# APPS024-Codensity™ T408 Windows MSYS2 Setup Application Note

v2.0

# Table of Contents

# 1 Table of Abbreviations

| **Abbreviation:** | **Full Form:** |
|---|---|
| AI | Artificial Intelligence |
| API | Application Programming Interface |
| AUD<br>AV | Access Unit Delimiters<br>Audio Video |
| AV1 | Alliance Open Media Video 1 Codec |
| CABAC | Context Adaptive Binary Arithmetic Coding |
| CBR | Constant Bit Rate |
| CPU | Central Processing Unit |
| CRF | Constant Rate Factor |
| DSP | Digital Signal Processing |
| EOF | End Of File |
| FIXQB | Fixed Quantization Parameter |
| FF | Form Factor |
| FPS | Frames Per Second |
| GOP | Group of pictures |
| HDR | High Dynamic Range |
| HHHL | Half Height, Half Length |
| HHFL | Half Height, Full Length |
| HLG | Hybrid Log Gamma |
| HRD | Hypothetical Reference Decoder |
| IE | Inference Engine |
| Mbps | Mega Bit per second |
| MBps | Mega Byte per second |
| NLP | Natural Language Processing |
| NPU | Neural Processing Unit |
| NVMe | Non-Volatile Memory express |
| PCIe | Peripheral Component Interconnect express |
| PPS | Picture Parameter Set |
| QP | Quoted Printable |
| RGB | Red Green Blue |
| RGBA | Red Green Blue Alpha |
| RDO | Optimization |
| ROI | Region of interest |
| SDK | Software Development Kit |
| SEI | Supplemental Enhancement Information |

| | |
|---|---|
| SRIOV | Single Root I/O Virtualization |
| SPS | Sequence Parameter Set |
| VBR | Variable Bit Rat |
| VCL | Video Coding Layer |
| VPS | Video Parameter Set |
| VUI | Video Usability Information |

# 2 References

Technical Notes:

Specifications:

# 3 Background

This document describes the Windows MSYS2 environment setup and usage.

## 3.1 Intended Audience

This document is intended to help engineers/technicians/developers operating NETINT transcoders on a windows environment.

## 3.2 Compatibility

**Software Compatibility**

This guide is intended to be used with NETINT Codensity T408 Video Transcoder software Release 2.5.0 or newer.

**Hardware Compatibility**

Release 2.5.0 supports NETINT Codensity T408 Video Transcoder hardware.

**Operation system**

Windows operating system.

## 3.3 Operating Systems

A host server with one of the following operating systems installed is recommended (minor versions can be inconsistent):

- OS: Windows 10 [Version 10.0.19042.1052]
- OS: Windows Server 2012 R2 [Version 6.3.9600]
- OS: Windows Server 2016 [Version 10.0.14393.693]
- OS: Windows Server 2019 [Version 10.0.17763.737]

# 4 Setup MSYS2 tools on Windows

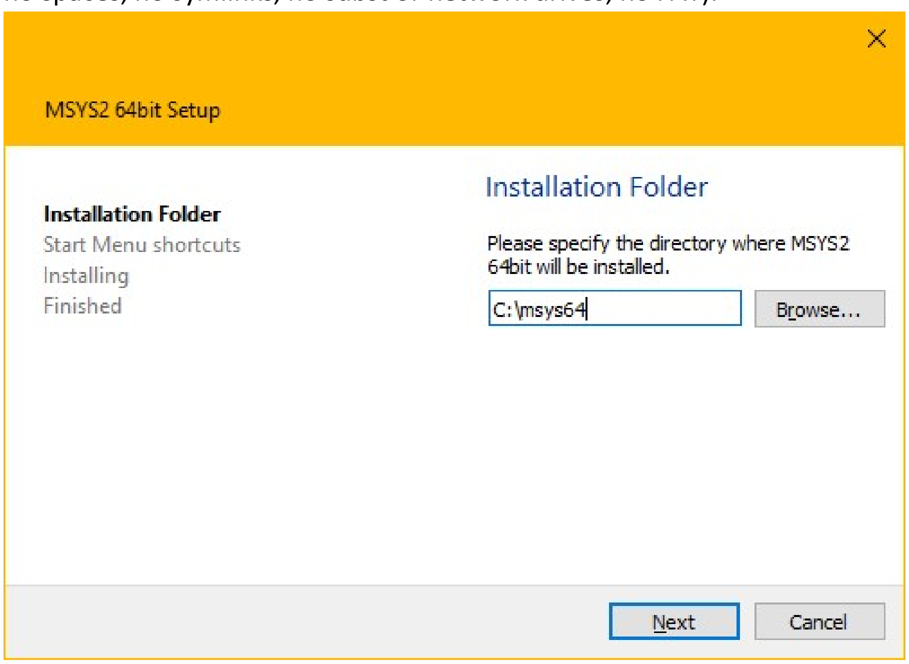
Provide two ways to install the MSYS2 tools package on Windows:
- Online installation, and
- Offline installation.

