



**GLOBAL LEADER IN  
BROADCAST QUALITY  
LIVE VIDEO OVER IP**

# **Zixi Transcoder Installation Guide**

Software Version 17  
Document Version DOC23-340-0018  
Copyright © Zixi 2024  
All Rights Reserved

## Legal Notice

This document contains proprietary and confidential material of Zixi LLC. Any unauthorized reproduction, use, or disclosure of this material, or any part thereof, is strictly prohibited. This document is solely for the use of Zixi employees and authorized Zixi customers.

The material furnished in this document is believed to be accurate and reliable. However, no responsibility is assumed by Zixi LLC. for the use of this document or any material included herein. Zixi LLC. reserves the right to make changes to this document or any material included herein at any time and without notice.

For more information visit: [www.zixi.com](http://www.zixi.com).

Copyright © Zixi 2024 Waltham, MA U.S.A.

All Rights Reserved.

*\* See back for additional licensing information*

# Table of Contents

Introduction .....	4
Software-based Transcoding Features .....	7
NVIDIA GPU-based Transcoding Features .....	8
Software-based Transcoder Installation .....	9
Hardware Requirements.....	9
Operating System Requirements.....	9
License Option Verification.....	9
NVIDIA-based Transcoder Installation.....	12
Hardware Requirements.....	12
Operating System Requirements.....	12
Installation .....	13
Broadcaster V16.....	13
Broadcaster V17.....	16
Installer Screens and Prompts .....	18
Validating the NVIDIA Driver Installation .....	21

# Introduction

Zixi Broadcaster's live transcoding offers a single solution for the universal origination and distribution of high quality, adaptive live broadcast content to OTT workflows, digital MVPDs, MSOs, hotels, cinemas, iPhone and Android player apps as well as other destinations. The Zixi Transcoder supports H.265/HEVC, H.264/AVC, and MPEG2, and Raw video codecs as well as a variety of audio codecs (e.g. AAC, OPUS, Raw) most commonly used for live broadcast and OTT workflows. For OTT and Social Media/eSports, the transcoder can repackage a live stream to HLS, DASH, and RTMP, backhaul via Zixi protocol, and output MPEG-TS, RTP, UDP, NDI, and WebRTC for reintegration into broadcast systems.

With the Zixi transcoder you can:

- Deliver to any device anywhere in different bitrates and protocols simultaneously
- Repackage streams to a variety of formats and protocols including MPEG 2, H.264, HEVC encode and decode
- Synchronize GOPs for multi-bitrate delivery

The transcoder typically employed in the following cases:

- Where a target device does not support the source format.
- Where a target has limited storage capacity that requires a reduced file size.
- To convert incompatible obsolete data to a better-supported or modern format.
- You can add video (H.264, H.265, MPEG2, Raw) and audio (AAC, OPUS, Raw) profiles.

### Typical Workflows

Typical workflows for the Zixi transcoder include:

- **Broadcast backhaul** – Compress high bitrate live mezzanine MPEG2 to lower bitrate H264 or H265, at the same quality level, to reduce bandwidth requirements for video uplink. When uplink bandwidth is scarce or cost is at a premium, compression of live content can make possible an otherwise impractical workflow.
- **Adaptive Bitrate (ABR)** – Deliver live mezzanine video transcoded into multiple renditions of H264 or H265 and packaged into HLS or DASH formats to laptops, mobile devices and set top boxes via Content Delivery Networks, YouTube Live or similar services. Adaptive streams can also be delivered via the Zixi protocol to Zixi enabled endpoints.
- **Bitrate reduction and packaging for Social Media** – Deliver live mezzanine video transcoded to lower bitrate H264 and packaged into RTMP or HLS to Social Media sites such as Facebook Live, YouTube Live, and Twitch.

## Introduction

---

### Software-based Transcoding Features

#### Transcoding Options

The transcoding feature in the Zixi Broadcaster is available using the following transcoding options:

- **Software-based codec** - uses the AVCODEC library for video decode and x264 library for video encode.
- **NVIDIA GPU-based codec** – used nvdec and nvenc libraries and is available in PCIe boards for bare-metal installations and on cloud providers such as AWS, Google Cloud Platform, and Azure.

#### Ad Insertion Options

Zixi's transcoder supports the following ad insertion options:

- SCTE-35 passthrough to Zixi TS
- SCTE-35 converted to CUE-OUT/CUE-IN in HLS
- SCTE-35 converted to Splice Events in DASH
- SCTE-35 converted to OnCuePoint in RTMP
- API Injection with output to SCTE-35 in Zixi TS
- API Injection with output to CUE-OUT/CUE-IN in HLS
- API Injection with output to Splice Events in DASH
- API Injection with output to OnCuePoint in RTMP

#### Captioning Options

Zixi's transcoder supports the following captioning options:

- ATSC 608/708 ES passthrough to Zixi TS, HLS, DASH, and RTMP
- DVB track passthrough to Zixi TS
- WebVTT generation from ATSC 708 Elementary Stream

#### Audio Processing Options

- Transcode up to 16 audio tracks
- Passthrough of multiple audio tracks
- Passthrough of unsupported codecs such as E-AC-3
- Audio Sample Rate Conversion - 44.1KHz to/from 48KHz
- PID mapping of audio tracks

