



NVAPI Open Source SDK for Driver Release 570

Release Notes

Table of Contents

NVAPI Release Notes.....	3
Introduction.....	3
Changes in NVAPI for Driver Release 570.....	4
New Functions	4
New/Updated Structures.....	4
New/Updated Enums.....	6
New Unions	8
New Macros	8
New Errors.....	8
TCC Support	8
MCDM Support.....	9
NVAPI Security Information	9
Deprecated NVAPI Functions	9
Deprecated Enum Values	9
NVAPIDriverSettings Additions/Removals	9
HLSL Extension Additions/Removals	10
NVAPI Security Information.....	10
Sample Code	11

NVAPI Release Notes

Introduction

NVAPI is NVIDIA Corporation's core software development kit that allows direct 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.

The following files are provided by NVIDIA:

- > nvapi.h
- > nvapi_interface.h
- > nvapi_lite_common.h
- > nvapi_lite_d3dext.h
- > nvapi_lite_salend.h
- > nvapi_lite_salstart.h
- > nvapi_lite_sli.h
- > nvapi_lite_stereo.h
- > nvapi_lite_surround.h
- > NvApiDriverSettings.c
- > NvApiDriverSettings.h
- > nvHLSLExtns.h
- > nvHLSLExtnsInternal.h
- > nvShaderExtnEnums.h
- > \x86\nvapi.lib
- > \amd64\nvapi64.lib
- > \docs\NVAPI_Reference_OpenSource.chm
- > \docs\NVAPI_SDKs_Samples_and_Tools_License_Agreement(Public).pdf

These release notes describe the new features, enhancements, and changes in the NVAPI SDK for this release.

Changes in NVAPI for Driver Release 570

New Functions

- > Added `NvAPI_GPU_GetEncoderStatistics`
- > Added `NvAPI_GPU_GetEncoderSessionsInfo`
- > Added `NvAPI_D3D12_ConvertCooperativeVectorMatrix`
- > Added `NvAPI_D3D12_ConvertCooperativeVectorMatrixMultiple`
- > Added `NvAPI_D3D12_GetPhysicalDeviceCooperativeVectorProperties`
- > Added `NvAPI_D3D12_GetRaytracingMultiIndirectClusterOperationRequirementsInfo`
- > Added `NvAPI_D3D12_RaytracingExecuteMultiIndirectClusterOperation`
- > Added `NvAPI_D3D12_GetRaytracingPartitionedTlasIndirectPrebuildInfo`
- > Added `NvAPI_D3D12_BuildRaytracingPartitionedTlasIndirect`
- > Added `NvAPI_NGX_GetNGXOverrideState`
- > Added `NvAPI_NGX_SetNGXOverrideState`
- > Added `NvAPI_Vulkan_InitLowLatencyDevice`
- > Added `NvAPI_Vulkan_DestroyLowLatencyDevice`
- > Added `NvAPI_Vulkan_GetSleepStatus`
- > Added `NvAPI_Vulkan_SetSleepMode`
- > Added `NvAPI_Vulkan_Sleep`
- > Added `NvAPI_Vulkan_GetLatency`
- > Added `NvAPI_Vulkan_SetLatencyMarker`
- > Added `NvAPI_Vulkan_NotifyOutOfBandVkQueue`
- > Added `NvAPI_RegisterRiseCallback`
- > Added `NvAPI_RequestRise`
- > Added `NvAPI_UninstallRise`

New/Updated Structures

- > Added `NV_ENCODER_STATISTICS_V1`
- > Added `NV_ENCODER_PER_SESSION_INFO_V1`
- > Added `NV_ENCODER_SESSIONS_INFO_V1`
- > Added `rrx1k` to `NV_GET_VIRTUAL_REFRESH_RATE_DATA_V1`
- > Added `bIsGamingVrr` to `NV_GET_VIRTUAL_REFRESH_RATE_DATA_V1`
- > Added `rrx1k` to `NV_SET_VIRTUAL_REFRESH_RATE_DATA_V1`
- > Added `bIsGamingVrr` to `NV_SET_VIRTUAL_REFRESH_RATE_DATA_V1`