## 4.1 Install the MSYS2 - Online

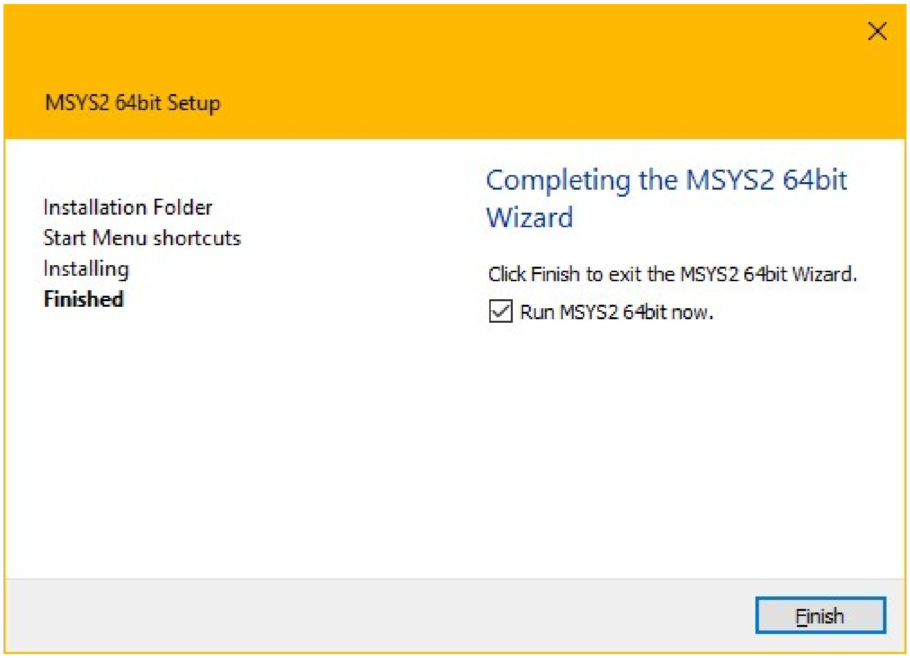
This is a reference to MSYS2. A few notes have been added.

1. Download the latest msys2 installer from https://www.msys2.org/.
2. Run the installer. MSYS2 requires 64 bit Windows 7 or newer.

3. Enter your desired Installation Folder (short ASCII-only path on a NTFS volume, no accents, no spaces, no symlinks, no subst or network drives, no FAT).




