



NVIDIA CAPTURE SDK FAQ

PG-06185-001_v1.07 | October 2018

Frequently Asked Questions



DOCUMENT CHANGE HISTORY

PG-06185-001_v1.07

Version	Date	Authors	Description of Change
0.5	5/13/2013	BO	Initial draft
0.8	12/5/2013	EY	Revised and added more FAQ
0.9	12/5/2013	NS/DG	Added 6 new FAQ items
1.0	1/17/2014	EY	First public release
1.01	9/16/2014	EY	Added updates on how to assign GPUs to PhysX with GRID bare metal multi-GPU configurations
1.02	11/3/2014	EY	Updated guide on specific GPU products supported.
1.03	7/6/2015	EY	Updated GRID SDK 4.0 guide with Maxwell products supported.
1.04	2/12/2016	EY	Updated for NVIDIA Capture SDK 5.0
1.05	8/8/2017	SD	Updated for NVIDIA Capture SDK 6.1
1.06	2/19/2018	MB	Update details of mouse capture in Linux systems
1.07	2/24/2018	SD	Remove references to deprecated interfaces

TABLE OF CONTENTS

NVIDIA Capture SDK FAQ	5
1. What is the NVIDIA Capture SDK? How do developers get access to it?.....	5
2. I have downloaded the SDK, how do I get started?	5
3. What components are included with NVIDIA Capture SDK? What are they used for?	5
4. What are the interfaces for NVFBC Capture?	5
5. When should NVFBC be used?	6
6. What hardware and drivers are required for the NVIDIA Capture SDK? OR.....	6
Which NVIDIA GPU products are supported by the NVIDIA Capture SDK?	6
7. What Operating Systems are supported by the NVIDIA Capture SDK?.....	6
8. Can I distribute any NVIDIA Capture SDK components to my customers?	6
9. I have questions about integrating my software with the NVIDIA Capture SDK. How do I contact support?	6
10. What are the available GRID products and their features?	7
11. How many simultaneous Capture and HW encode sessions can run on a single GRID\TESLA board?	7
12. Which Remote Desktop solution is recommended to connect with a GRID server or Amazon G2 instance? Can Microsoft Remote Desktop Protocol be used?.....	7
13. How do I perform DirectX 9 Capture and Encode using the NVIDIA Capture SDK?	8
14. How is NVFBC enabled?.....	8
15. What are the optimal encoder settings for desktop applications streaming?	8
16. What are the recommended hypervisors if I am using virtualized environment?	8
17. Which Virtual Environments are supported by the NVIDIA Capture SDK?.....	9
18. Can hybrid card combinations of GRID boards be used in GRID servers	9
.....	9
19. What recommended SBIOS/IPMI Firmware should I use?	9
20. What special Server settings do I need to set for the SBIOS and IPMI for use with the NVIDIA Capture SDK?.....	9
21. For AMD servers (i.e. Super Micro 1022GG), when obtaining the system topology information, I am unable to determine which NUMA domain the GRID device is closest to.	9
22. When setting up a new Server in a bare metal environment, there are problems installing and using the GRID devices.....	10
23. When streaming an H.264 bitstream, what are some of the methods I can use to recover from channel errors and lost frames on the decoder?.....	10
24. How do I capture the mouse input while using NVFBC?.....	11
25. How do I assign the appropriate GPU for PhysX on a multiple GPU device system that is running bare metal?	11
26. How do I determine when to capture a frame?.....	12
27. Does NVFBC support capture on Mosaic?	12
28. Does NVFBC support 10 bit ARGB capture format?	12
29. How do I use 10-bit HDR capture format?.....	12
30. Where are NVIFR samples?	12
31. Where is the NVFBC HW Encode sample?	12

NVIDIA CAPTURE SDK FAQ

1. What is the NVIDIA Capture SDK? How do developers get access to it?

The NVIDIA Capture SDK, previously called GRID SDK, enables fast capture and compression of the desktop display from supported NVIDIA GPUs. You will need to register in order to download the NVIDIA Capture SDK. Registration is free, and the download page can be found here:

<https://developer.nvidia.com/capture-sdk>.

