



Virtnet CLI Commands

Table of contents

User Front End CLI

Hotplug

Syntax

Output

Example

Unplug

Syntax

Output

Example

List

Syntax

Output

Example

Query

Syntax

Output

Example

Stats

Syntax

Output

Example

Modify Device

Syntax

Device Options

Queue Options

Output

Example

Log

Syntax

Output

Example

Validate

Syntax

Output

Example

Version

Syntax

Output

Example

Update

Syntax

Output

Example

Restart

Syntax

Output

Example

Health

Syntax

Output

Example

Error Code

User Front End CLI

To communicate with the virtio-net-controller backend service, a user frontend program, virtnet, is installed on the BlueField which is based on remote procedure call (RPC) protocol with JSON format output. Run the following command to check its usage:

```
usage: virtnet [-h] [-v]
{hotplug,unplug,list,query,modify,log,version,restart,validate,update,stats}
...

Nvidia virtio-net-controller command line interface v24.10.20

positional arguments:
{hotplug,unplug,list,query,modify,log,version,restart,validate,update,stats}
    ** Use -h for sub-command usage
hotplug                hotplug virtnet device
unplug                 unplug virtnet device
list                   list all virtnet devices
query                  query all or individual virtnet device(s)
modify                 modify virtnet device
log                     set log level
version                 show virtio net controller version info
restart                 Do fast restart of controller without
killing the service
validate                validate configurations
update                  update controller
health                  controller health utility
debug                   For debug purpose, cmds can be changed
without notice
stats                   stats of virtnet device

options:
-h, --help              show this help message and exit
```

```
-v, --version          show program's version number and exit
```

Virtnet supports command line autocomplete by inputting one command with tab.

To check the currently running controller version:

```
# virtnet -v
```

Hotplug

This command hotplugs a virtio-net PCIe PF device exposed to the host side.

Syntax

```
virtnet hotplug -i IB_DEVICE -m MAC -t MTU -n MAX_QUEUES -s  
MAX_QUEUE_SIZE [-h] [-u SF_NUM] [-f FEATURES] [-1]
```

Option	Abbr	Argument Type	Required	Description
<code>--help</code>	<code>-h</code>	N/A	No	Show the help message and exit
<code>--ib_device</code>	<code>-i</code>	String	Yes	RDMA device (e.g., <code>m1x5_0</code>) of the physical port on top of which the hotplug device is created. Options: <ul style="list-style-type: none"><code>m1x5_0</code> – port 1<code>m1x5_1</code> – port 2<code>m1x5_bond_0</code> – LAG

Option	Abbr	Argument Type	Required	Description
<code>--features</code>	<code>-f</code>	Hex Number	No	<p>Feature bits to be enabled in hex format. Refer to the "Virtio-net Feature Bits" page.</p> <p>Note Note that some features are enabled by default. Query the device to show the supported bits.</p>
<code>--mac</code>	<code>-m</code>	Number	Yes	<p>MAC address of the virtio-net device.</p> <p>Note Controller does not validate the MAC address (other than its length). The user must ensure MAC is valid and unique.</p>
<code>--mtu</code>	<code>-t</code>	Number	Yes	<p>Maximum transmission unit (MTU) size of the virtio-net device. It must be less than the uplink rep MTU size.</p>
<code>--num_queues</code>	<code>-n</code>	Number	Yes	<p>Mutually exclusive with <code>max_queue_pairs</code>. Max number of virt queues could be created for the virtio-net device. TX, RX, ctrl queues are counted separately (e.g., 3 has 1 TX VQ, 1 RX VQ, 1 Ctrl VQ).</p> <p>Note This option will be depreciated in the future.</p>

Option	Abbr	Argument Type	Required	Description
<code>--max_queue_pairs</code>	<code>-qp</code>	Number	Yes	Mutually exclusive with <code>num_queues</code> . Number of data VQ pairs. One VQ pair has one TX queue and one RX queue. It does not count control or admin VQ. From the host side, it appears as <code>Pre-set maximums->Combined</code> in <code>ethtool -l <virtio-dev></code> .
<code>--max_queue_size</code>	<code>-s</code>	Number	Yes	Maximum number of buffers in the virt queue, between 0x4 and 0x8000. Must be power of 2.
<code>--sf_num</code>	<code>-u</code>	Number	No	SF number to be used for this hotplug device, must between 2000 and 2999.
<code>--legacy</code>	<code>-l</code>	N/A	No	Create legacy (transitional) hotplug device