4. When done, tick Run MSYS2 now.




5. Update the package database and base packages. If you miss the step 4 and the Msys2 is not started, go to the installation folder “C:\msys64” and run “***msys2.exe***”. Unless your setup file is very recent, it will take two steps. First run ***pacman -Syu*** (***Note**: Select Y at the end):

```
$ pacman -Syu
:: Synchronizing package databases...
 mingw32                          805.0 KiB
 mingw32.sig                      438.0    B
 mingw64                          807.9 KiB
 mingw64.sig                      438.0    B
 msys                             289.3 KiB
 msys.sig                         438.0    B
:: Starting core system upgrade...
warning: terminate other MSYS2 programs before proceeding
resolving dependencies...
looking for conflicting packages...

Packages (6) bash-5.1.004-1    filesystem-2021.01-1
             mintty-1~3.4.4-1    msys2-runtime-3.1.7-4
             pacman-5.2.2-9    pacman-mirrors-20201208-1

Total Download Size:      11.05 MiB
```

```
Total Installed Size:   53.92 MiB
Net Upgrade Size:       -1.24 MiB

:: Proceed with installation? [Y/n]
:: Retrieving packages...
 bash-5.1.004-1-x86_64                 2.3 MiB
 filesystem-2021.01-1-any             33.2 KiB
 mintty-1~3.4.4-1-x86_64             767.2 KiB
 msys2-runtime-3.1.7-4-x86_64          2.6 MiB
 pacman-mirrors-20201208-1-any         3.8 KiB
 pacman-5.2.2-9-x86_64                 5.4 MiB
(6/6) checking keys in keyring          100%
(6/6) checking package integrity        100%
(6/6) loading package files             100%
(6/6) checking for file conflicts       100%
(6/6) checking available disk space     100%
:: Processing package changes...
(1/6) upgrading bash                    100%
(2/6) upgrading filesystem              100%
(3/6) upgrading mintty                  100%
(4/6) upgrading msys2-runtime           100%
(5/6) upgrading pacman-mirrors          100%
(6/6) upgrading pacman                  100%
:: To complete this update all MSYS2 processes including this terminal will be closed.

Confirm to proceed [Y/n]
```

6. Go to the installation folder “C:\msys64” and run “***msys2.exe***”. Update the rest of the base packages with ***pacman -Su*** (***Note**:):

```
$ pacman -Su
:: Starting core system upgrade...
 there is nothing to do
:: Starting full system upgrade...
resolving dependencies...
looking for conflicting packages...

Packages (20) base-2020.12-1   bsdtar-3.5.0-1
              [... more packages listed ...]

Total Download Size:    12.82 MiB
```

Total Installed Size: 44.25 MiB
Net Upgrade Size: 3.01 MiB

:: Proceed with installation? [Y/n]
[... downloading and installation continues ...]

7. Now MSYS2 is ready for you. You need to install some essential tools to compile libxcoder and ffmpeg. Below are the commands to install tools:

```
$ pacman -S –needed gcc
$ pacman -S –needed mingw-w64-x86_64-toolchain
$ pacman -S –needed base-devel
$ pacman -S –needed yasm
$ pacman -S –needed mingw-w64-x86_64-SDL2
$ pacman -S –needed git
$ pacman -S –needed xz
```

8. To start building using the mingw-w64 GCC, go to the installation folder "C:\msys64" and run "***mingw64.exe***". Now the tool is ready to build ffmpeg for Windows.

Note – Refer Appendix-A to update Windows environment variables PATH to include MSYS2 environment. This will help access MSYS2 linux commands from Windows Command Terminal window as well.

## 4.2 Install the MSYS2 – Offline

*Skip these steps if you have already run steps in Section 2.1.*

1. Get the offline installation package "msys64.zip" from NETINT Support team (Internal Share Point).
2. Unzip the msys64.zip to the C: folder. The essential tools have already been installed in this package.
3. To start building using the mingw-w64 GCC, go to the installation folder "C:\msys64" and run "***mingw64.exe***".

Note 1 – This offiline zip package comes with home folder C:\msys64\home\Administrator.
Note 2 – This offiline zip package comes with a ffmpeg tar file base_ffmpeg_n4.2.1.tar.gz. If following the offline Section 2.2, you will untar this file in the steps for Section 3.2. File can be found in C:\msys64\home\Administrator folder.

# 5 Compile Libxcoder and FFmpeg

## 5.1 Overview

In order to use NETINT T408 encoding and decoding features using FFmpeg, the user needs the following minimum steps:

```
Get a FFmpeg Repository → Get a T408 Release Package with
libxcoder and patch files → Patch the FFmpeg Repository →
Compile libxcoder → Compile ffmpeg → Locate and run
init_rsrc.exe in a command window → Run ffmpeg command in a
second command window.
```

## 5.2 Download FFmpeg Repository

Start a MSYS2 → Mingw64 terminal.

