

APPS607– Quadra Multi-Namespace Application Note

V3.0

20/5/2025

1 Table of Abbreviations 3
2 References 4
3 Background 5
3.1 Intended Audience 5
3.2 Compatibility 5
3.3 Reference Documentation: 5
4 Steps to Enable Multi-Namespaces 7
4.1 Enable namespaces without SR-IOV 8
4.1.1 Increasing the namespace count without SRIOV 8
4.1.2 Applying changes 9
4.1.3 Verification 10
4.2 Enabling namespaces with SR-IOV 11
4.2.1 Specifying the number of Namespaces 11
4.2.2 Verification 13
4.3 Adding namespaces up to the 128 maximum limit (Release 5.1.0 onwards) 15
4.3.1 Disable other namespaces 15
4.3.2 Set the maximum namespace to non PF 16
4.3.3 Unsetting active VF 17
4.3.4 Enabling SR-IOV function on previously unset namespace 17
4.4 Add nvme driver to virtual machine 18
5 Revision History 20
6 Reviewers 20
7 Legal Notice 22

1 Table of Abbreviations

Abbreviation:	Full Form:
NS	Namespace
LBA	Logic Block Addressing
SR-IOV	Single Root – Input Output Virtualization
AI	Artificial Intelligence
API	Application Programming Interface
AUD	Access Unit Delimiters
AV	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
Multi-NS	Multi-Namespace
NVMe	Non-Volatile Memory express
PCIe	Peripheral Component Interconnect express
PF	Physical Function
PPS	Picture Parameter Set
QP	Quantization Parameter
RGB	Red Green Blue

RGBA	Red Green Blue Alpha
RDO	Rate Distortion 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 Rate
VCL	Video Coding Layer
VF	Virtual Function
VPS	Video Parameter Set
VUI	Video Usability Information

2 References

Technical Notes:

Specifications :

3 Background

This document describes how to enable multi-namespaces with and without SR-IOV on the QUADRA NETINT Transcoding device.

3.1 Intended Audience

This document is intended for customers and engineers who want to utilize the multi-namespaces available in NVMe.

3.2 Compatibility

Software Compatibility

This guide is applicable to NETINT Quadra Video Transcoder, software Release 4.2.0, and beyond.

Hardware Compatibility

This feature is supported by all NETINT Quadra Transcoder hardware devices.

Operation systems

Both Windows and Linux support these features.

3.3 Reference Documentation:

See Application Note

APPS525 Codensity QUADRA SR-IOV Configuration and Usage Guide for additional information.

4 Steps to Enable Multi-Namespaces

Because Multi-Namespaces are configured dynamically we require a tool to configure them if the Quadra device is executing sessions.

The tool **ni_rsrc_namespace** is available within the build directory after libxcoder has been built.

Run **ni_rsrc_namespace** with -**h** to see its usage information and the command help.

There are three important parameters :

-d is used to specify the device, such as **/dev/nvme0n1**

-n is used to specify the number of namespaces

-s is used to specify the SRIOV number if enabled

4.1 Enable namespaces without SR-IOV

If SR-IOV is not enabled, specify -**d** and -**n** to **ni_rsrc_namespace.** The maximum number of namespaces is 128.

As of Release 4.5.0, the maximum total number of namespaces is 64 across all physical and virtual functions.

As of Release 5.1.0, the maximum total number of namespaces is 128 across all physical and virtual functions.

See the Application Note **APPS525 Codensity QUADRA SR-IOV Configuration and Usage Guide** for additional information.