Output

Entry	Type	Description
<code>bdf</code>	String	The PCIe BDF (bus:device:function) number enumerated by host. The user should see this PCIe device from host side.
<code>vuid</code>	String	Unique device SN. It can be used as an index to query/modify/unplug this device.
<code>id</code>	Num	Unique device ID. It can be used as an index to query/modify/unplug this device.
<code>transitional</code>	Num	Is the current device a transitional hotplug device. <ul style="list-style-type: none"> • 0 – modern device • 1 – transitional device

Entry	Type	Description
<code>sf_rep_net_device</code>	String	The SF representor name represents the virtio-net device. It should be added into the OVS bridge.
<code>mac</code>	String	The hotplug virtio-net device MAC address
<code>errno</code>	Num	Error number if hotplug failed. <ul style="list-style-type: none"> • 0 – success • non-0 – failed
<code>errstr</code>	String	Explanation of the error number

Example

The following example of hot plugging one device with MAC address `0C:C4:7A:FF:22:93`, MTU 1500, and 1 pair of virtual queue (QP) pair with a depth of 1024 entries. The device is created on the physical port of `mlx5_0`.

```
# virtnet hotplug -i mlx5_0 -m 0C:C4:7A:FF:22:93 -t 1500 -qp 1 -s
1024
{
  "bdf": "15:00.0",
  "vuid": "MT2151X03152VNETS1D0F0",
  "id": 0,
  "transitional": 0,
  "sf_rep_net_device": "en3f0pf0sf2000",
  "mac": "0C:C4:7A:FF:22:93",
  "errno": 0,
  "errstr": "Success"
}
```

Unplug

This command unplugs a virtio-net PCIe PF device.

Syntax

```
virtnet unplug [-h] [-p PF | -u VUID]
```

Only one of `--pf` and `--vuid` is needed to unplug the device.

Option	Abbr	Argument Type	Required	Description
<code>--help</code>	<code>-h</code>	N/A	No	Show the help message and exit
<code>--pf</code>	<code>-p</code>	Number	Yes	Unique device ID returned when doing hotplug. Can be retrieved by using <code>virtnet list</code> .
<code>--vuid</code>	<code>-u</code>	String	Yes	Unique device SN returned when doing hotplug. Can be retrieved by using <code>virtnet list</code> .

Output

Entry	Type	Description
<code>errno</code>	Num	Error number if operation failed <ul style="list-style-type: none">• 0 – success• non-0 – failed
<code>errstr</code>	String	Explanation of the error number

Example

Unplug-hotplug device using the PF ID:

```
# virtnet unplug -p 0
{'id': '0x1'}
{
  "errno": 0,
  "errstr": "Success"
}
```

List

This command lists all existing virtio-net devices, with global information and individual information for each device.

Syntax

```
virtnet list [-h]
```

Option	Abbr	Argument Type	Required	Description
<code>--help</code>	<code>-h</code>	N/A	No	Show the help message and exit

Output

The output has two main sections. The first section wrapped by the `controller` are global configurations and capabilities.

Entry	Type	Description
<code>controller</code>	String	Entries under this section is global information for the controller

Entry	Type	Description
<code>emulation_manager</code>	String	The RDMA device manager used to manage internal resources. Should be default <code>mlx5_0</code> .
<code>max_hotplug_devices</code>	String	Maximum number of devices that can be hotplugged
<code>max_virt_net_devices</code>	String	Total number of emulated devices managed by the device emulation manager
<code>max_virt_queues</code>	String	Maximum number of virt queues supported per device
<code>max_tunnel_descriptors</code>	String	Maximum number of descriptors the device can send in a single tunnel request
<code>supported_features</code>	String	Total list of features supported by device
<code>supported_virt_queue_types</code>	String	Currently supported virt queue types: Packed and Split
<code>supported_event_modes</code>	String	Currently supported event modes: <code>no_msix_mode</code> , <code>qp_mode</code> , <code>msix_mode</code>

Each device has its own section under `devices`.