Using ffmpeg GitHub

To clone the GitHub repository for ffmpeg version, say 4.2.1, from the command line, run the following command:

(**Note** – the example below assumes a username ‘nvme’, so the home folder is C:/msys64/home/nvme. Replace ‘nvme’ with the correct username on your PC. Check under C: → msys64 → home)

```
cd C:

cd msys64/home/nvme

git clone -b n4.2.1 --depth=1
https://github.com/FFmpeg/FFmpeg.git FFmpeg
```

Other valid options for ‘-b’ are n4.2.4, n4.3.2, and n4.4

Using ffmpeg tar.xz download

‘Download xz tarball’ from https://ffmpeg.org/download.html#releases → Move the file to C:/msys64/home/nvme directory → Run the following steps. Release 4.3.2 used in untaring example below.

```
cd C:

cd msys64/home/nvme
```

```
xz -d ffmpeg-4.3.2.tar.xz
tar -xvf ffmpeg-4.3.2.tar
mv ffmpeg-4.3.2 FFmpeg
```

<u>Using offline pre-packaged ffmpeg tar file</u>

If you followed the offline steps in Section 2.2, then you can also find the downloaded ffmpeg package “base_ffmpeg_n4.2.1.tar.gz” in offline package folder “C:\msys64\home\Administrator\”. Run the following command to untar this tar file:

```
tar -zxf base_ffmpeg_n4.2.1.tar.gz
```

## 5.3 Download T408 Release Package

Contact the NETINT Support team to get the latest T408 software release package. The software package will include a tar.gz file titled ‘Codensity_T408_Software_Release_V?.?.?.tar.gz’. Where, ?.?.? is the software release version like 2.5.9.

## 5.4 Apply Patch to FFmpeg Repository

The following instructions need to be done in sequence to prepare FFmpeg for use with the Codensity T408 Video Transcoder.

1. Untar the Codensity T408 Software Release package to the $HOME folder:

    ```
    cd C:
    cd msys64/home/nvme
    tar -zxf Codensity_T408_Software_Release_V*.tar.gz
    ```

2. Copy the *libxcoder/* folder from the release package to the parent folder of FFmpeg (i.e. same level as FFmpeg):

    ```
    cp -r release/libxcoder ./
    ```

3. Copy the FFmpeg patch file from the release package to the *FFmpeg* folder:
   Note – this example is for patching a FFmepg 4.2.x release.

    ```
    cp release/FFmpeg-n4.2.1_t408_patch $HOME/FFmpeg
    ```

4. Change directories to the *FFmpeg/* folder:

```
cd C:
cd msys64/home/nvme
cd FFmpeg
```

5. Apply the patch depending on the FFmpeg `<version>` (ex. 2.7.1, 3.4.2, 4.1.3, 4.2.1, 4.3.1):

```
patch -t -p 1 < FFmpeg-n4.2.1_T408_patch
```

| |
|---|
| It is critical to apply the correct NETINT patch for the FFmpeg version in use. Check with NETINT Support team for the patch to use. In general, patch file FFmpeg-**n4.2.x**_T408_patch will work with all FFmpeg 4.2 versions, patch file FFmpeg-n4.3.x_T408_patch will work with all FFMpeg 4.3 versions, and so on. |

## 5.5 Build FFmpeg with NETINT Codec Library

The following instructions need to be done in sequence to build FFmpeg with the NETINT Codec Library.

1. From the *FFmpeg/* folder, go to the *libxcoder/* folder with the following command:

```
cd C:
cd msys64/home/nvme
cd FFmpeg
cd ../libxcoder
```

2. Build and install libxcoder with the following commands. Upon a successful compile the files are auto installed to /usr/local/bin folder.

```
bash build.sh -w
ls /usr/local/bin
```

```
$ ls /usr/local/bin
init_rsrc  ni_rsrc_list  ni_rsrc_mon  ni_rsrc_update
```