4.1.1 Increasing the namespace count without SRIOV

There is 1 namespace for the physical function enabled by default. The command below will define 4 namespaces for the physical function

```
ni_rsrc_namespace -d /dev/nvme0n1 -n 4
```

4.1.2 Applying changes

The NVMe driver must be reloaded to activate any changes in multi-namespace.

```
sudo modprobe -r nvme
```

```
sudo modprobe nvme
```

4.1.3 Verification

To verify the results, check the NVMe list with the following command

```
sudo nvme list
```

Sample output:

Node	SN	Model	Namespace	Usage	Format	FW Rev
/dev/nvme0n1	Q1A003A01FC049-0116	QuadraT1A-EP1	1	8.59 TB / 8.59 TB	4 KiB + 0 B	---6HDEV
/dev/nvme0n2	Q1A003A01FC049-0116	QuadraT1A-EP1	2	8.59 TB / 8.59 TB	4 KiB + 0 B	---6HDEV
/dev/nvme0n3	Q1A003A01FC049-0116	QuadraT1A-EP1	3	8.59 TB / 8.59 TB	4 KiB + 0 B	---6HDEV
/dev/nvme0n4	Q1A003A01FC049-0116	QuadraT1A-EP1	4	8.59 TB / 8.59 TB	4 KiB + 0 B	---6HDEV

4.2 Enabling namespaces with SR-IOV

If SR-IOV is enabled then we must specify -**d**, -**n** and **-s** to **ni_rsrc_namespace**. The maximum number of namespaces is 128, and the maximum number of virtual functions is 6.

See **APPS525 Codensity QUADRA SR-IOV Configuration and Usage Guide** for additional information.

4.2.1 Specifying the number of Namespaces

Different numbers of namespaces can be set with different SR-IOV configurations. The command below configures three SRIOVs, each with a different number of namespaces.

```
sudo sh -c " echo 2 > /sys/bus/pci/devices/pci
device/sriov_numvfs"
ni_rsrc_namespace -d /dev/nvme0n1 -n 3 -s 0
ni_rsrc_namespace -d /dev/nvme1n1 -n 4 -s 1
ni_rsrc_namespace -d /dev/nvme2n1 -n 5 -s 2
```

Re-load the NVMe driver for the number of multi-namespaces to take effect.

```
sudo modprobe -r nvme
sudo modprobe nvme
```

Note: If driver can't be unloaded, you can rebind the device with **unbind** and **bind**.

```
echo 0000:01:00.0 > /sys/bus/pci/drivers/nvme/unbind
echo 0000:01:00.1 > /sys/bus/pci/drivers/nvme/unbind
echo 0000:01:00.2 > /sys/bus/pci/drivers/nvme/unbind
echo 0000:01:00.0 > /sys/bus/pci/drivers/nvme/bind
echo 0000:01:00.1 > /sys/bus/pci/drivers/nvme/bind
echo 0000:01:00.2 > /sys/bus/pci/drivers/nvme/bind
```

4.2.2 Verification

Verify the correct numbers have been applied using the NVMe device list.

```
sudo nvme list
```

Sample output:

Node	SN	Model	Namespace	Usage	Format	FW Rev
/dev/nvme0n1	Q1A003A01FC049-0116	QuadraT1A-EP1	1	8.59 TB /8.59 TB	4 KiB + 0 B	---6HDEV
/dev/nvme0n2	Q1A003A01FC049-0116	QuadraT1A-EP1	2	8.59 TB /8.59 TB	4 KiB + 0 B	---6HDEV
/dev/nvme0n3	Q1A003A01FC049-0116	QuadraT1A-EP1	3	8.59 TB /8.59 TB	4 KiB + 0 B	---6HDEV
/dev/nvme1n1	Q1A003A01FC049-0116	QuadraT1A-EP1	1	8.59 TB /8.59 TB	4 KiB + 0 B	---6HDEV
/dev/nvme1n2	Q1A003A01FC049-0116	QuadraT1A-EP1	2	8.59 TB /8.59 TB	4 KiB + 0 B	---6HDEV
/dev/nvme1n3	Q1A003A01FC049-0116	QuadraT1A-EP1	3	8.59 TB /8.59 TB	4 KiB + 0 B	---6HDEV
/dev/nvme1n4	Q1A003A01FC049-0116	QuadraT1A-EP1	4	8.59 TB /8.59 TB	4 KiB + 0 B	---6HDEV
/dev/nvme2n1	Q1A003A01FC049-0116	QuadraT1A-EP1	1	8.59 TB /8.59 TB	4 KiB + 0 B	---6HDEV
/dev/nvme2n2	Q1A003A01FC049-0116	QuadraT1A-EP1	2	8.59 TB /8.59 TB	4 KiB + 0 B	---6HDEV
/dev/nvme2n3	Q1A003A01FC049-0116	QuadraT1A-EP1	3	8.59 TB /8.59 TB	4 KiB + 0 B	---6HDEV