Entry	Type	Description
<code>devices</code>	String	Entries under this section is per device information
<code>pf_id</code>	Number	Physical function ID
<code>function_type</code>	String	Function type: Static PF, hotplug PF, VF

Entry	Type	Description
<code>transitional</code>	Number	The current device a transitional hotplug device: <ul style="list-style-type: none"> • 0 – modern device • 1 – transitional device
<code>vuid</code>	String	Unique device SN, it can be used as an index to query/modify/unplug a device
<code>pci_bdf</code>	String	Bus:device:function to describe the virtio-net PCIe device
<code>pci_vhca_id</code>	Number	Virtual HCA identifier for the general virtio-net device. For debug purposes only.
<code>pci_max_vfs</code>	Number	Maximum number of virtio-net VFs that can be created for this PF. Valid only for PFs.
<code>enabled_vfs</code>	Number	Currently enabled number of virtio-net VFs for this PF
<code>msix_num_pool_size</code>	Number	Number of free dynamic MSIX available for the VFs on this PF
<code>min_msix_num</code>	Number	The minimum number of dynamic MSI-Xs that can be set for an virtio-net VF
<code>max_msix_num</code>	Number	The maximum number of dynamic MSI-Xs that can be set for an virtio-net VF
<code>min_num_of_qp</code>	Number	The minimum number of dynamic data VQ pairs (i.e., each pair has one TX and 1 RX queue) that can be set for an virtio-net VF
<code>max_num_of_qp</code>	Number	The maximum number of dynamic data VQ pairs (i.e., each pair has one TX and 1 RX queue) that can be set for an virtio-net VF

Entry	Type	Description
<code>qp_pool_size</code>	Number	Number of free dynamic data VQ pairs (i.e., each pair has one TX and 1 RX queue) available for the VFs on this PF
<code>num_msix</code>	Number	Maximum number of MSI-X available for this device
<code>num_queues</code>	Number	Maximum virtual queues can be created for this device, driver can choose to create less
<code>enabled_queues</code>	Number	Currently enabled number of virtual queues by the driver
<code>max_queues_size</code>	Number	Maximum virtual queue depth in byte can be created for each VQ, driver can use less
<code>msix_config_vector</code>	String	MSIX vector number used by the driver for the virtio config space. 0xFFFF means that no vector is requested.
<code>mac</code>	String	The virtio-net device permanent MAC address, can be only changed from controller side via modify command
<code>link_status</code>	Number	Link status of the virtio-net device on the driver side <ul style="list-style-type: none"> • 0 – down • 1 – up
<code>max_queue_pairs</code>	Number	Number of data VQ pairs. One VQ pair has one TX queue and one RX queue. Control or admin VQ are not counted. From the host side, it appears as <code>Pre-set maximums->Combined</code> in <code>ethtool -l <virtio-dev></code> .
<code>mtu</code>	Number	The virtio-net device MTU. Default is 1500.
<code>speed</code>	Number	The virtio-net device link speed in Mb/s

Entry	Type	Description
	er	
<code>rss_max_key_size</code>	Number	The maximum supported length of the RSS key. Only applicable when <code>VIRTIO_NET_F_RSS</code> or <code>VIRTIO_NET_F_HASH_REPORT</code> is enabled.
<code>supported_hash_types</code>	Number	Supported hash types for this device in hex. Only applicable when <code>VIRTIO_NET_F_HASH_REPORT</code> is enabled: <ul style="list-style-type: none"> <code>VIRTIO_NET_HASH_TYPE_IPv4</code> (bit 0) <code>VIRTIO_NET_HASH_TYPE_TCPv4</code> (bit 1) <code>VIRTIO_NET_HASH_TYPE_UDPv4</code> (bit 2) <code>VIRTIO_NET_HASH_TYPE_IPv6</code> (bit 3) <code>VIRTIO_NET_HASH_TYPE_TCPv6</code> (bit 4) <code>VIRTIO_NET_HASH_TYPE_UDPv6</code> (bit 5)
<code>ctrl_mac</code>	String	Admin MAC address configured by driver. Not persistent with driver reload or host reboot.
<code>ctrl_mq</code>	Number	Number of queue pairs/channels configured by the driver. From the host side, it appears as <code>Current hardware settings->Combined</code> in <code>ethtool -l <virtio-dev></code> .
<code>sf_num</code>	Number	Scalable function number used for this virtio-net device
<code>sf_parent_device</code>	String	The RDMA device to use to create the SF
<code>sf_parent_device_pci_addr</code>	String	The PCIe device address (bus:device:function) to use to create the SF
<code>sf_rep_net_device</code>	String	Represents the virtio-net device