- > Added vendorInternal to NV_ASYNC_FRAME_MARKER_PARAMS_V1
- > Added NVAPI_D3D12_RAYTRACING_GEOMETRY_SPHERES_DESC
- > Added NVAPI_D3D12_RAYTRACING_GEOMETRY_LSS_DESC
- > Added spheres to NVAPI_D3D12_RAYTRACING_GEOMETRY_DESC_EX
- > Added lss to NVAPI_D3D12_RAYTRACING_GEOMETRY_DESC_EX
- > Added NVAPI_DEVICE_OR_HOST_ADDRESS
- > Added NVAPI_CONVERT_COOPERATIVE_VECTOR_MATRIX_DESC_V1
- > Added NVAPI_COOPERATIVE_VECTOR_PROPERTIES_V1
- > Added
NVAPI_D3D12_RAYTRACING_MULTI_INDIRECT_CLUSTER_OPERATION_INPUT_CLASSES_DESC
- > Added
NVAPI_D3D12_RAYTRACING_MULTI_INDIRECT_CLUSTER_OPERATION_INPUT_TRIANGLES_DESC
- > Added
NVAPI_D3D12_RAYTRACING_MULTI_INDIRECT_CLUSTER_OPERATION_INPUT_MOVES_DESC
- > Added
NVAPI_D3D12_RAYTRACING_MULTI_INDIRECT_CLUSTER_OPERATION_INPUTS
- > Added
NVAPI_D3D12_RAYTRACING_MULTI_INDIRECT_CLUSTER_OPERATION_REQUIREMENTS_INFO
- > Added
NVAPI_GET_RAYTRACING_MULTI_INDIRECT_CLUSTER_OPERATION_REQUIREMENTS_INFO_PARAMS_V1
- > Added
NVAPI_D3D12_RAYTRACING_ACCELERATION_STRUCTURE_MULTI_INDIRECT_CLUSTER_ARGS
- > Added
NVAPI_D3D12_RAYTRACING_ACCELERATION_STRUCTURE_MULTI_INDIRECT_TRIANGLE_CLUSTER_ARGS
- > Added
NVAPI_D3D12_RAYTRACING_ACCELERATION_STRUCTURE_MULTI_INDIRECT_TRIANGLE_TEMPLATE_ARGS
- > Added
NVAPI_D3D12_RAYTRACING_ACCELERATION_STRUCTURE_MULTI_INDIRECT_INSTANTIATE_TEMPLATE_ARGS
- > Added
NVAPI_D3D12_RAYTRACING_ACCELERATION_STRUCTURE_MULTI_INDIRECT_MOVE_ARGS
- > Added NVAPI_D3D12_RAYTRACING_MULTI_INDIRECT_CLUSTER_OPERATION_DESC
- > Added
NVAPI_RAYTRACING_EXECUTE_MULTI_INDIRECT_CLUSTER_OPERATION_PARAMS_V1