### Audio Decode

- AAC Low Complexity, High Efficiency V1, High Efficiency V2
- AC3 (Dolby Digital)
- MP2 (MPEG-2 Audio Layer II)
- MP1 (MPEG-1 Audio Layer II)
- Opus

### Audio Encode

- AAC Low Complexity
- AAC High Efficiency V1
- AAC High Efficiency V2
- E-AC-3 (Enhanced Dolby Digital)
- MP1 (MPEG-1 Audio Layer II)
- Opus

### Video Processing

- MPTS De-multiplexing
- Passthrough
- 10-bit to 8-bit color downscaling
- Thumbnail extraction as HLS playlist
- Deinterlacing (except telecine 60i/59.94i to 24p)
- Interlaced and progressive output with software
- Progressive output with NVIDIA
- Smooth up/down framerate conversion with frame interpolation with NVIDIA
- Frame duplication/removal framerate conversion with software
- PID mapping and PID pass-thru
- Preserve aspect ratio with black bar insertion
- Cropping
- Static image overlay/logo Insertion
- Dynamic HTML overlay
- Copy GOP from source or generate new GOP
- GOP synchronization when generate new GOP

## Introduction

---

### Software-based Transcoding Features

- Omit or keep closed captions
- I-frame synchronization of outputs with different FPS
- I-frame injection at SCTE-35 markers
- Preserve all video source color primaries

## Software-based Transcoding Features

Software-based transcoding will operate on any hardware configuration supported by the Zixi Broadcaster, including on-premise hardware and cloud-based hardware and on X86-64 and ARM64 processors, but excluding currently excluding MacOS. Although software-based transcoding offers lower density than NVIDIA GPU based transcoding, it is the most flexible from a hardware compatibility perspective.

Software-based transcoding offers the following features:

	<i>Format</i>	<i>Chroma Subsampling</i>	<i>Color Depth</i>
<i>Video Decode</i>	MPEG-2	4:2:0	8-bit
	H.264/AVC	4:2:0, 4:2:2, 4:4:4	8-bit, 10-bit
	H.265/HEVC	4:2:0, 4:2:2, 4:4:4	8-bit, 10-bit
<i>Video Encode</i>	MPEG-2	4:2:0	8-bit
	H.264/AVC	4:2:0, 4:2:2, 4:4:4	8-bit, 10-bit*

\* Input stream must be 10-bit

- **Video Decoding:**
  - Frame Size: SD, HD (720p), Full HD (1080p), 4K (2160p)
  - Frame Rate: 24fps, 25fps, 29.97fps, 30fps, 59.94fps, 60fps
- **Video Encoding:**
  - Frame Size: SD, HD (720p), Full HD (1080p)
  - Frame Rate: 24fps, 25fps, 29.97fps, 30fps, 59.94fps, 60fps

# NVIDIA GPU-based Transcoding Features

NVIDIA GPU transcoding uses NVIDIA GPUs for hardware accelerated video transcoding. Servers with NVIDIA GPUs can be on-premise or cloud-based. NVIDIA GPU based transcoding offers the following features:

	<i>Format</i>	<i>Chroma Subsampling</i>	<i>Color Depth</i>
<i>Video Decode</i>	MPEG-2	4:2:0	8-bit
	H.264/AVC	4:2:0	8-bit
	H.265/HEVC	4:2:0, 4:4:4	8-bit, 10-bit
<i>Video Encode</i>	H.264/AVC	4:2:0, 4:4:4	8-bit
	H.265/HEVC	4:2:0, 4:4:4	8-bit, 10-bit*

\*Input stream must be 10-bit

- **Video Decoding:**
  - Frame Size: SD, HD (720p), Full HD (1080p), 4K (2160p)
  - Frame Rate: 24fps, 25fps, 29.97fps, 30fps, 59.94fps, 60fps
- **Video Encoding:**
  - Frame Size: SD, HD (720p), Full HD (1080p), 4K (2160p)
  - Frame Rate: 24fps, 25fps, 29.97fps, 30fps, 59.94fps, 60fps



# Software-based Transcoder Installation

Software based transcoding uses AVCODEC and x264 and will operate on any hardware configuration supported by the Zixi Broadcaster, including on-premise hardware and cloud-based hardware and on X86-64 and ARM64 processors, but excluding MacOS.



To enable transcoding, you will need a Zixi transcoding license. To obtain a license file, please contact Zixi support - [support@zixi.com](mailto:support@zixi.com).

## Hardware Requirements

The software-based codec option doesn't have special hardware-requirements, however more powerful CPUs will support more transcoding channels with more profiles. To learn more about Zixi Broadcaster's hardware requirements, refer to the **Zixi Installation Guide**.

## Operating System Requirements

The operating system requirements are listed in the **Zixi Installation Guide**.

## License Option Verification

Software transcoding does not require the installation of special hardware. The x264 codec needs to be enabled in the license provided by Zixi. To verify if the x264 license option is enabled, follow the instructions below.