Entry	Type	Description
<code>sf_rep_net_ifindex</code>	Number	The SF representor network interface index
<code>sf_rdma_device</code>	String	The SF RDMA device interface name
<code>sf_cross_mkey</code>	Number	The cross-device MKEY created for the SF. For debug purposes only.
<code>sf_vhca_id</code>	Number	Virtual HCA identifier for the SF. For debug purposes only.
<code>rqt_num</code>	Number	The RQ table ID used for this virtio-net device. For debug purposes only.
<code>aarfs</code>	String	Whether Accelerated Receive Flow Steering configuration is enabled or disabled
<code>dim</code>	String	Whether dynamic interrupt moderation (DIM) is enabled or disabled

Example

The following is an example of a list with 1 static PF created:

```
# virtnet list
{
  "controller": {
    "emulation_manager": "mlx5_0",
    "max_hotplug_devices": "0",
    "max_virt_net_devices": "1",
    "max_virt_queues": "256",
    "max_tunnel_descriptors": "6",
```



```

"supported_features": {
  "value": "0x8b00037700ef982f",
  " 0": "VIRTIO_NET_F_CSUM",
  " 1": "VIRTIO_NET_F_GUEST_CSUM",
  " 2": "VIRTIO_NET_F_CTRL_GUEST_OFFLOADS",
  " 3": "VIRTIO_NET_F_MTU",
  " 5": "VIRTIO_NET_F_MAC",
  " 11": "VIRTIO_NET_F_HOST_TS04",
  " 12": "VIRTIO_NET_F_HOST_TS06",
  " 15": "VIRTIO_F_MRG_RX_BUFFER",
  " 16": "VIRTIO_NET_F_STATUS",
  " 17": "VIRTIO_NET_F_CTRL_VQ",
  " 18": "VIRTIO_NET_F_CTRL_RX",
  " 19": "VIRTIO_NET_F_CTRL_VLAN",
  " 21": "VIRTIO_NET_F_GUEST_ANNOUNCE",
  " 22": "VIRTIO_NET_F_MQ",
  " 23": "VIRTIO_NET_F_CTRL_MAC_ADDR",
  " 32": "VIRTIO_F_VERSION_1",
  " 33": "VIRTIO_F_IOMMU_PLATFORM",
  " 34": "VIRTIO_F_RING_PACKED",
  " 36": "VIRTIO_F_ORDER_PLATFORM",
  " 37": "VIRTIO_F_SR_IOV",
  " 38": "VIRTIO_F_NOTIFICATION_DATA",
  " 40": "VIRTIO_F_RING_RESET",
  " 41": "VIRTIO_F_ADMIN_VQ",
  " 56": "VIRTIO_NET_F_HOST_USO",
  " 57": "VIRTIO_NET_F_HASH_REPORT",
  " 59": "VIRTIO_NET_F_GUEST_HDRLEN",
  " 63": "VIRTIO_NET_F_SPEED_DUPLEX"
},
"supported_virt_queue_types": {
  "value": "0x1",
  " 0": "SPLIT"
},
"supported_event_modes": {
  "value": "0x5",

```

```

        "0": "NO_MSIX_MODE",
        "2": "MSIX_MODE"
    }
},
"devices": [
    {
        "pf_id": 0,
        "function_type": "static PF",
        "transitional": 0,
        "vuid": "MT2306XZ00BNVNETS0D0F2",
        "pci_bdf": "e2:00.2",
        "pci_vhca_id": "0x2",
        "pci_max_vfs": "0",
        "enabled_vfs": "0",
        "msix_num_pool_size": 0,
        "min_msix_num": 0,
        "max_msix_num": 256,
        "min_num_of_qp": 0,
        "max_num_of_qp": 127,
        "qp_pool_size": 0,
        "num_msix": "256",
        "num_queues": "255",
        "enabled_queues": "0",
        "max_queue_size": "256",
        "msix_config_vector": "0xFFFF",
        "mac": "16:B0:E0:41:B8:0D",
        "link_status": "1",
        "max_queue_pairs": "127",
        "mtu": "1500",
        "speed": "100000",
        "rss_max_key_size": "0",
        "supported_hash_types": "0x0",
        "ctrl_mac": "00:00:00:00:00:00",
        "ctrl_mq": "0",
        "sf_num": 1000,
        "sf_parent_device": "mlx5_0",
    }
]