- > Added NVAPI_D3D12_BUILD_RAYTRACING_PARTITIONED_TLAS_INDIRECT_INPUTS
- > Added NVAPI_D3D12_BUILD_RAYTRACING_PARTITIONED_TLAS_INDIRECT_PREBUILD_INFO
- > Added NVAPI_GET_BUILD_RAYTRACING_PARTITIONED_TLAS_INDIRECT_PREBUILD_INFO_PARAMS_V1
- > Added NVAPI_D3D12_BUILD_RAYTRACING_PARTITIONED_TLAS_OP_ARG_WRITE_INSTANCE
- > Added NVAPI_D3D12_BUILD_RAYTRACING_PARTITIONED_TLAS_OP_ARG_UPDATE_INSTANCE
- > Added NVAPI_D3D12_BUILD_RAYTRACING_PARTITIONED_TLAS_OP_ARG_WRITE_PARTITION
- > Added NVAPI_D3D12_BUILD_RAYTRACING_PARTITIONED_TLAS_OP
- > Added NVAPI_D3D12_BUILD_RAYTRACING_PARTITIONED_TLAS_INDIRECT_DESC
- > Added NVAPI_BUILD_RAYTRACING_PARTITIONED_TLAS_INDIRECT_PARAMS_V1
- > Added NV NGX_DLSS_OVERRIDE_GET_STATE_PARAMS_V1
- > Added NV NGX_DLSS_OVERRIDE_SET_STATE_PARAMS_V1
- > Added NV_VULKAN_GET_SLEEP_STATUS_PARAMS_V1
- > Added NV_VULKAN_SET_SLEEP_MODE_PARAMS_V1
- > Added NV_VULKAN_LATENCY_RESULT_PARAMS_V1
- > Added NV_VULKAN_LATENCY_MARKER_PARAMS_V1
- > Added NV_RISE_CALLBACK_DATA_V1
- > Added NV_RISE_CALLBACK_SETTINGS_V1
- > Added NV_REQUEST_RISE_SETTINGS_V1
- > Added NV_UNINSTALL_RISE_SETTINGS_V1

New/Updated Enums

- > Added NV_GPU_ARCHITECTURE_GB200 to NV_GPU_ARCHITECTURE_ID
- > Added NV_GPU_ARCH_IMPLEMENTATION_GB202 to NV_GPU_ARCH_IMPLEMENTATION_ID
- > Added NV_GPU_SCANOUT_COMPOSITION_PARAMETER_SCANOUT_INTENSITY_METHOD to NV_GPU_SCANOUT_COMPOSITION_PARAMETER
- > Added NV_GPU_SCANOUT_COMPOSITION_PARAMETER_VALUE_SCANOUT_INTENSITY_NO_GAMMA to NV_GPU_SCANOUT_COMPOSITION_PARAMETER_VALUE

- > Added
NV_GPU_SCANOUT_COMPOSITION_PARAMETER_VALUE_SCANOUT_INTENSITY_COLOR_GAMMA to NV_GPU_SCANOUT_COMPOSITION_PARAMETER_VALUE
- > Added
NV_GPU_SCANOUT_COMPOSITION_PARAMETER_VALUE_SCANOUT_INTENSITY_COLOR_AND_OFFSET_GAMMA to NV_GPU_SCANOUT_COMPOSITION_PARAMETER_VALUE
- > Added NV_ENCODER_TYPE
- > Added OUT_OF_BAND_RENDER_PRESENT to NV_OUT_OF_BAND_CQ_TYPE
- > Added NVAPI_D3D12_RAYTRACING_CLUSTER_OPERATIONS_CAPS
- > Added NVAPI_D3D12_RAYTRACING_PARTITIONED_TLAS_CAPS
- > Added NVAPI_D3D12_RAYTRACING_SPHERES_CAPS
- > Added NVAPI_D3D12_RAYTRACING_LINEAR_SWEPT_SPHERES_CAPS
- > Added NVAPI_D3D12_RAYTRACING_CAPS_TYPE_CLUSTER_OPERATIONS to NVAPI_D3D12_RAYTRACING_CAPS_TYPE
- > Added NVAPI_D3D12_RAYTRACING_CAPS_TYPE_PARTITIONED_TLAS to NVAPI_D3D12_RAYTRACING_CAPS_TYPE
- > Added NVAPI_D3D12_RAYTRACING_CAPS_TYPE_SPHERES to NVAPI_D3D12_RAYTRACING_CAPS_TYPE
- > Added NVAPI_D3D12_RAYTRACING_CAPS_TYPE_LINEAR_SWEPT_SPHERES to NVAPI_D3D12_RAYTRACING_CAPS_TYPE
- > Added
NVAPI_D3D12_PIPELINE_CREATION_STATE_FLAGS_ENABLE_CLUSTER_SUPPORT to NVAPI_D3D12_PIPELINE_CREATION_STATE_FLAGS
- > Added
NVAPI_D3D12_PIPELINE_CREATION_STATE_FLAGS_ENABLE_SPHERE_SUPPORT to NVAPI_D3D12_PIPELINE_CREATION_STATE_FLAGS
- > Added NVAPI_D3D12_PIPELINE_CREATION_STATE_FLAGS_ENABLE_LSS_SUPPORT to NVAPI_D3D12_PIPELINE_CREATION_STATE_FLAGS
- > Added NVAPI_D3D12_RAYTRACING_GEOMETRY_TYPE_SPHERES_EX to NVAPI_D3D12_RAYTRACING_GEOMETRY_TYPE_EX
- > Added NVAPI_D3D12_RAYTRACING_GEOMETRY_TYPE_LSS_EX to NVAPI_D3D12_RAYTRACING_GEOMETRY_TYPE_EX
- > Added
NVAPI_D3D12_RAYTRACING_OPACITY_MICROMAP_SPECIAL_INDEX_CLUSTER_SKIP_OMM to NVAPI_D3D12_RAYTRACING_OPACITY_MICROMAP_SPECIAL_INDEX
- > Added NVAPI_D3D12_RAYTRACING_LSS_ENDCAP_MODE
- > Added NVAPI_D3D12_RAYTRACING_LSS_PRIMITIVE_FORMAT
- > Added NVAPI_COOPERATIVE_VECTOR_COMPONENT_TYPE
- > Added NVAPI_COOPERATIVE_VECTOR_MATRIX_LAYOUT
- > Added NVAPI_D3D12_RAYTRACING_MULTI_INDIRECT_CLUSTER_OPERATION_FLAGS