### ➔ To verify that the x264 license option is enabled in V16:

1. In the **Status** page of the Zixi Broadcaster, under **License Information**, click on the message *"Click here for full license details"* in V16.

The screenshot shows the Zixi Broadcaster interface. The top navigation bar includes STATUS, INPUTS, OUTPUTS, ADAPTIVE, VOD, FILES, TRANSCODER, SETTINGS, EVENT LOG, MATRIX, and LOGOUT. The main content area is divided into two sections: General and License Information.

**General**

- ID: QA-Alon-Tech-Writer-mwn4
- Host ID: 06816b45f320
- Start Time: 2020-Aug-27 10:32:36
- Up Time: 507:54:10
- Inputs: 1
- Outputs: 0
- Remote In [kbps]: 0
- Local In [kbps]: 0
- Remote Out [kbps]: 0
- Local Out [kbps]: 0
- HTTP Out [kbps]: 0
- Kilobytes Sent: 1012114571
- Kilobytes Received: 1394047305
- Kilobytes Transcoded: 674493
- RTMP Server: Active

**License Information**

Zixi License Status			
Output GB	9999900 [GB]	Used	3512202 [GB]
Expiration	2021-Mar-16		
<a href="#">Click here for full license details.</a>			

The Zixi License Status window opens.

The screenshot shows the Zixi License Status window with four main sections:

- Application:** A table with columns Product and Key. The row shows Product: Broadcaster\_11 and Key: demo-0125-efcd-b09e.
- Meters:** A table with columns Meter and Expiration. The row shows Meter: Output GB 3512262 [GB] / 9999900 [GB] and Expiration: 16-Mar-2021. Below this, it says Expires 2021-Mar-16.
- Limits:** A table with columns Input and Output. The row for Transcoders shows Input: 999 and Output: 999.
- Features:** A list of features with a Supported checkbox. The feature x264 Transcoding is checked.

2. Under **Limits**, verify that the number of transcoders is greater than 0.
3. Under **Features**, verify that the **x264 Transcoding** option is enabled.

➔ To verify that the x264 license option is enabled in V17:

1. Click on “Full License Details”.

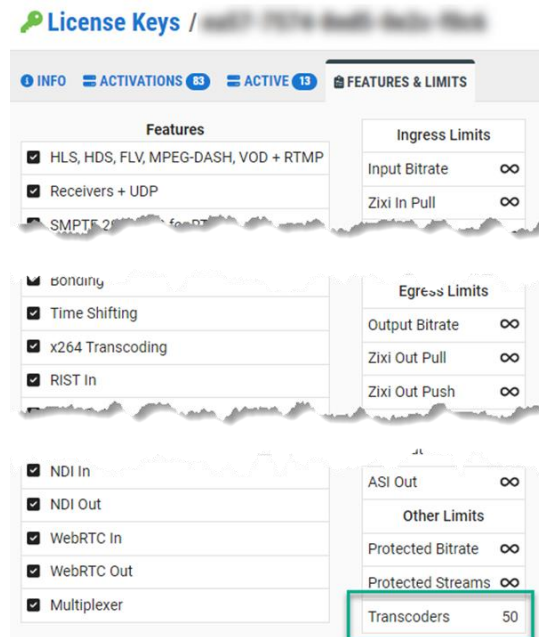
The screenshot shows the License Information section of the Zixi License Status window. It includes a link for Full License Details (requires account) and a table with the following data:

Meter Type	No Reset
All Outputs: all egress	Used: 10.5 TB : Max: 9.0 ZB : 0.0%

## Software-based Transcoder Installation

### License Option Verification

The **Zixi License Status** window opens.



The screenshot shows the Zixi License Status window with the following sections:

- License Keys /** (with a blurred key ID)
- INFO** | **ACTIVATIONS 83** | **ACTIVE 19** | **FEATURES & LIMITS**
- Features**
  - HLS, HDS, FLV, MPEG-DASH, VOD + RTMP
  - Receivers + UDP
  - SMPTC 2K (for RTMP)
  - Bonding
  - Time Shifting
  - x264 Transcoding
  - RIST In
  - NDI In
  - NDI Out
  - WebRTC In
  - WebRTC Out
  - Multiplexer
- Ingress Limits**
  - Input Bitrate: ∞
  - Zixi In Pull: ∞
- Egress Limits**
  - Output Bitrate: ∞
  - Zixi Out Pull: ∞
  - Zixi Out Push: ∞
- Other Limits**
  - ASI Out: ∞
  - Protected Bitrate: ∞
  - Protected Streams: ∞
  - Transcoders: 50** (highlighted with a red box)

2. Under **Limits**, verify that the number of transcoders is greater than 0.
3. Under **Features**, verify that the **x264 Transcoding** option is enabled.

# NVIDIA-based Transcoder Installation

NVIDIA GPU transcoding uses NVIDIA GPUs for hardware accelerated video transcoding and will operate on any hardware configuration supported by the Zixi Broadcaster, including on-premise hardware and cloud-based hardware and on X86-64 and ARM64 processors.



To enable transcoding, you will need a Zixi transcoding license. To obtain a license file, please contact Zixi support - [support@zixi.com](mailto:support@zixi.com).