```

```

    "sf_parent_device_pci_addr": "0000:03:00.0",
    "sf_rep_net_device": "en3f0pf0sf1000",
    "sf_rep_net_ifindex": 10,
    "sf_rdma_device": "mlx5_3",
    "sf_cross_mkey": "0x12642",
    "sf_vhca_id": "0x124",
    "sf_rqt_num": "0x0",
    "aarfs": "disabled",
    "dim": "disabled"
  }
]
}

```

Query

This command queries detailed information for a given device, including all VQ information if created.

Syntax

```

virtnet query [-h] {[-a] | [-p PF] [-v VF] | [-u VUID]} [--
dbg_stats] [-b] [--latency_stats] [-q QUEUE_ID] [--stats_clear]

```

Info

The options `--pf`, `--vf`, `--vuid`, and `--all` are mutually exclusive (except `--pf` and `--vf` which can be used together), but one of them must be applied.

Option	Abbr	Argument Type	Required	Description
<code>--help</code>	<code>-h</code>	N/A	No	Show the help message and exit
<code>--all</code>	<code>-a</code>	N/A	No	Query all the detailed information for all available devices. It can be time consuming if a large number of devices is available.
<code>--pf</code>	<code>-p</code>	Number	No	Unique device ID for the PF. Can be retrieved by using <code>virtnet list</code> .
<code>--vf</code>	<code>-v</code>	Number	No	Unique device ID for the VF. Can be retrieved by using <code>virtnet list</code> .
<code>--vuid</code>	<code>-u</code>	String	No	Unique device SN for the device (PF/VF). Can be retrieved by using <code>virtnet list</code> .
<code>--queue_id</code>	<code>-q</code>	Number	No	Queue index of the device VQs
<code>--brief</code>	<code>-b</code>	N/A	No	Query brief information of the device (does not print VQ information)
<code>--dbg_stats</code>	N/A	N/A	No	Print debug counters and information Note This option will be depreciated in the future.
<code>--stats_clear</code>	N/A	N/A	No	Clear all the debug counter stats Note

Option	Abbr	Argument Type	Required	Description
				This option will be depreciated in the future.

Output

Output has two main sections.

- The first section, wrapped by `devices`, are configuration and capabilities on the device level, the majority of which are the same as the `list` command. This section only covers the differences between the two.


Entry	Type	Description
<code>devices</code>	String	Entries under this section is per-device information
<code>pci_dev_id</code>	String	Virtio-net PCIe device ID. Default: 0x1041. Note This option will be depreciated in the future.
<code>pci_vendor_id</code>	String	Virtio-net PCIe vendor ID. Default: 0x1af4. Note This option will be depreciated in the future.

Entry	Type	Description
pci_class_code	String	Virtio-net PCIe device class code. Default: 0x20000. <div style="background-color: #ffffcc; padding: 10px;"> <p>Note This option will be deprecated in the future.</p> </div>
pci_subsys_id	String	Virtio-net PCIe vendor ID. Default: 0x1041. <div style="background-color: #ffffcc; padding: 10px;"> <p>Note This option will be deprecated in the future.</p> </div>
pci_subsys_vendor_id	String	Virtio-net PCIe subsystem vendor ID. Default: 0x1af4. <div style="background-color: #ffffcc; padding: 10px;"> <p>Note This option will be deprecated in the future.</p> </div>
pci_revision_id	String	Virtio-net PCIe revision ID. Default: 1. <div style="background-color: #ffffcc; padding: 10px;"> <p>Note This option will be deprecated in the future.</p> </div>

Entry	Type	Description
<code>device_features</code>	String	Enabled device feature bits according to the virtio spec. Refer to Virtio-net Feature Bits .
<code>driver_features</code>	String	Enabled driver feature bits according to the virtio spec. Valid only when the driver probes the device. Refer to Virtio-net Feature Bits .
<code>status</code>	String	Device status field bit masks according to the virtio spec: <ul style="list-style-type: none"> ◦ <code>ACKNOWLEDGE</code> (bit 0) ◦ <code>DRIVER</code> (bit 1) ◦ <code>DRIVER_OK</code> (bit 2) ◦ <code>FEATURES_OK</code> (bit 3) ◦ <code>DEVICE_NEEDS_RESET</code> (bit 6) ◦ <code>FAILED</code> (bit 7)
<code>reset</code>	Number	Shows if the current virtio-net device undergoing reset: <ul style="list-style-type: none"> ◦ 0 – not undergoing reset ◦ 1 – undergoing reset
<code>enabled</code>	Number	Shows if the current virtio-net device is enabled: <ul style="list-style-type: none"> ◦ 0 – disabled, likely FLR has occurred ◦ 1 – enabled