/dev/nvme2n4	Q1A003A01FC049-0116	QuadraT1A-EP1	4	8.59 TB	8.59 TB	4 KiB + 0 B	---6HDEV
/dev/nvme2n5	Q1A003A01FC049-0116	QuadraT1A-EP1	5	8.59 TB	8.59 TB	4 KiB + 0 B	---6HDEV

The above output shows multi-namespaces enabled on the host.

4.3 Adding namespaces up to the 128 maximum limit (Release 5.1.0 onwards)

To reiterate, the default namespace count for each function is 1. This means that for every SR-IOV function enabled, its starting namespace count is 1. To enable up to 128 namespaces, and to satisfy the 128 namespace limit follow the instructions below.

4.3.1 Disable other namespaces

Setting: We want to make the PF have 128 namespaces, no SR-IOV functions are to be used

1. Unset the namespace count of SR-IOV slots

```
a. ni_rsrc_namespace -d /dev/nvme0n1 -n 0 -s 1
b. ni_rsrc_namespace -d /dev/nvme0n1 -n 0 -s 2
c. ni_rsrc_namespace -d /dev/nvme0n1 -n 0 -s 3
d. ni_rsrc_namespace -d /dev/nvme0n1 -n 0 -s 4
e. ni_rsrc_namespace -d /dev/nvme0n1 -n 0 -s 5
f. ni_rsrc_namespace -d /dev/nvme0n1 -n 0 -s 6
g. ni_rsrc_namespace -d /dev/nvme0n1 -n 0 -s 7
```

2. Grant the PF 128 namespaces

```
a. ni_rsrc_namespace -d /dev/nvme0n1 -n 128 -s 0
```

3. Reload the nvme driver to apply changes

```
sudo modprobe -r nvme
sudo modprobe nvme
```

4.3.2 Set the maximum namespace to non PF

There are up to 127 namespaces for the VF. The PF must be configured with at least 1 namespace. Follow the steps above whilst choosing which slots to unset and set.

4.3.3 Unsetting active VF

Note that if you try to unset an active SR-IOV function, it will fail silently. The proper procedure would be to unbind the device before issuing the unset:

```
echo 0000:01:00.2 > /sys/bus/pci/drivers/nvme/unbind
ni_rsrc_namespace -d /dev/nvme1n1 -n 127 -s 2
echo 0000:01:00.2 > /sys/bus/pci/drivers/nvme/bind
```

4.3.4 Enabling SR-IOV function on previously unset namespace

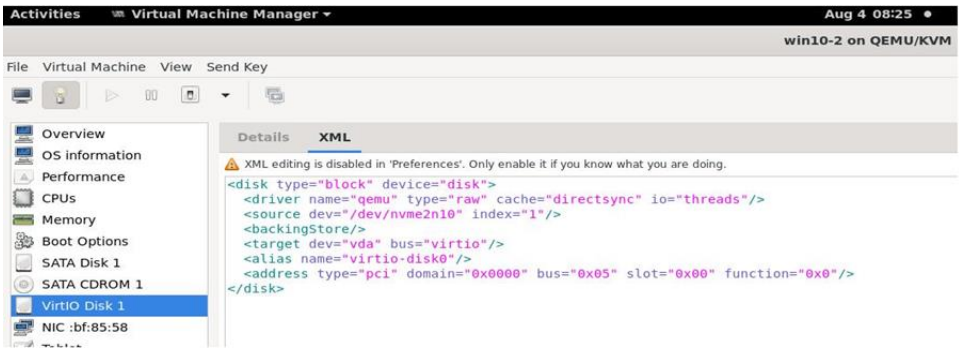
A namespace count of 0 implies that namespaces are disabled. If the following hypothetical command was successfully issued, then the SR-IOV function 2 would now be disabled in Quadra FW.

```
ni_rsrc_namespace -d /dev/nvme1n1 -n 0 -s 2
```