3. Update pkg-config path:

```
export PKG_CONFIG_PATH=/usr/local/lib/pkgconfig:$PKG_CONFIG_PATH
```

4. Go to the *FFmpeg/* directory with the following command:

```
cd ../FFmpeg
```

5. Run the `build_FFmpeg.sh` script with the following commands. When building successfully, the EXEs and libraries can be found in */usr/local/bin* and */usr/local/lib* folders:

```
make clean

bash build_FFmpeg.sh -w

make install
```

6. Some related libraries are compiled dynamically, so the following libraries “libiconv-2.dll, libwinpthread-1.dll, SDL2.dll, zlib1.dll” need to be copied from C:\msys64\mingw64\bin to C:\msys64\usr\local\bin

   In a MSYS2 → Mingw64 terminal window run the following commands to confirm the folder contents:

```
ls /usr/local/bin

ls /usr/local/lib/*
```

```
$ ls /usr/local/bin

SDL2.dll    init_rsrc.exe   libwinpthread-1.dll  ni_rsrc_mon.exe
zlib1.dll ffmpeg.exe  libiconv-2.dll  ni_rsrc_list.exe
ni_rsrc_update.exe
```

```
$ ls /usr/local/lib/*

/usr/local/lib/libavcodec.a   /usr/local/lib/libpostproc.a
/usr/local/lib/libavdevice.a  /usr/local/lib/libswresample.a
/usr/local/lib/libavfilter.a  /usr/local/lib/libswscale.a
/usr/local/lib/libavformat.a  /usr/local/lib/libxcoder.a
/usr/local/lib/libavutil.a

/usr/local/lib/pkgconfig:
libavcodec.pc   libavfilter.pc  libavutil.pc    libswresample.pc
xcoder.pc
libavdevice.pc  libavformat.pc  libpostproc.pc  libswscale.pc
```

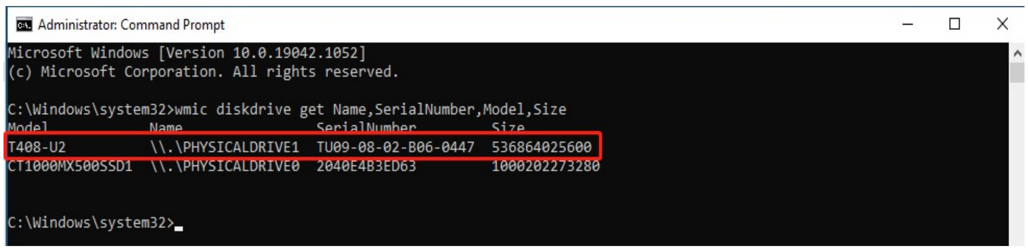
This completes the installation steps in prep to start using T408 card on your Windows host.

# 6 Run FFmpeg with T408 Cards

After a successful compilation, we can test with T408 cards.

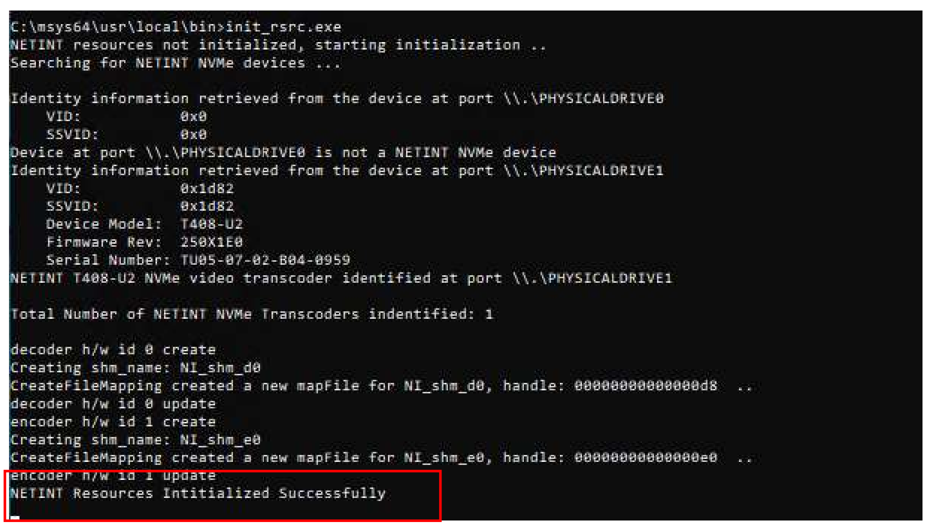
1. Check Hardware.
   With administrator privileges, open a Windows command terminal. Check the T408 card presence with the following command:

```
$ wmic diskdrive get Name,SerialNumber,Model,Size
```