Please refer to *NVIDIA Capture SDK release notes* for the list of supported GPUs.

2. I have downloaded the SDK, how do I get started?

For a list of recommended hardware, please refer to Question #23 of this FAQ. Also please refer to the *Guide to Setup a GRID Server* and the *NVIDIA Capture SDK Programming Guide* for how to begin using the SDK once you have downloaded the SDK and driver. These samples are simple examples for how to use the GRID APIs on Linux or Windows:

```
{Installation Directory}\Samples
```

3. What components are included with NVIDIA Capture SDK? What are they used for?

The NVIDIA Capture SDK publishes NVFBC : a software API that captures entire visible desktop to System Memory, DirectX 9, or CUDA buffers.

4. What are the interfaces for NVFBC Capture?

NVFBC has the interfaces listed below. Select the appropriate interface, depending on the method of capture that you want, and the destination for the data:

- NVFBC_TO_SYS captures the desktop and copies it to pinned system memory on

- the host.
- `NVFBC_TO_CUDA` captures the desktop and copies it to CUDA device memory on the GPU.
 - `NVFBC_TO_DX9VID` captures the desktop and copies it to a DX9 surface. This can then be sent to the HW encoder. (This interface is available on Windows only, not available on Linux)
 - `NVFBC_TO_GL` captures the desktop and copies it to an OpenGL texture. This can then be sent to the HW encoder. (This interface is available on Linux only, not available on Windows)

Note: `NVFBC_TO_DX9VID` on Windows (`NVFBC_TO_GL` on Linux) can be used when running a vGPU profile that supports two or more virtual machines sharing a single GPU. `NVFBC_TO_CUDA` is not supported for such vGPU profiles.

5. When should NVFBC be used?

NVFBC is better for remote desktop applications. For a more detailed explanation of the differences, please refer to the “Amazon G2 Instance Getting Started Guide” included with the SDK.

6. What hardware and drivers are required for the NVIDIA Capture SDK? Which NVIDIA GPU products are supported by the NVIDIA Capture SDK?

Please refer to the release notes included in the NVIDIA Capture SDK package you intend to use for a list of supported hardware and operating systems.

7. What Operating Systems are supported by the NVIDIA Capture SDK?

Microsoft Windows 7, Windows 8, Windows Server 2008 R2, Windows Server 2012 R2, Windows 8.1, Windows 10, and Linux OS. For servers running Windows, we recommend Windows 8 and Windows Server 2012 R2 because these operating systems are more efficient and can serve more concurrent users compared to Windows 7 and Windows Server 2008R2 respectively.

8. Can I distribute any NVIDIA Capture SDK components to my customers?

No, the NVIDIA Capture SDK is subject to NVIDIA’s Software License Agreement. Any re-distribution rights will require explicit permission from NVIDIA.

9. I have questions about integrating my software with the NVIDIA Capture SDK. How do I contact support?

Please refer to the Programming Guide and included documentation. For detailed questions, please contact GRID developer support at GridPublicSupport@nvidia.com.

10. What are the available GRID products and their features?

Please visit <https://www.nvidia.com/grid> for more information.

11. How many simultaneous Capture and HW encode sessions can run on a single GRID\TESLA board?

The total number of concurrent sessions for a system will depend on the games or applications being streamed, the hardware encoder settings being used, and the CPUs in the system. For workloads that are not very graphics-intensive, no. of simultaneous sessions while keeping the framerate above a certain threshold will be limited by the HW Encoder throughput, whereas for very graphics-intensive workloads, the Graphics engine throughput may have a greater impact on this number.

Information related to HW Encoder performance and throughput capacity can be found in the NVIDIA Video Codec SDK Documentation.

The nvidia-smi tool can be used to profile GPU utilization on the target system.

