



NVIDIA Rivermax Frequently Asked Questions (FAQ)

Rev 2.2

Document Revision History

Doc_2.2

Version	Date	Description of Change
2.2	Nov 30, 2021	Rivermax 1.9 Release
2.1	Aug 3, 2021	Updated 2022-7 streams support
2.0	Jan 1, 2021	Updated the document based on Rivermax version 1.7 release
1.6	May 20, 2019	Updated list of currently supported adapters.
1.5	April 01, 2019	Updated various sections throughout the document.
1.4	Sept 06, 2018	Official release of this document.
1.3	April 12, 2018	Final Remarks/Edits.
1.2	March 26, 2018	Added additional FAQ.
1.1	March 15, 2018	Initial version of the document.
1.0	March 14, 2018	Updated list of currently supported adapters.

Table of Contents

- INTRODUCTION..... 4**
- FAQ: NVIDIA RIVERMAX..... 5**
 - 1. Supported Operating Systems/Network Interface Cards (NICs) 5
 - 2. Virtual Machine (VM) and Containers 7
 - 3. SDP, RTP and Network Headers..... 7
 - 4. Supported Video Formats 9
 - 5. Miscellaneous 10

INTRODUCTION

NVIDIA® Rivermax® offers a unique IP-based solution for any media and data streaming use case. Rivermax together with NVIDIA GPU accelerated computing technologies unlocks innovation for a wide range of applications in Media and Entertainment (M&E), Broadcast, Healthcare, Smart Cities and more.

Rivermax leverages NVIDIA ConnectX® and BlueField (DPU) hardware streaming acceleration technology that enables direct data transfers to and from the GPU, delivering best-in-class throughput and latency with minimal CPU utilization for streaming workloads.

Rivermax implements a dedicated solution for IP-based video streaming of HD to Ultra HD flows. Moreover, Rivermax is the only fully virtualized streaming solution which complies with the stringent timing and traffic flow requirement of the SMPTE ST2110-21 specification - Enabling the future cloud-based broadcast solutions.

The Rivermax library is a cross-platform library, with support for both Linux and Windows OSs over Bare Metal and Virtualized solutions. For more details, go to the [Rivermax Product Page](#)

The sections below address FAQ related to NVIDIA Rivermax.

FAQ: NVIDIA RIVERMAX

1. Supported Operating Systems/Network Interface Cards (NICs)

Question	What Operating Systems (OS) does Rivermax support? Linux/Windows?
Answer	<p>Rivermax supports both Linux and Windows</p> <ul style="list-style-type: none">• Windows Server 2019• Windows Server 2016• Windows 10 Pro• Linux distros: Ubuntu 18.04, 20.04; CentOS 7.7, 7.9, 8.1; and Red Hat 7.8, 7.9, 8.1 and 8.3 <p>For a complete update list, go to Rivermax Product Page</p>

Question	Which NVIDIA Network Adapters (NICs) does Rivermax support?
Answer	<p>Rivermax supports both Linux and Windows.</p> <p>For Windows OS and strict timing requirements (PTP, SMPTE ST2110) BlueField-2 DPU is the recommended adapter.</p> <p>Rivermax supports the following list of adapter cards:</p> <ul style="list-style-type: none">• [Not for new projects] MCX512A-ACAT - ConnectX-5 10/25GbE• [Not for new projects] MCX516A-CDAT - ConnectX-5 Ex 100GbE• MCX623106AN-CDAT - ConnectX-6 Dx 100GbE• MCX621102AN-ADAT - Connect6 Dx 25GbE• MCX623105AN-CDAT ConnectX®-6 Dx 100 GbE single port• MBF2M516A-CENOT-Bluefield®-2 – 100GbE x16 QSFP• MBF2H332A-AENOT-BlueField-2 – 25GbE x8 SFP• MCX631102AN-ADAT - ConnectX-6 Lx – 25GbE <p>For a complete supported platforms and Network adapter selection guide , go to Rivermax SDK Page</p>

Question	How can I ensure that Rivermax is using the supported NIC type?
Answer	The device IP is used when creating Rivermax resources. Rivermax uses the IP address to detect if the associated NVIDIA NIC is supported by the Rivermax library. If the NIC is not supported, the API returns an error code.