- > Added
NVAPI_D3D12_RAYTRACING_MULTI_INDIRECT_CLUSTER_OPERATION_CLUSTER_FLAGS
- > Added
NVAPI_D3D12_RAYTRACING_MULTI_INDIRECT_CLUSTER_OPERATION_GEOMETRY_FLAGS
- > Added NVAPI_D3D12_RAYTRACING_MULTI_INDIRECT_CLUSTER_OPERATION_TYPE
- > Added
NVAPI_D3D12_RAYTRACING_MULTI_INDIRECT_CLUSTER_OPERATION_INDEX_FORMAT
- > Added NVAPI_D3D12_RAYTRACING_MULTI_INDIRECT_CLUSTER_OPERATION_MODE
- > Added
NVAPI_D3D12_RAYTRACING_MULTI_INDIRECT_CLUSTER_OPERATION_MOVE_TYPE
- > Added
NVAPI_D3D12_RAYTRACING_MULTI_INDIRECT_CLUSTER_OPERATION_ADDRESS_RESOLUTION_FLAGS
- > Added NVAPI_D3D12_RAYTRACING_PARTITIONED_TLAS_FLAGS
- > Added NVAPI_D3D12_RAYTRACING_PARTITIONED_TLAS_INSTANCE_FLAGS
- > Added NVAPI_D3D12_RAYTRACING_PARTITIONED_TLAS_PARTITION_INDEX
- > Added NVAPI_D3D12_BUILD_RAYTRACING_PARTITIONED_TLAS_OP_TYPE
- > Added NV NGX_DLSS_OVERRIDE_BITFIELD
- > Added NV_VULKAN_LATENCY_MARKER_TYPE
- > Added NV_VULKAN_OUT_OF_BAND_QUEUE_TYPE
- > Added NV_RISE_CONTENT_TYPE

New Unions

- > None

New Macros

- > None

New Errors

- > None

TCC Support

- > None

MCDM Support

- > None

NVAPI Security Information

- > None

Deprecated NVAPI Functions

- > `NvAPI_D3D12_GetRaytracingDisplacementMicromapArrayPrebuildInfo`
- > `NvAPI_D3D12_BuildRaytracingDisplacementMicromapArray`
- > `NvAPI_D3D12_RelocateRaytracingDisplacementMicromapArray`
- > `NvAPI_D3D12_EmitRaytracingDisplacementMicromapArrayPostbuildInfo`