## Hardware Requirements

NVIDIA GPUs are available on consumer gaming graphics PCIe cards, professional CAD acceleration PCIe cards, custom motherboards, and on cloud services such as AWS, Azure, and Google Cloud.

Below are the recommended modern NVIDIA GPUs, although older and new GPUs should work as well:

- NVIDIA Quadro & RTX PCIe Cards
- AWS VM - g4dn (T4)
- GCP VM – G2 (L4), T4
- Azure VM – NCas (T4)
- Tata Cloud VM – A16, T4

NVIDIA Pascal architectures or later are required for HEVC/H.265 decoding.

NVIDIA Tesla architectures cannot decode HEVC/H.265, but can decode MPEG2 and H.264. As of this writing, NVIDIA does not support decode or encode of 4:2:2 content. More details on specific features supported by different architectures can be found in the encode/decode support matrix from NVIDIA at:

<https://developer.nvidia.com/video-encode-decode-gpu-support-matrix>

Key information to look for when selecting a GPU card and architecture is the “Max # of concurrent sessions”. This value determines how many encodes can be performed in parallel, which is relevant for ABR streaming where you will typically need between three to nine bitrate profiles. The best choice for transcoding for ABR streaming is a GPU card with unrestricted sessions.

## Operating System Requirements

The operating system requirements are listed in the **Zixi Installation Guide**.

## Installation

This section focuses on installing NVIDIA drivers necessary for transcoding on bare metal servers and virtual machines. For AWS, GCP, and Azure, it is possible to use ZEN Master to launch Broadcasters on instances with NVIDIA GPUs. Alternatively, you can manually install NVIDIA drivers following the examples below. Install NVIDIA drivers as described below.

### Broadcaster V16

Broadcaster V16 requires CUDA 11 and the driver version to be  $\geq 470$  and  $\leq 515$ . You can search for NVIDIA driver downloads at <https://www.nvidia.com/Download/Find.aspx?lang=en-us> or simply use the version in the example below.

#### ➔ CentOS 7.9 on GCP, Azure or any cloud or bare metal or Amazon Linux 2:

1. Update the OS and install development tools:

```
# yum -y upgrade
# yum -y groupinstall "Development Tools"
# yum -y install kernel-devel
```

2. Reboot to load the latest kernel from the upgrade.
3. Check to see if 'nouveau' module is installed.

```
# lsmod | grep -i nouveau
```

4. If 'nouveau' is installed follow steps below to remove it.

```
# vi /etc/default/grub
```

- Append the following line to the line starting with "GRUB\_CMDLINE\_LINUX":

```
rd.driver.blacklist=nouveau nouveau.modset=0
```

- Update the grub configuration:

```
# grub2-mkconfig -o /boot/grub2/grub.cfg
```

- Add 'nouveau' to the module black list:

```
# vi /etc/modprobe.d/local-blacklist.conf
```

- Append the following line:

```
blacklist nouveau
```

- Rebuild initramfs.



The /boot partition or folder needs to have enough space for the updated initramfs file.

```
# cp /boot/initramfs-$(uname -r).img /root/initramfs-$(uname
-r).img.before_update.bak
# dracut -v -f
```

- Reboot the server and check that the 'nouveau' module is not installed.

```
# lsmod | grep -i nouveau
```

### 5. Retrieve and launch the correct installer for the architecture:

- For X86 64-bit machines, install the following NVIDIA Linux driver:

```
# wget http://us.download.nvidia.com/XFree86/Linux-x86_64/470.86/NVIDIA-
Linux-x86_64-470.86.run
# chmod 777 NVIDIA-Linux-x86_64-470.86.run
# ./NVIDIA-Linux-x86_64-470.86.run
```

- On an ARM based machine with NVIDIA, download the following driver:

```
wget https://us.download.nvidia.com/XFree86/aarch64/470.141.03/NVIDIA-
Linux-aarch64-470.141.03.run
chmod 777 NVIDIA-Linux-aarch64-470.141.03.run
./NVIDIA-Linux-aarch64-470.141.03.run
```

### ➔ Amazon Linux 2023 (AMI al2023-ami-2023.3.20240108.0-kernel-6.1-x86\_64):

#### 1. Update the OS and install development tools:

```
# dnf -y upgrade
# dnf -y groupinstall "Development Tools"
# dnf -y kernel-modules-extra
```

#### 2. Reboot to load the latest kernel from the upgrade.

#### 3. Retrieve and launch the correct installer for the architecture:

- For X86 64-bit machines, install the following NVIDIA Linux driver:

## NVIDIA-based Transcoder Installation

---

### Installation

```
# wget http://us.download.nvidia.com/XFree86/Linux-x86_64/515.105.01/NVIDIA-Linux-x86_64-515.105.01.run
# chmod 777 NVIDIA-Linux-x86_64-515.105.01.run
# ./NVIDIA-Linux-x86_64-515.105.01.run
```

- On an ARM based machine with NVIDIA, download the following driver:

```
# wget https://us.download.nvidia.com/XFree86/aarch64/515.105.01/NVIDIA-Linux-aarch64-515.105.01.run
# chmod 777 NVIDIA-Linux-aarch64-515.105.01.run
# ./NVIDIA-Linux-aarch64-515.105.01.run
```