Question	Does the same Rivermax SDK work on all supported NIC types?
Answer	Yes. Advanced features are hardware-dependent; this means they might be supported on specific NICs and not on other types.

Question	Does Rivermax support Arm architecture?
Answer	Yes. Rivermax supports both x86 and Arm architectures.

Question	Does Rivermax support AMD ?
Answer	Yes. Rivermax supports both AMD and Intel. Refer to the tuning guide for tips on achieving maximum performance.

Question	Which applications are supported by Rivermax SDK?
Answer	<p>Rivermax provides very high bandwidth, low latency, GPU-Direct and zero memory copy and can be used with any data streaming application.</p> <p>Some examples of typical use cases suitable for using Rivermax include:</p> <ul style="list-style-type: none"> • Broadcast use cases: <ul style="list-style-type: none"> ○ IP-enabled TV Studio Media (SMPTE ST 2110) ○ Virtual Production ○ H.264 and H.265 compressed data • Applications that need to support compressed video streaming such as JPEG-XS over SMPTE ST2110-22 • Healthcare use cases that require high bandwidth imaging ingest (camera, ultrasound etc.) and DICOM-RTV that is based on SMPTE ST2110 • Immersive Live Shows • Immersive Video Experiences (i.e., . wall displays) • Location-Based Entertainment <p>Refer to NVIDIA Rivermax Product Page for customers use cases</p>

2. Virtual Machine (VM) and Containers

Question	Does Rivermax support a VM/SR-IOV?
Answer	Rivermax supports SR-IOV virtualization on top of VMware ESXi and OpenStack (KVM)

Question	Can Rivermax run from Docker Containers
Answer	Yes. Rivermax can run inside a container without any impact on performance or functionality; code examples are available.

3. SDP, RTP and Network Headers

Question	Does Rivermax generate RTP packets?
Answer	No. Rivermax receives buffers containing RTP headers and an RTP payload from the application from which it generates network packets (adds MAC/IP/UDP). The Rivermax SDK is provided with full code examples which show how to generate the RTP headers.

Question	Does Rivermax create the Ethernet & IP layer?
Answer	Yes. The Rivermax layer creates all the Ethernet/IP/UDP headers (in media specific use cases this information is extracted from an SDP provided as input to Rivermax).

Question	Does Rivermax support receiving or transmitting RTP payload/headers with variable length?
Answer	Yes. For send requests, the sizes must be indicated upon the output stream creation via <code>rmax_mem_block.app_hdr_size_arr/rmax_mem_block.data_size_arr</code> . For receive requests, the min and max sizes must be indicated upon the input stream creation via <code>rmax_in_memblock.min_size</code> and <code>rmax_in_memblock.max_size</code> for both payload and header packets.

Note: Rivermax input stream can receive traffic that consists packets with variant sizes that can dynamically change during runtime but must be in the range:

[min_size - max_size]

Note: It is highly recommended to work with constant length in order to fully utilize hardware offloads and achieve the optimal performance.

Question	On the sender application (application TX), can the data be separated from the RTP headers? Can application (RTP) headers and data be separated?
Answer	Yes. Rivermax supports both options. The RTP headers can be sent/received in a separate buffer from the data or together with the data in the same buffer.

Question	Is the usage of SDP mandatory?
Answer	No. Rivermax supports any data stream that is based on UDP/IP. SDP provides important information used by Rivermax to configure the correct parameters for the transmitted stream. It is used by applications that support SMPTE ST 2110. A generic TX API can be used for other types of application that would like to benefit from the Rivermax performance but don't support SMPTE ST 2110. Receiving any data stream by Rivermax doesn't require an SDP file.

Question	Media attributes are provided to Rivermax via string, why?
Answer	For media specific streaming use cases, Rivermax uses a standard SDP format which is textual. The SDP string stores a lot of information. Some of this information is used by Rivermax today, while some of it may be used in future versions. The Generic TX API doesn't require media attributes.

Question	How is the SDP content provided to Rivermax?
Answer	An SDP must be provided to Rivermax for creation of TX streams for media use cases only. The SDP is provided as a null terminated character string consisting of records as defined in RFC4566 (including the CRLF character sequence terminating each record. Rivermax also accepts SDPs whose records are terminated by a single newline character).