Deprecated Enum Values

- > None

NVAPIDriverSettings Additions/Removals

- > Added `NGX_DLAA_OVERRIDE_ID`
- > Added `NGX_DLSSG_MULTI_FRAME_COUNT_ID`
- > Added `NGX_DLSS_FG_OVERRIDE_ID`
- > Added `NGX_DLSS_FG_OVERRIDE_RESERVED_KEY1_ID`
- > Added `NGX_DLSS_FG_OVERRIDE_RESERVED_KEY2_ID`
- > Added `NGX_DLSS_OVERRIDE_OPTIMAL_SETTINGS_ID`
- > Added `NGX_DLSS_RR_MODE_ID`
- > Added `NGX_DLSS_RR_OVERRIDE_ID`
- > Added `NGX_DLSS_RR_OVERRIDE_RENDER_PRESET_SELECTION_ID`
- > Added `NGX_DLSS_RR_OVERRIDE_RESERVED_KEY1_ID`
- > Added `NGX_DLSS_RR_OVERRIDE_RESERVED_KEY2_ID`
- > Added `NGX_DLSS_SR_MODE_ID`
- > Added `NGX_DLSS_SR_OVERRIDE_ID`
- > Added `NGX_DLSS_SR_OVERRIDE_RENDER_PRESET_SELECTION_ID`
- > Added `NGX_DLSS_SR_OVERRIDE_RESERVED_KEY1_ID`
- > Added `NGX_DLSS_SR_OVERRIDE_RESERVED_KEY2_ID`

HLSL Extension Additions/Removals

- > Added DXR Cluster Geometry Extension
 - o `NvRtGetClusterID`
- > Added DXR Linear Swept Sphere Extension
 - o `NvRtSphereObjectPositionAndRadius`
 - o `NvRtLssObjectPositionsAndRadii`
 - o `NvRtIsSphereHit`
 - o `NvRtIsLssHit`
- > Added DXR RayQuery functions
 - o `__NvRtGetCandidateClusterID`
 - o `__NvRtGetCommittedClusterID`
 - o `__NvRtCandidateTriangleObjectPositions`
 - o `__NvRtCommittedTriangleObjectPositions`
 - o `__NvRtCandidateIsNonOpaqueSphere`
 - o `__NvRtCandidateIsNonOpaqueLss`
 - o `__NvRtCandidateLssHitParameter`
 - o `__NvRtCandidateSphereObjectPositionAndRadius`
 - o `__NvRtCandidateLssObjectPositionsAndRadii`
 - o `__NvRtCandidateBuiltinPrimitiveRayT`
 - o `__NvRtCommittedIsSphere`
 - o `__NvRtCommittedIsLss`
 - o `__NvRtCommittedLssHitParameter`
 - o `__NvRtCommittedSphereObjectPositionAndRadius`
 - o `__NvRtCommittedLssObjectPositionsAndRadii`
 - o `__NvRtCommitNonOpaqueBuiltinPrimitiveHit`

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 an `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 API requires administrator privileges:

- > `NvAPI_GPU_SetEDID`

Sample Code

The SDK package contains the `Sample_Code` directory, which provides code examples for the following features:

Feature	Sample Code Subdirectory	Sample Code
Custom timing	CustomTiming	CustomTiming.cpp
Display color control	DisplayColorControl	> DisplayColorControl.cpp > NVHelper.cpp > NVHelper.h
Display configuration	DisplayConfiguration	> DisplayConfiguration.cpp > targetver.h
GPU handle enumeration	GPUHandleEnumeration	gpuHandleEnumeration.c
QSYNC event registration	QSYNC_Event_Registration	QSYNC_Event_Registration.cpp
Sync configuration	Sync_Configuration	> Sync_Configuration.cpp > targetver.h
I2C	i2c	> i2c.cpp > targetver.h

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

© 2025 NVIDIA Corporation. All rights reserved.