#### ➔ Rocky Linux 9 Optimized for GCP (x86/64, x86\_64 built on 20240111):

1. Update the OS and install development tools:

```
# dnf -y upgrade
# dnf -y groupinstall "Development Tools"
# dnf -y install vulkan-loader libglvnd-devel wget
```

2. Reboot to load the latest kernel from the upgrade.
3. Retrieve and launch the installer:

```
# wget http://us.download.nvidia.com/XFree86/Linux-x86_64/515.105.01/NVIDIA-Linux-x86_64-515.105.01.run
# chmod 777 NVIDIA-Linux-x86_64-515.105.01.run
# ./NVIDIA-Linux-x86_64-515.105.01.run
```

#### ➔ Rocky Linux 9 – Free – x64 Gen2 on Azure (official image from Rocky Enterprise Software Foundation):

1. Update the OS and install development tools:

```
# dnf -y upgrade
# dnf -y groupinstall "Development Tools"
# dnf -y install vulkan-loader libglvnd-devel wget
```

2. Reboot to load the latest kernel from the upgrade.
3. Retrieve and launch the installer:

## NVIDIA-based Transcoder Installation

```
# wget http://us.download.nvidia.com/XFree86/Linux-x86_64/515.105.01/NVIDIA-Linux-x86_64-515.105.01.run
# chmod 777 NVIDIA-Linux-x86_64-515.105.01.run
# ./NVIDIA-Linux-x86_64-515.105.01.run
```

### ➔ To upgrade to a new version of the NVIDIA driver:

1. Using the existing installer or any installer version use the --uninstall option.

- For X86 64-bit machines

```
# ./NVIDIA-Linux-x86_64-xxx.xxx.xx.run --uninstall
```

- On an ARM based machine with NVIDIA

```
# ./NVIDIA-Linux-aarch64-xxx.xxx.xx.run --uninstall
```

- Then reboot to load the kernel with the NVIDIA driver removed
2. Retrieve and install the new version as described above.

## Broadcaster V17

Broadcaster V17 requires a CUDA 12.1 and the driver version to be  $\geq 535$ . If you are upgrading from V16 to V17 you will have an older NVIDIA driver and will see an error like this on the Broadcaster Status page:

### OS Information

```
OS: Amazon Linux release 2023 (Amazon Linux)
Build: 6.1.66-93.164.amzn2023.x86_64
Architecture: x86_64
Nvidia Driver: 515.105.01
Cuda Version: 11.7 - Cuda version not up to date - minimum required version: 12.1
```

and an error like the following on the Inputs page:

Transcoded sources						
Status	ID	Bitrate[kbps]	Up Time	TR 101 290	Error	Outputs
Offline	N_H264_2997_0234p1_hp_1	0	00:00:05	Off	Cuda not up to date	0

You can search for NVIDIA driver downloads at

<https://www.nvidia.com/Download/Find.aspx?lang=en-us> or simply use the version in the example below.



## NVIDIA-based Transcoder Installation

---

### Installation

#### ➔ Amazon Linux 2023 (AMI al2023-ami-2023.3.20240108.0-kernel-6.1-x86\_64):

1. Update the OS and install development tools:

```
# dnf -y upgrade
# dnf -y groupinstall "Development Tools"
# dnf -y install kernel-modules-extra
```

2. Retrieve and launch the correct installer for the architecture:

- For X86 64-bit machines, install the following NVIDIA Linux driver:

```
# wget http://us.download.nvidia.com/XFree86/Linux-
x86_64/535.129.03/NVIDIA-Linux-x86_64-535.129.03.run
# chmod 777 NVIDIA-Linux-x86_64-535.129.03.run
# ./NVIDIA-Linux-x86_64-535.129.03.run
```

- On an ARM based machine with NVIDIA, download the following driver:

```
# wget https://us.download.nvidia.com/XFree86/aarch64/535.129.03/NVIDIA-
Linux-aarch64-535.129.03.run
# chmod 777 NVIDIA-Linux-aarch64-535.129.03.run
# ./NVIDIA-Linux-aarch64-535.129.03.run
```

#### ➔ Rocky Linux 9 Optimized for GCP (x86/64, x86\_64 built on 20240111):

1. Update the OS and install development tools:

```
# dnf -y upgrade
# dnf -y groupinstall "Development Tools"
# dnf -y install vulkan-loader libglvnd-devel wget
```

2. Reboot to load the latest kernel from the upgrade.
3. Retrieve and launch the installer:

```
# wget http://us.download.nvidia.com/XFree86/Linux-
x86_64/535.129.03/NVIDIA-Linux-x86_64-535.129.03.run
# chmod 777 NVIDIA-Linux-x86_64-535.129.03.run
# ./NVIDIA-Linux-x86_64-535.129.03.run
```

### ➔ Rocky Linux 9 – Free – x64 Gen2 on Azure (official image from Rocky Enterprise Software Foundation):

1. Update the OS and install development tools:

```
# dnf -y upgrade
# dnf -y groupinstall "Development Tools"
# dnf -y install vulkan-loader libglvnd-devel wget
```