Question	How can I know which SDP attributes are required by Rivermax?
Answer	Documentation of <code>rmax_out_create_stream()</code> routine explains what the minimal set of required SDP attributes is. Please refer to the Rivermax SDK page for SDP requirements. Example SDP files can be found in the provided application example directory provided in the Rivermax package.

Question	Does Rivermax support offload of localhost addresses?
Answer	<p>No.</p> <p>Using addresses in the 127.0.0.0/8 subnet is not supported for loopback offload. To offload packets that must be looped back, use one of the following:</p> <ul style="list-style-type: none"> • NIC local IP address • Multicast address (which the host has joined) - please review documentation for further details

Question	Does Rivermax support PTP? (timing synchronization)
Answer	<p>Yes.</p> <p>Using Rivermax, the first packet of each frame must be committed with the time parameter set to PTP time (in nanoseconds) at which it must be sent.</p> <p>Time synchronization can be achieved using the BlueField-2 NIC with DPU time and scheduling service enabled. This solution provides OS and virtualization agnostic PTP synchronization which is completely offloaded from the CPU.</p> <p>In this case Linux PTP4L runs on the DPU arm cores and locks the NIC's Real Time Clock. Rivermax provides the mechanism of reading the PTP time from the NIC's Real Time Clock. For Linux OS the ConnectX-6 Dx and PTP4L on the host can lock the NIC's Real Time Clock and achieve the same performance.</p>

Question	Does Rivermax support unicast and multicast IP?
Answer	<p>Yes.</p> <p>Rivermax supports both unicast and multicast IP formats.</p>

4. Supported Video Formats

Question	Which SMPTE streams are supported by Rivermax?
Answer	Rivermax supports 2110-20, 2110-30, 2110-40, 2022-7, 2022-6, 2022-8, 2110-31 and 2110-22 (with external stack).

Question	Does Rivermax support the 2022-7 streams and how is this done?
Answer	<p>Yes.</p> <ul style="list-style-type: none"> • For transmit: <p>The application can request sending redundant streams by indicating this via the SDP string using the standard a=group attribute with the DUP semantic and associating corresponding media blocks to the group.</p> <p>When redundancy is requested, Rivermax internally replicates chunks to multiple streams.</p> <ul style="list-style-type: none"> • For receive:

	<p>ConnectX-5 provides software support: Rivermax supports 2022-7 on the receive side simply by allowing the application to receive 2 media streams. The redundancy is handled in the application layer.</p> <p>ConnectX-6 Dx, BlueField-2 and up provides hardware offload: Redundant stream's seamless reconstruction uses ConnectX-6 Dx hardware.</p>
--	--

Question	Does Rivermax support 8K UHD streams?
Answer	<p>ConnectX-5 Rivermax Linux version supports receiving a single 8K UHD video stream format of 20-80Gb/s. Accurate pacing for an 8K stream according to the SMPTE ST 2110 expected to be supported by ConnectX-6 Dx and BlueField-2 and up</p> <p>An example code will be provided in future releases.</p>

Question	Which video formats are supported by Rivermax?
Answer	<p>Using Rivermax media API, YCbCr and RGB ST2110-20 streams, compressed video such as JPGE-XS ST2110-22 streams and SDI over IP video ST2022-6 and ST2022-8 streams are supported. Rivermax supports sending any video content (not necessarily SMPTE compliant) using its TX generic API.</p>

Question	Does Rivermax support both interlaced & progressive (i.e. 1080i & 1080p) modes?
Answer	<p>Yes.</p> <p>Rivermax supports both interlaced and progressive scan modes.</p>

5. Miscellaneous

Question	Does Rivermax support NMOS IS-04 /IS-05?
Answer	<p>Yes.</p> <p>NMOS is used for discovering, registering, and managing Media flows—it is part of the application layer.</p> <p>The NMOS specifications describe a control plane that makes a ST 2110-based infrastructure manageable and simpler to operate.</p> <p>The Rivermax SDK includes an NMOS media_node code example, which utilizes the Rivermax media_sender and media_receiver code examples along with the Sony nmos-cpp implementation, to demonstrate an NMOS control plane for Rivermax.</p> <p>The media_node is provided both as a stand-alone example and as a containerized docker solution for best out of the box experience.</p> <p>Full documentation of the media_node can be found on the Rivermax SDK Page SDK</p>

Question	Which SMPTE sender types are supported by Rivermax?
Answer	Rivermax supports sending the following types of streams: ST2110-20 Narrow Sender (2110TPN), ST2021-22 compressed video, ST2110-30 audio, ST2110-31 compressed audio and ST2110-40 ancillary. Rivermax also supports ST2022-6, ST2022-7 and ST2022-8 senders (sending ST2022-8 streams is supported only with BlueField-2 or ConnectX-6 Dx when configured to work with Real Time Clock mode).