2. Initialize T408
   With administrator privileges, open a **Windows command terminal**. Go to C: → msys64 → usr → local → bin folder. Initialize the T408 with the following command (*Note: this process must keep running during transcoding).

```
cd C:/msys64/usr/local/bin

init_rsrc.exe
```




3. Run FFmpeg.

With administrator privileges, open a **Windows command terminal**. Go to your *C:/msys64/usr/local/bin folder*. Run the following commands with ffmpeg to do the transcoding.

The following examples use clips folder *C:/msys64/home/nvme/libxcoder/test*. Update this path if you wish to run custom clips for decoding or encoding.

```
cd C:/msys64/usr/local/bin
```

```
ffmpeg.exe
```

- test h264 decoder:

```
$ ffmpeg.exe -y -hide_banner -nostdin -vsync 0 -c:v h264_ni_dec -i
C:/msys64/home/nvme/libxcoder/test/akiyo_352x288p25.264 -c:v rawvideo
output_5.yuv
```

- test h265 decoder:

```
$ ffmpeg.exe -y -hide_banner -nostdin -vsync 0 -c:v h265_ni_dec -i
C:/msys64/home/nvme/libxcoder/test/akiyo_352x288p25.265 -c:v rawvideo
output_6.yuv
```

- test h264 encoder:

```
$ ffmpeg.exe -y -hide_banner -nostdin -f rawvideo -pix_fmt yuv420p -s:v
352x288 -r 25 -i C:/msys64/home/nvme/libxcoder/test/akiyo_352x288p25.yuv -c:v
h264_ni_enc output_7.h264
```

- test h265 encoder:

```
$ ffmpeg.exe -y -hide_banner -nostdin -f rawvideo -pix_fmt yuv420p -s:v
352x288 -r 25 -i C:/msys64/home/nvme/libxcoder/test/akiyo_352x288p25.yuv -c:v
h265_ni_enc output_8.h265
```

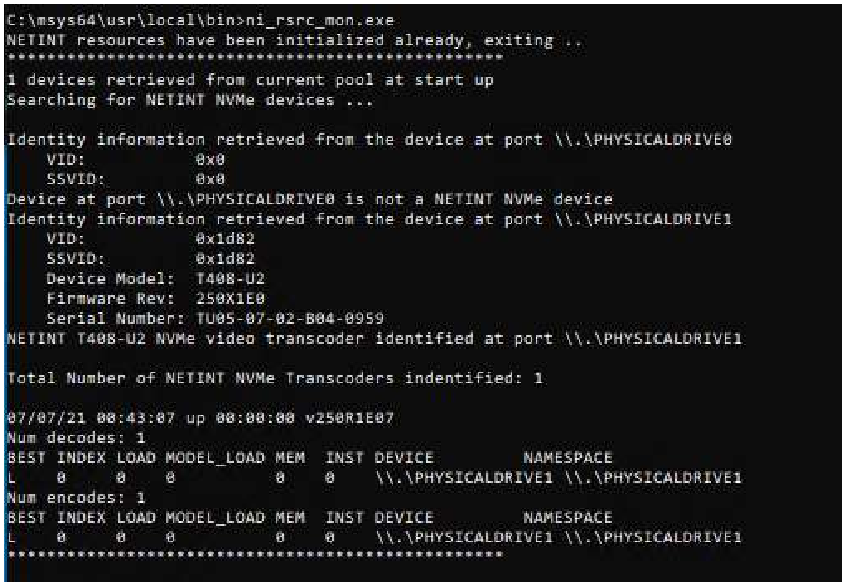
- test 264->265 transcoder:

```
$ ffmpeg.exe -y -hide_banner -nostdin -vsync 0 -c:v h264_ni_dec -i
C:/msys64/home/nvme/libxcoder/test/1280x720p_Basketball.264 -c:v h265_ni_enc
output_9.h265
```