2. Reboot to load the latest kernel from the upgrade.
3. Retrieve and launch the installer:

```
# wget http://us.download.nvidia.com/XFree86/Linux-
x86_64/535.129.03/NVIDIA-Linux-x86_64-535.129.03.run
# chmod 777 NVIDIA-Linux-x86_64-535.129.03.run
# ./NVIDIA-Linux-x86_64-535.129.03.run
```

### ➔ To upgrade to a new version of the NVIDIA driver:

1. Using the existing installer or any installer version use the --uninstall option.
  - For X86 64-bit machines

```
# ./NVIDIA-Linux-x86_64-xxx.xxx.xx.run --uninstall
```

- On an ARM based machine with NVIDIA

```
# ./NVIDIA-Linux-aarch64-xxx.xxx.xx.run --uninstall
```

- Then reboot to load the kernel with the NVIDIA driver removed.
2. Retrieve and install the new version as described above.

## Installer Screens and Prompts

After launching the NVIDIA installer, you will see various messages. Below are explanations of the more common messages and what action you might need to take.

If you see the following error, you might have upgraded the kernel but have not yet rebooted so are still running a previous version of the kernel. Select OK [ENTER] and exit the installer, reboot, and try again.

## NVIDIA-based Transcoder Installation

### Installer Screens and Prompts

```
ERROR: Unable to find the kernel source tree for the currently running kernel. Please make sure you have installed the kernel source files for your kernel and that they are properly configured; on Red Hat Linux systems, for example, be sure you have the 'kernel-source' or 'kernel-devel' RPM installed. If you know the correct kernel source files are installed, you may specify the kernel source path with the '--kernel-source-path' command line option.
```

OK

If you see the following error on Amazon Linux 2023 you are most likely missing the kernel-modules-extra package. Select OK [ENTER] and exit the installer, install the package, and try again.

```
ERROR: Unable to load the kernel module 'nvidia.ko'. This happens most frequently when this kernel module was built against the wrong or improperly configured kernel sources, with a version of gcc that differs from the one used to build the target kernel, or if another driver, such as nouveau, is present and prevents the NVIDIA kernel module from obtaining ownership of the NVIDIA device(s), or no NVIDIA device installed in this system is supported by this NVIDIA Linux graphics driver release.
```

```
Please see the log entries 'Kernel module load error' and 'Kernel messages' at the end of the file '/var/log/nvidia-installer.log' for more information.
```

OK

If you see the following error you likely need to install the kernel-devel package. Select OK [ENTER] and exit the installer, install the package, and try again.

```
ERROR: Unable to find the development tool `cc` in your path; please make sure that you have the package 'gcc' installed. If gcc is installed on your system, then please check that `cc` is in your PATH.
```

OK

The following message is related to the X display and is not relevant to the Broadcaster use of NVIDIA GPUs. Select OK [ENTER] and keep going.

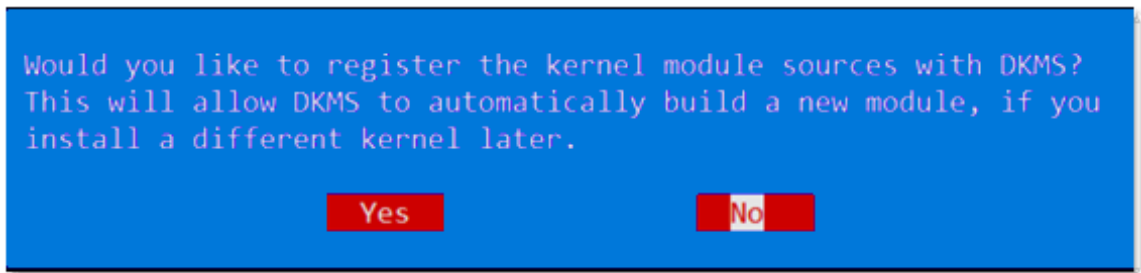
```
WARNING: nvidia-installer was forced to guess the X library path '/usr/lib64' and X module path '/usr/lib64/xorg/modules'; these paths were not queryable from the system. If X fails to find the NVIDIA X driver module, please install the `pkg-config` utility and the X.Org SDK/development package for your distribution and reinstall the driver.
```

OK

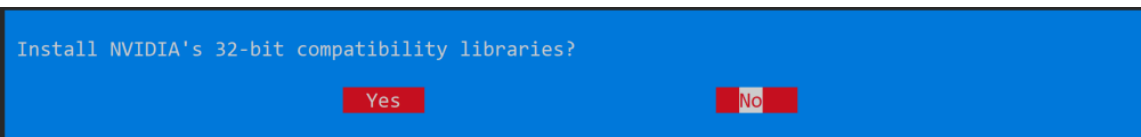
The following message is not relevant to the Broadcaster use of NVIDIA GPUs. Select No [ENTER] and keep going.