Note: ST2110-20 Narrow Linear and Wide senders types have not yet been fully tested.

Question	Does Rivermax support the extended UDP Size?
Answer	Yes, make sure you increase interface MTU and set Rivermax strides sizes correctly.

Question	Does Rivermax support GPUDirect (direct access to GPU memory)?
Answer	Rivermax supports GPUDirect—memory can be allocated directly on the GPU. Example code is provided as part of Rivermax release 1.7 Windows GPUDirect is also supported as part of Rivermax release 1.8 on selected GPUs Refer to the Rivermax SDK Page_ for more details.

Question	Memory allocation: Is it done by Rivermax or user/application?
Answer	Both modes are supported: memory can be allocated by application or by Rivermax. For input stream, it is necessary to precede the memory allocation with a call to <code>rmax_in_query_buffer_size()</code> to determine the amount of memory required to be allocated by the application.

Question	Does Rivermax check the content of the data?
Answer	No

Question	Can an application dynamically, during runtime, change the output stream data/app headers sizes or can these sizes be set only once upon output stream creation?
Answer	The sizes must be set once upon output stream creation. Dynamic modification of data/app header sizes is not supported for media streams. Using the Tx Generic API, the data/header sizes can be modified on the fly.

Question	Does Rivermax support Receive and Transmit?
Answer	Yes. Both are supported and provide very high bandwidth with low CPU consumption

Question	Does Rivermax provide information such as how many buffers are available, committed, or sent?
Answer	<p>Yes.</p> <p>Rivermax SDK provides the rmax_stat tool to collect real time statistics about the Rivermax streams currently in progress. The rmax_stat tool provides the process ID of the specific application that is sending or receiving via Rivermax for which statistics are to be collected. It then provides both general and internal useful information which allows an in-depth analysis of the stream being sent or received.</p> <p>Please refer to the Rivermax user manual for details regarding the precise statistics collected by Rivermax and full instructions on how to enable statistics collection.</p>

Question	Does Rivermax support audio or ancillary on top of video?
Answer	Yes—these are 2110-30, 2110-31 and 2110-40. An example code for 2110-30 and 2110-40 is available as part of the SDK.

Question	Does the Rivermax SDK provide an example that sends real video content?
Answer	Yes—Rivermax media_sender can send a pre-prepared video template file in YCbCr format (part of the example code). Rivermax SDK also provides rivermax_player—an example code that integrates FFMPEG with Rivermax. This example allows sending an MP4 video file.

Question	Is Rivermax SMPTE ST 2110 JT-NM tested ?
Answer	<p>Rivermax is “JT-NM Tested” and has been tested for SMPTE ST 2110-20/30/40, 2022-7 compliance. Rivermax also supports ST2022-6 and ST 2110-31. The full report can be downloaded from the JT-NM.org: https://jt-nm.org/jt-nm_tested/index.shtml</p> <p>Rivermax is been JT-NM tested on each SDK release</p>

Question	Which NVIDIA SDK support Rivermax?
Answer	Rivermax is part of DeepStream 6.0 and Clara Holoscan SDK.

Question	Can I use SDKManager to install Rivermax?
Answer	Yes. The SDKManager can be used to install Rivermax SDK on selected platforms, such as Jetson Xavier AGX and Clara AGX Developer Kit.

Question	How to start using Rivermax
Answer	<p>Refer to the Rivermax SDK Page: https://developer.nvidia.com/networking/rivermax. Press the link to “Get Started” and then press “Join Now” and enter your details.</p> <p>Once accepted, purchase the required hardware, obtain a Rivermax license, and use the SDK example code as a starting point.</p>

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 NON-INFRINGEMENT, 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.

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.

Trademarks

NVIDIA, the NVIDIA logo, ConnectX, and Rivermax are trademarks and/or registered trademarks of NVIDIA Corporation and its affiliates 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

© 2021 NVIDIA Corporation & Affiliates. All rights reserved.