12. Which Remote Desktop solution is recommended to connect with a GRID server or Amazon G2 instance? Can Microsoft Remote Desktop Protocol be used?

(a) Remoting Solution: For setting up, debugging, and configuring these servers, it is recommended to use TeamViewer. This software allows a remote client to connect one desktop window at a time. In comparison, VNC also captures and streams with the NVIDIA GPU accelerated driver, but for bare metal systems, it captures all desktop windows. This adds additional overhead and results in reduced performance when streaming. The overhead for capturing one desktop window in TeamViewer is significantly less versus capturing all windows desktops with the VNC solution.

Note: TeamViewer uses a proprietary compression format for remote streaming. The NVENC H.264 hardware engine is not used by TeamViewer.

For streaming applications and games (not using H.264 for compression), TeamViewer is the recommended solution for streaming the desktop. For the best experience on GRID, use NVENC with H.264 compression for the best remote streaming experience.

(b) Remote Desktop: While it is more efficient in terms of bitrate and performance of remote graphics in comparison to VNC, Microsoft Remote Desktop uses a proprietary software based graphics driver that does not support all of the NVIDIA GPU accelerated capabilities, and does not enable the NVIDIA GRID driver. Any applications running under Microsoft Remote Desktop will not be using the NVIDIA driver and will not have full benefits of GPU acceleration.

Note: Windows Server 2008 R2 SP1 and later versions support for GPU acceleration of applications in the Remote Desktop environment. Please refer to the Windows documentation for

details.

13. How do I perform DirectX 9 Capture and Encode using the NVIDIA Capture SDK?

NVIDIA Capture SDK enables the application to receive output in DirectX 9 surfaces. These can be passed to NVENC API for video encoding. The NVFBCDx9NvEnc sample application demonstrates this. Please refer to the NVIDIA Capture SDK Sample Description document for guidance on all of the NVIDIA Capture SDK samples.

14. How is NVFBC enabled?

Here are the NVFBC settings:

Windows: NVFBC needs to be enabled using this tool after a clean installation.

```
a. {Installation Directory} \bin> NVFBCEnable.exe -enable
```

Linux: For the NVFBC GRID API functions, please refer to the flag `NVFBC_CREATE_CAPTURE_SESSION_PARAMS.bWithCursor`

With NVIDIA Capture SDK 5.0, there is a new API that allows you to enable NVFBC through the function `NvFBC_Enable(NVFBC_STATE NvFBCState)`. Please refer to the header `NVFBC.h` and the new `NvFBCEnableAPI` sample.

15. What are the optimal encoder settings for desktop applications streaming?

Please refer to NVIDIA Video Codec SDK documentation for encoder settings.

16. What are the recommended hypervisors if I am using virtualized environment?

We recommend XenServer 6.2 (with SP1) for NMOS for stability and performance. Other hypervisors which are supported are: Citrix, VMware, and KVM. XCP 1.5 is now deprecated.

Note: When using XenServer 6.2 with a K340 in a virtualized environment, the Xen distribution needs to be patched to enable device class configuration space matching and to ensure that the Windows VM running on Xen can boot with GPU passthrough. For K520 based servers, this step is not required.

You can find the patch under the XenPatch folder in the SDK. In the public SDK, it includes the path for Xen. For Xen, to apply the patch:

```
rpm -Uvh xen-device-model-1.8.0-89.7554.i686.rpm
```

This step is required to ensure that the Windows VM running on Xen can boot with GPU passthrough.

17. Which Virtual Environments are supported by the NVIDIA Capture SDK?

NMOS (NVIDIA Multi-OS). One GPU is attached to one Virtual Machine, vGPU profiles that support only one virtual machine per GPU, and vGPU profiles that support two or more virtual machines per GPU.

18. Can hybrid card combinations of GRID boards be used in GRID servers

Using hybrid combinations is not recommended. For example, mixing K340 and K520 GPUs in the same system may work fine, but is not the recommended way of setting up a bare metal (or virtualized) system. Mixing and matching configurations with Tesla M60 is also not recommended.