4. Monitoring Load

   With administrator privileges, open a **Windows command terminal**. Go to your *C:/msys64/usr/local/bin* folder. Run the following command:

```
ni_rsrc_mon.exe
```

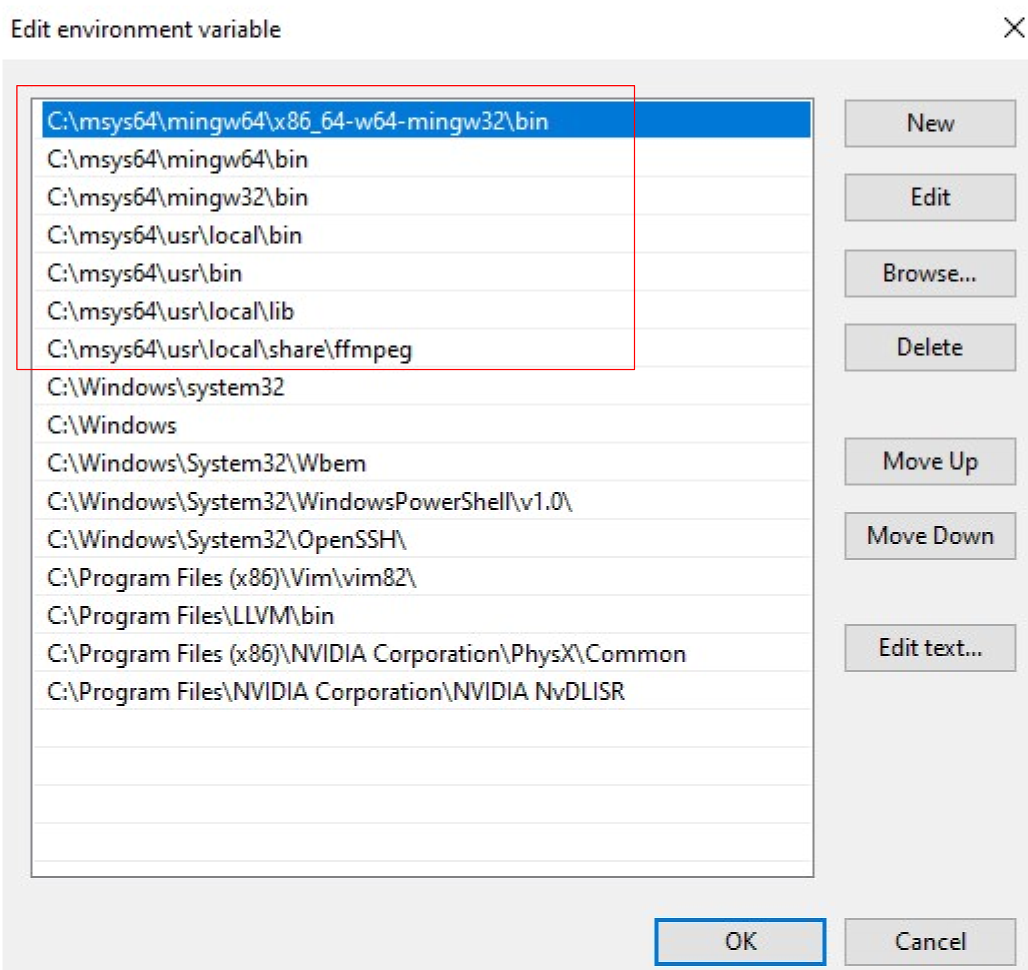


```
C:\msys64\usr\local\bin>ni_rsrc_mon.exe
NETINT resources have been initialized already, exiting ..
**************************************************
1 devices retrieved from current pool at start up
Searching for NETINT NVMe devices ...

Identity information retrieved from the device at port \\.\PHYSICALDRIVE0
    VID:          0x0
    SSVID:        0x0
Device at port \\.\PHYSICALDRIVE0 is not a NETINT NVMe device
Identity information retrieved from the device at port \\.\PHYSICALDRIVE1
    VID:          0x1d82
    SSVID:        0x1d82
    Device Model:  T408-U2
    Firmware Rev:  250X1E0
    Serial Number: TU05-07-02-B04-0959
NETINT T408-U2 NVMe video transcoder identified at port \\.\PHYSICALDRIVE1

Total Number of NETINT NVMe Transcoders indentified: 1

07/07/21 00:43:07 up 00:00:00 v250R1E07
Num decodes: 1
BEST INDEX LOAD MODEL_LOAD MEM  INST DEVICE         NAMESPACE
L    0     0    0          0    0    \\.\PHYSICALDRIVE1 \\.\PHYSICALDRIVE1
Num encodes: 1
BEST INDEX LOAD MODEL_LOAD MEM  INST DEVICE         NAMESPACE
L    0     0    0          0    0    \\.\PHYSICALDRIVE1 \\.\PHYSICALDRIVE1
**************************************************
```

```
Reporting columns
BEST         flag showing card of lowest realtime load and to be
             selected for next auto allocated job
INDEX        index number used by resource manager to identify the
             resource
LOAD         realtime load
MODEL_LOAD   estimated load based on framerate and resolution
INST         number of job instances
DEVICE       path to NVMe device file handle
NAMESPACE    path to NVMe namespace file handle
```