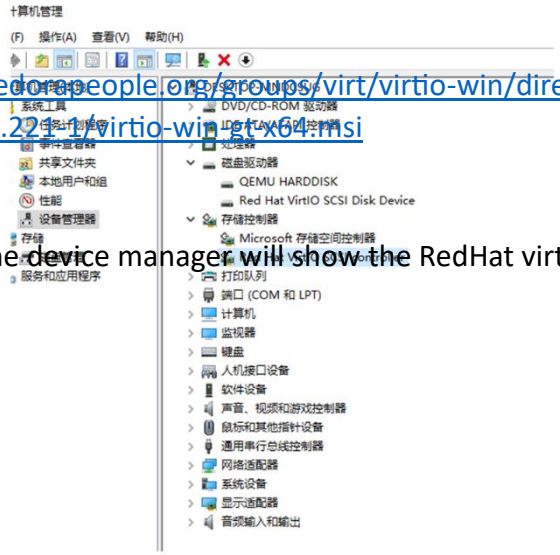
If an attempt is made on the host to enable it the firmware will ignore the enable signal, thus blocking all IO to and from function 2. This is by-design, and enforces the 0 namespace allotment. Errors may be reported in the kernel log due to the timeout of these function enable attempts.

4.4 Add nvme driver to virtual machine

```
<disk type="block" device="disk">
    <driver name="qemu" type="raw" cache="directsync" io="threads"/>
    <source dev="/dev/nvme2n10" index="1"/>
    <backingStore/>
    <target dev="vda" bus="virtio"/>
    <alias name="virtio-disk0"/>
    <address type="pci" domain="0x0000" bus="0x05" slot="0x00"
function="0x0"/>
  </disk>
```



Download https://fedorapeople.org/groups/virt/virtio-win/direct-downloads/archive-virtio/virtio-win-0.1.271-1/virtio-win-gt-x64.msi



After installation, the device manager will show the RedHat virtIO SCSI Disk Device

5 Revision History

Version	Change	Date	Author Name
0.1	Draft Document Created	10/3/2023	Mingping.zhao
1.0	First Release to customer	15/3/2023	K. Robinson
1.1	Maximum namespace limit to 64 introduced, Reviewers page added	14/7/2023	P. Tang
1.2	Expand maximum namespace limit to 128	13/2/2025	Cube.Sun
2.2	Tidied Up Document	16/05/2025	Marcus OBrien
3.0	Released (same content as 2.2)	20/5/2025	Marcus OBrien

6 Reviewers

Mandatory Reviewer	Reviewed Version	Date
Marcus	0.1	2023-03-14
Marcus	1.1	2023-08-14
Marcus	2.1	2025-05-16

There must be at least 1 mandatory reviewer. Mandatory reviewers must review and accept change, before a new version can be published.

Optional Reviewer	Reviewed Version	Date
Jack Wang	0.1	2023-03-14
Mingping Zhao	1.1	2023-07-21
N. Simon	2.1	2025-05-16

Optional reviewers will be notified when there are changes and asked to review, but they are not required to review the document before it is published.

REMOVE THIS PAGE BEFORE PUBLISHING, AND RE GEN TABLE OF CONTENTS

7 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 rights1 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.

© 2025 NETINT Technologies Inc. All rights reserved.