > Added NVAPI_NVLINK_STATUS_NVLINK_VERSION_2_0
- > Added NVAPI_NVLINK_STATUS_NVLINK_VERSION_INVALID
- > Added NVAPI_NVLINK_STATUS_NCI_VERSION_1_0
- > Added NVAPI_NVLINK_STATUS_NCI_VERSION_2_0
- > Added NVAPI_NVLINK_STATUS_NCI_VERSION_INVALID
- > Added NVAPI_NVLINK_STATUS_NVHS_VERSION_1_0
- > Added NVAPI_NVLINK_STATUS_NVHS_VERSION_INVALID
- > Added NVAPI_NVLINK_STATUS_GRS_VERSION_1_0
- > Added NVAPI_NVLINK_STATUS_GRS_VERSION_INVALID
- > Added NVAPI_NVLINK_STATUS_CONNECTED_TRUE
- > Added NVAPI_NVLINK_STATUS_CONNECTED_FALSE
- > Added NVAPI_NVLINK_STATUS_LOOP_PROPERTY_LOOPBACK
- > Added NVAPI_NVLINK_STATUS_LOOP_PROPERTY_LOOPOUT
- > Added NVAPI_NVLINK_STATUS_LOOP_PROPERTY_NONE
- > Added NVAPI_NVLINK_STATUS_REMOTE_LINK_NUMBER_INVALID
- > Added NVAPI_NVLINK_REFCLK_TYPE_INVALID
- > Added NVAPI_NVLINK_REFCLK_TYPE_NVHS
- > Added NVAPI_NVLINK_REFCLK_TYPE_PEX

New Errors

- > None

TCC Support

- > None

Deprecated NVAPI Functions

- > None

Deprecated Enums

- > None

NVAPIDriverSettings Additions/Removals

- > Removed enum value PS_SHADERDISKCACHE_FLAGS_BACKEND_GFN_DLL from EValues_PS_SHADERDISKCACHE_FLAGS
- > Added enum value PS_SHADERDISKCACHE_FLAGS_BACKEND_DLL to EValues_PS_SHADERDISKCACHE_FLAGS
- > Added enum value PS_SHADERDISKCACHE_FLAGS_ALL_BUILDS_MISMATCH_TEST to EValues_PS_SHADERDISKCACHE_FLAGS

NVAPI Security Information

User administrator privilege is required to access certain driver features per NVIDIA's overall security vision. This helps mitigate the impact of malware.

Each API requiring administrator access, will return a NVAPI_INVALID_USER_PRIVILEGE error, when run with standard user privilege.

The application will require Administrator privileges to access this API, which can be elevated to a higher permission level by selecting "Run as Administrator" in Admin approval mode.

The following APIs now require Administrator privileges:

- > NvAPI_GSync_SetSyncStateSettings
- > NvAPI_GSync_SetControlParameters

Notice

This document is provided for information purposes only and shall not be regarded as a warranty of a certain functionality, condition, or quality of a product. NVIDIA Corporation ("NVIDIA") makes no representations or warranties, expressed or implied, as to the accuracy or completeness of the information contained in this document and assumes no responsibility for any errors contained herein. NVIDIA shall have no liability for the consequences or use of such information or for any infringement of patents or other rights of third parties that may result from its use. This document is not a commitment to develop, release, or deliver any Material (defined below), code, or functionality.

NVIDIA reserves the right to make corrections, modifications, enhancements, improvements, and any other changes to this document, at any time without notice.

Customer should obtain the latest relevant information before placing orders and should verify that such information is current and complete.

NVIDIA products are sold subject to the NVIDIA standard terms and conditions of sale supplied at the time of order acknowledgement, unless otherwise agreed in an individual sales agreement signed by authorized representatives of NVIDIA and customer ("Terms of Sale"). NVIDIA hereby expressly objects to applying any customer general terms and conditions with regards to the purchase of the NVIDIA product referenced in this document. No contractual obligations are formed either directly or indirectly by this document.

NVIDIA products are not designed, authorized, or warranted to be suitable for use in medical, military, aircraft, space, or life support equipment, nor in applications where failure or malfunction of the NVIDIA product can reasonably be expected to result in personal injury, death, or property or environmental damage. NVIDIA accepts no liability for inclusion and/or use of NVIDIA products in such equipment or applications and therefore such inclusion and/or use is at customer's own risk.

NVIDIA makes no representation or warranty that products based on this document will be suitable for any specified use. Testing of all parameters of each product is not necessarily performed by NVIDIA. It is customer's sole responsibility to evaluate and determine the applicability of any information contained in this document, ensure the product is suitable and fit for the application planned by customer, and perform the necessary testing for the application in order to avoid a default of the application or the product. Weaknesses in customer's product designs may affect the quality and reliability of the NVIDIA product and may result in additional or different conditions and/or requirements beyond those contained in this document. NVIDIA accepts no liability related to any default, damage, costs, or problem which may be based on or attributable to: (i) the use of the NVIDIA product in any manner that is contrary to this document or (ii) customer product designs.

No license, either expressed or implied, is granted under any NVIDIA patent right, copyright, or other NVIDIA intellectual property right under this document. Information published by NVIDIA regarding third-party products or services does not constitute a license from NVIDIA to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property rights of the third party, or a license from NVIDIA under the patents or other intellectual property rights of NVIDIA.

Reproduction of information in this document is permissible only if approved in advance by NVIDIA in writing, reproduced without alteration and in full compliance with all applicable export laws and regulations, and accompanied by all associated conditions, limitations, and notices.

THIS DOCUMENT AND 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. TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL NVIDIA BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF ANY USE OF THIS DOCUMENT, EVEN IF NVIDIA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Notwithstanding any damages that customer might incur for any reason whatsoever, NVIDIA's aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms of Sale for the product.

Trademarks

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

VESA DisplayPort

DisplayPort and DisplayPort Compliance Logo, DisplayPort Compliance Logo for Dual-mode Sources, and DisplayPort Compliance Logo for Active Cables are trademarks owned by the Video Electronics Standards Association in the United States and other countries.

HDMI

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

OpenCL

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

Copyright

© 2023 NVIDIA Corporation. All rights reserved.