19. What recommended SBIOS/IPMI Firmware should I use?

Please contact your NVIDIA GRID support team representative for the recommended SBIOS/IPMI firmware.

20. What special Server settings do I need to set for the SBIOS and IPMI for use with the NVIDIA Capture SDK?

For the SBIOS: Setting “VT-d” for Intel platforms and ‘IOMMU’ for AMD platforms must be enabled for virtualization environment.

Note: For IPMI the FAN speed must be set to optimal for Virtualization and Bare Metal environments.

21. For AMD servers (i.e. Super Micro 1022GG), when obtaining the system topology information, I am unable to determine which NUMA domain the GRID device is closest to.

This system is a NUMA system and the System BIOS for NUMA is enabled. However, it fails when making the function call to retrieve the NUMA information. How do I resolve the problem?

Make sure the motherboard has SBIOS “H8DGG.926” or newer for the SuperMicro SM-1022 server. Update it to this SBIOS if it is older.

In order to verify that the GPU and CPU are on the same NUMA node, use a bandwidth test application (NVIDIA CUDA SDK contains a test application named ‘*bandwidthTest.exe*’ that is suitable for this purpose) to measure the PCI-e bandwidth between the CPU and GPU. You can compare other NUMA nodes to make sure you are getting the best I/O throughput using the bandwidthTest.exe tool from the CUDA Toolkit.

22. When setting up a new Server in a bare metal environment, there are problems installing and using the GRID devices.

The server OS is Windows 7 or Windows 2008 R2. The NVIDIA driver installation succeeds, but there are problems recognizing the GRID GPUs when I launch my application. What is the solution?

Please refer the GRID Game Server Configuration document that is included with the SDK. Sections 1.2 explain how to configure the server and set up the driver. Before installing the NVIDIA driver, please make sure:

For remote configuration and connection, use either a VNC Server or TeamViewer. Microsoft Remote Desktop uses a software VGA driver, which does not enable the NVIDIA GPU driver upon login.

Onboard VGA must be disabled for Windows 7 and Windows Server 2008 R2. Prior to installing the NVIDIA drivers, the onboard VGA must be disabled. Refer to the SuperMicro server manual or the GRID Game Server Configuration Guide) for information how to disable VGA jumper (**JPG1**).

For Windows 7 and Windows Server 2008 R2, GPUs from different vendors cannot be used at the same time. This is due to a restriction in the OS, as only one GPU vendor driver can be installed and running at a time. If the onboard VGA is enabled, the PCI-e based boards will not work. If the onboard VGA with the NVIDIA GPUs is required with the GRID server, we recommend using Windows 8 and Windows Server 2012 R2, as these OS versions remove this restriction.

Note: A benefit of using Windows 8 and Windows Server 2012 R2 is that both allow IPMI management tools to run. If the onboard VGA is disabled, IPMI is not available.

23. When streaming an H.264 bitstream, what are some of the methods I can use to recover from channel errors and lost frames on the decoder?

If the decoder loses an entire frame, you can use the following error recovery mechanisms:

1. Force an IDR-frame to be generated for the next frame on the encoder. This is the easiest and brute-force method for error recovery. However, this method is prone to problems because the IDR frame generated may have a lower quality/larger size (depending upon which RC mode is in use) and since the channel is already bad, there are higher chances of the IDR frame itself getting lost, resulting in cyclically worsening problem.
2. Force an intra-refresh wave. This is the recommended method for error recovery. This method has the advantage of lower channel overhead (since only part of the frame needs to be intra-coded), but the disadvantage that it takes slightly longer to recover from the error. Please consult NVIDIA Video SDK documentation for

implementation guidelines.

3. Use reference-picture-invalidation. This method works only if there is an upstream communication channel from the decoder to the encoder. To use this method, the decoder signals a lost frame to the encoder and the encoder then invalidates the reference frame. This method prevents the invalidated frames from being used for subsequently encoded frames as references and prevents error propagation. When encoder exhausts all reference frames (set via `dwMaxNumRefFrames`), it automatically generates an IDR frame. Please consult NVIDIA Video SDK documentation for implementation guidelines.