## NVIDIA-based Transcoder Installation

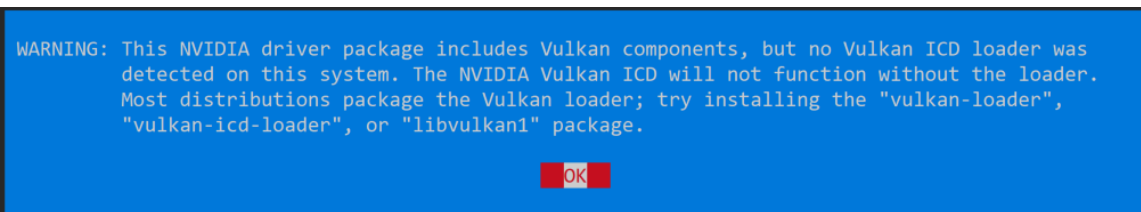
---



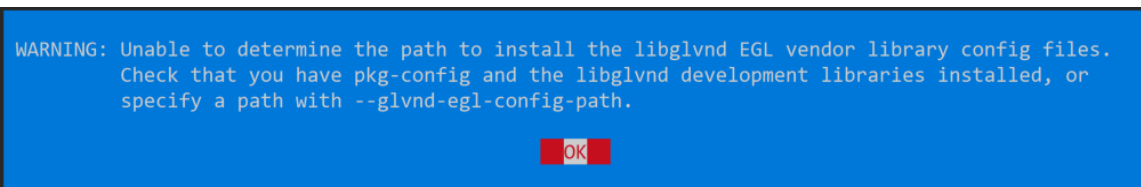
The following message is not relevant to the Broadcaster use of NVIDIA GPUs. Select No [ENTER] and keep going.



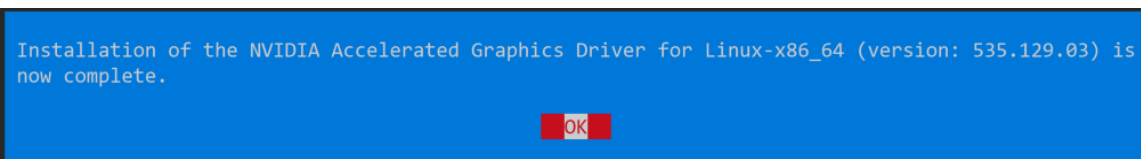
The following message is due to a missing vulkan-loader package, but is not relevant to the Broadcaster use of NVIDIA GPUs. Select OK [ENTER] and keep going.



The following message is due to a missing libglvnd-devel package, but is not relevant to the Broadcaster use of NVIDIA GPUs. Select OK [ENTER] and keep going.



This final message confirms the installation is complete.



## Validating the NVIDIA Driver Installation

Run the following command to verify that the NVIDIA drivers have been installed:

```
# nvidia-smi
```

The output will look something like the following:

```
[root@localhost ~]# nvidia-smi
Wed Dec 1 15:12:20 2021

+-----+
| NVIDIA-SMI 470.86          Driver Version: 470.86          CUDA Version: 11.4          |
+-----+-----+-----+-----+-----+-----+
| GPU  Name           Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf    Pwr:Usage/Cap|      Memory-Usage | GPU-Util  Compute M. |
|====+=====+====+=====+=====+=====+
|  0   Quadro P2000     Off          | 00000000:01:00:0 | Off      N/A   |
| 97%   55C    P0      20W / 75W |  0MiB / 5058MiB |    3%    Default |
|                               |                      |          N/A   |
+-----+-----+-----+-----+-----+-----+
|
| Processes:
| GPU   GI    CI          PID   Type   Process name          GPU Memory
|      ID    ID                             |                  |           Usage
|-----+-----+-----+-----+-----+-----+
| No running processes found
+-----+-----+-----+-----+-----+-----+

```

If the `nvidia-smi` command does produce a result like shown above then the NVIDIA driver has not been successfully installed and you will need to repeat the installation process and try to determine where it failed.

If the `nvidia-smi` command was successful, open the Zixi Broadcaster UI in your web browser and on the Status page find the Machine Information and OS Information sections. You should see an NVIDIA GPU listed and a driver and CUDA version as shown below:

## NVIDIA-based Transcoder Installation

---

### Machine Information

System Model: Alienware Alienware X51  
Processor Model: Intel(R) Core(TM) i7-3770 CPU @ 3.40GHz  
Processors: 1  
Logical Cores: 8  
Installed RAM: 15 GB  
GPU(s): Quadro P2000 [ Quadro ] [ Pascal ]

### OS Information

OS: CentOS Linux release 7.9.2009 (Core)  
Build: 3.10.0-1160.45.1.el7.x86\_64  
Architecture: x86\_64  
Nvidia Driver: 470.86  
Cuda Version: 11.4

In the License Information section on the status page click the available link to verify that your license has transcoding enabled. Broadcaster V16 looks like the image below.

# NVIDIA-based Transcoder Installation

## Validating the NVIDIA Driver Installation

The screenshot displays the Zixi Transcoder web interface. At the top, there is a navigation bar with tabs: STATUS, INPUTS, OUTPUTS, ADAPTIVE, VOD, FILES, TRANSCODER, SETTINGS, EVENT LOG, MATRIX, and LOGOUT. Below the navigation bar, there are buttons for Network Test, Options, and Restart. The main content area is divided into several sections:

- License Information:** Contains a "Zixi License Status" box with the following data:

Output GB/Month	500 [GB]	Used	4 [GB]
Expiration	2019-Oct-19		

A link "Click here for full license details." is highlighted with a green box.
- Application:** A table with columns Product and Key.