# 7 Appendix–A : Sourcing MSYS2 environment from Windows Environment Variables

In Windows ‘Control Panel → System → Advanced System Settings → System Properties → Environment Variables… → System Variables → Path → Edit’ you can add new environment variables for C:\msys64 as per sample picture below. Use the ‘Move Up’ button to prioritize the C:\msys64 paths over C:\Windows paths.




This will help source MSYS2 linux type commands from Windows Command Terminal window. Once the above is configured, open a new Windows Command Terminal Window and try commands like:

```
which bash
which sh
which ffmpeg.exe
```

```
which init_rsrc.exe
```

# 8 Revision History

| Version | Change | Creation Date | Author |
|---|---|---|---|
| 1.0 | Document Created | November 29,2020 | Achraf Jerbi |
| 1.1 | Modify Msys Env setup | May 11,2021 | Roy Liu |
| 1.2 | Modify directories | May 14, 2021 | Achraf Jerbi |
| 1.3 | Add off-line flow | July 01, 2021 | Roy Liu |
| 1.4 | Extend Section 3 and 4 to clarify NETINT patches usage for ffmpeg 4.2.4 and newer releases. Update install steps to use standard C:/msys64/home/*username* and C:/msys64/usr/local/bin paths. | July 07, 2021 | Balwinder Sidhu |
| 2.0 | Updated as per new app note template | December 7, 2021 | Balwinder Sidhu |

# 9 Legal Notice

Information in this document is provided in connection with NETINT products. No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted by this document. Except as provided in NETINT's terms and conditions of sale for such products, NETINT assumes no liability whatsoever and NETINT disclaims any express or implied warranty, relating to sale and/or use of NETINT products including liability or warranties relating to fitness for a particular purpose, merchantability, or infringement of any patent, copyright or other intellectual property right.

A "Mission Critical Application" is any application in which failure of the NETINT Product could result, directly or indirectly, in personal injury or death. Should you purchase or use NETINT's products for any such mission critical application, you shall indemnify and hold NETINT and its subsidiaries, subcontractors and affiliates, and the directors, officers, and employees of each, harmless against all claims costs, damages, and expenses and reasonable attorney's fees arising out of, directly or indirectly, any claim of product liability, personal injury, or death arising in any way out of such mission critical application, whether or not NETINT or its subcontractor was negligent in the design, manufacture, or warning of the NETINT product or any of its parts.

NETINT may make changes to specifications, technical documentation, and product descriptions at any time, without notice. The information here is subject to change without notice. Do not finalize a design with this information. The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications.

NETINT, Codensity, and NETINT Logo are trademarks of NETINT Technologies Inc. All other trademarks or registered trademarks are the property of their respective owners.