Note: In addition to the above, you can also use the error concealment on the decoder to avoid discarding an entire frame on the decoder. This works if the error impact is not large. Please refer to the NVENC API in NVIDIA Video SDK for more details

24. How do I capture the mouse input while using NVFBC?

The choice of having HW rendered mouse cursor blended on to the image grabbed by NVFBC is available only with the `NvFBCToSys` interface. The other two interfaces always blend the HW rendered mouse cursor on the desktop at server-end.

For `NvFBCToSys`, to enable capturing desktop images with the HW cursor blended on to the grabbed image, please refer to the documentation for the parameter `NVFBC_TOSYS_SETUP_PARAMS::bWithHWCursor` in the NVIDIA Capture SDK Programming Guide.

Please refer to the documentation for the API `NvFBCToSysGrabMouse` in the NVIDIA Capture SDK Programming Guide for details about capturing HW mouse images separately, so that they can be blended at client-side during streaming.

Note: On Linux, to capture the HW mouse cursor image separately, application should disable the mouse capture feature in NVFBC and use an existing Xlib API instead.

25. How do I assign the appropriate GPU for PhysX on a multiple GPU device system that is running bare metal?

PhysX uses CUDA which has the option to choose to run physics simulations on the GPU for better effects and performance. To specifically set which GPU to use for PhysX, you need to set the following environment flag before launching the game.

➤ `set PHYSXDEVICE_GPU_CUDA_ORDINAL=2`

This will use the GPU ordinal 2 for PhysX. You can also refer to the sample under `Tools\DXCUDADevices` for details on how to enumerate the DirectX and CUDA device ordinals.

26. How do I determine when to capture a frame?

We would like to capture a frame only when it has changed from the previous frame. Does a frame grabbed with NVFBC differ from the previous frame?

NVIDIA Capture SDK 5.0 includes a new SDK sample `NvFBCDX9DiffMap`. This sample will return a 128x128 difference map of the desktop and also capture a downscaled version of the desktop. This is particularly useful for remote desktop environments where the server may decide on a different encoding method depending on what has changed on the screen.

27. Does NVFBC support capture on Mosaic?

Mosaic capture is supported from driver version 372.54 onwards. Mosaic capture is limited by available GPU memory.

28. Does NVFBC support 10 bit ARGB capture format?

10-bit ARGB capture format is supported from driver version 372.54. New output format enum `NVFBC_ToXXX_ARGB10` is supported in NVIDIA Capture SDK 6.0. Any required 8-bit to 10-bit format conversion is performed by the driver.

29. How do I use 10-bit HDR capture format?

The "bHDRRequest" flag has been added to `NVFBC_ToXXX_SETUP_PARAMS` in NVIDIA Capture SDK 6.0. Capturing HDR is feasible only if this flag is set. The newly added "bIsHDR" flag in `NvFBCFrameGrabInfo` structure informs the client if the current captured frame is in HDR.

30. Where are NVIFR samples?

NVIFR API was deprecated starting with NVIDIA Capture SDK 7.0. NVIFR interface definitions, samples and documentation are not distributed, starting with NVIDIA Capture SDK 7.1. Please refer to NVIDIA Capture SDK 7.0 release notes for support details.

31. Where is the NVFBCHwEncode sample?

`NvFBCHwEncode` interface was deprecated starting with NVIDIA Capture SDK 5.0. `NvFBCHwEncode` interface definitions, samples and documentation are not distributed, starting with NVIDIA Capture SDK 7.1. Please refer to NVIDIA Capture SDK 7.1 release notes for support details.

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.

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 U.S. and other countries. Other company and product names may be trademarks of the respective companies with which they are associated.

Copyright

© 2017, 2018 NVIDIA Corporation. All rights reserved.