Product	Key
Broadcaster_11	demo-33a4-ec29-6e94
- Meters:** A table with columns Meter and Expiration.

Meter	Expiration
Output GB/Month	500 [GB] 19-Oct-2019
Available	500 [GB] Used 382 [GB]
Expires	2019-Oct-19
- Limits:** A table with columns Input and Output.

	Input	Output
Bitrate	Unlimited	Unlimited
Zixi Pull	Unlimited	Unlimited
Zixi Push	Unlimited	Unlimited
RTMP Pull	Unlimited	Unlimited
RTMP Push	Unlimited	Unlimited
UDP	Unlimited	Unlimited
ASI	Unlimited	Unlimited
Transcoders	100	

The "Transcoders" row is highlighted with a green box.
- Features:** A list of features with checkboxes indicating support.

	Supported
HLS, HDS, FLV, MPEG-DASH, VOD & RTMP	<input checked="" type="checkbox"/>
Receivers & UDP	<input checked="" type="checkbox"/>
SMPTE 2022 FEC for RTP	<input checked="" type="checkbox"/>
TR101 Stream analysis	<input checked="" type="checkbox"/>
File transfer & Zixi accelerated HLS/HDS	<input checked="" type="checkbox"/>
Cluster	<input checked="" type="checkbox"/>
Matrix	<input checked="" type="checkbox"/>
Bonding	<input checked="" type="checkbox"/>
Time shifting	<input checked="" type="checkbox"/>
x264 Transcoding	<input checked="" type="checkbox"/>

For Broadcaster V17 the License Information section looks like the following and the link will take you to the Zixi customer portal where you can see your license details under the "FEATURES & LIMITS" tab.

The screenshot shows the "License Information" section for Broadcaster V17. It features a "Zixi License Status" box with a link "Full License Details (requires account)." highlighted with a green box. Below the link, there is a table with columns Meter Type and No Reset.

Meter Type	No Reset
All Outputs: all egress	Used: 10.5 TB : Max: 9.0 ZB : 0.0%

# NVIDIA-based Transcoder Installation

The screenshot shows the 'License Keys' management interface. At the top, there are navigation tabs: 'INFO', 'ACTIVATIONS 83', 'ACTIVE 13', and 'FEATURES & LIMITS'. The 'FEATURES & LIMITS' tab is selected. The interface is divided into two main sections: 'Features' and 'Limits'.

**Features:** A list of features with checkboxes, all of which are checked:

- HLS, HDS, FLV, MPEG-DASH, VOD + RTMP
- Receivers + UDP
- SMPTE 2022 FEC for RTP
- TR101 Stream analysis
- SRT Out
- NDI In
- NDI Out
- WebRTC In
- WebRTC Out
- Multiplexer

**Limits:** A table of limits with values:

Ingress Limits	
Input Bitrate	∞
Zixi In Pull	∞
Zixi In Push	∞
RTMP In Pull	∞
Other Limits	
UDP Out	∞
ASI Out	∞
Protected Bitrate	∞
Protected Streams	∞
Transcoders	50

The 'Transcoders' row in the 'Other Limits' section is highlighted with a red box.



# NVIDIA-based Transcoder Installation

## Validating the NVIDIA Driver Installation

-----  
- License for OpenSSL  
-----

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit

(<http://www.openssl.org/>)

THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This product includes cryptographic software written by Eric Young (eay@cryptsoft.com). This product includes software written by Tim

Hudson (tjh@cryptsoft.com).

Copyright (c) 1998-2017 The OpenSSL Project. All rights reserved.

-----  
- License for Libssh2  
-----

Copyright (c) 2004-2007 Sara Golemon <sarag@libssh2.org>

Copyright (c) 2005,2006 Mikhail Gusarov <dottedmag@dottedmag.net>

Copyright (c) 2006-2007 The Written Word, Inc.

Copyright (c) 2007 Eli Fant <elifantu@mail.ru>

Copyright (c) 2009-2014 Daniel Stenberg

Copyright (C) 2008, 2009 Simon Josefsson

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of the copyright holder nor the names of any other contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

-----  
- License for DekTec's Linux SDK. Version 1, August 2012  
-----

Copyright (C) 2000-2015 DekTec Digital Video B.V.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary format must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation.

THIS SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL DEKTEC DIGITAL VIDEO BV, ITS AGENTS OR ITS EMPLOYEES BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, INCIDENTAL, OR OTHER DAMAGES (INCLUDING DAMAGES FOR THE LOSS OF USE, INFORMATION, GOODWILL, PROFIT, WORK STOPPAGE, DATA, BUSINESS OR REVENUE) UNDER ANY CIRCUMSTANCES, OR UNDER ANY LEGAL THEORY, WHETHER IN CONTRACT, IN TORT, IN NEGLIGENCE, OR OTHERWISE, ARISING FROM THE USE OF, OR INABILITY TO USE THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.