- The second section, wrapped by `enabled-queues-info`, provides per-VQ information:

Entry	Type	Description
<code>index</code>	Number	VQ index starting from 0 to <code>enabled_queues</code>
<code>size</code>	Number	Driver VQ depth in bytes. It is bound by device <code>max_queues_size</code> .
<code>msix_vector</code>	Number	The MSI-X vector number used for this VQ

Entry	Type	Description
	be r	
enable	Nu m be r	If current VQ is enabled or not <ul style="list-style-type: none"> o 0 – disabled o 1 – enabled
notify_offset	Nu m be r	Driver reads this to calculate the offset from start of notification structure at which this virtqueue is located
descriptor_address	Nu m be r	The physical address of the descriptor area
driver_address	Nu m be r	The physical address of the driver area
device_address	Nu m be r	The physical address of the device area
received_desc	Nu m be r	Total number of received descriptors by the device on this VQ <div style="background-color: #ffffcc; padding: 10px; margin-top: 10px;"> <p> Note This option will be depreciated in the future.</p> </div>

Entry	Type	Description
complete_desc	Number	<p>Total number of completed descriptors by the device on this VQ</p> <p>Note This option will be depreciated in the future.</p>
bad_desc_errors	Number	<p>Total number of bad descriptors received on this VQ</p> <p>Note This option will be depreciated in the future.</p>
error_cqes	Number	<p>Total number of error CQ entries on this VQ</p> <p>Note This option will be depreciated in the future.</p>
exceed_max_chain	Number	<p>Total number of chained descriptors received that exceed the maximum allowed chain by device</p> <p>Note This option will be depreciated in the future.</p>
invalid_buffer	Number	<p>Total number of times the device tried to read or write buffer that is not registered to the device</p>

Entry	Type	Description
		<p>Note This option will be depreciated in the future.</p>
batch_number	Number	<p>The number of RX descriptors for the last received packet. Relevant for BlueField-3 only.</p> <p>Note This option will be depreciated in the future.</p>
dma_q_used_number	Number	<p>The DMA q index used for this VQ. Relevant for BlueField-3 only.</p> <p>Note This option will be depreciated in the future.</p>
handler_scheduler_number	Number	<p>Scheduler number for this VQ. Relevant for BlueField-3 only.</p> <p>Note This option will be depreciated in the future.</p>

Entry	Type	Description
aux_handler_scheduler_number	Number	<p>Aux scheduler number for this VQ. Relevant for BlueField-3 only.</p> <p>Note This option will be depreciated in the future.</p>
max_post_descriptor_number	Number	<p>Maximum number of posted descriptors on this VQ. Relevant for DPA.</p> <p>Note This option will be depreciated in the future.</p>
total_bytes	Number	<p>Total number of bytes handled by this VQ. Relevant for BlueField-3 only</p> <p>Note This option will be depreciated in the future.</p>
rq_cq_max_count	Number	<p>Event generation moderation counter of the queue. Relevant for RQ.</p> <p>Note This option will be depreciated in the future.</p>
rq_cq_period	Number	<p>Event generation moderation timer for the queue in 1 μ sec granularity. Relevant for RQ.</p>

Entry	Type	Description
	boolean	<p>Note This option will be depreciated in the future.</p>
rq_cq_period_mode	Number	<p>Current period mode for RQ</p> <ul style="list-style-type: none"> 0x0 - default_mode - use device best defaults 0x1 - upon_event - queue_period timer restarts upon event generation 0x2 - upon_cqe - queue_period timer restarts upon completion generation <p>Note This option will be depreciated in the future.</p>

Example

The following is an example of querying the information of the first PF:

```
# virtnet query -p 0
{
  "devices": [
    {
      "pf_id": 0,
      "function_type": "static PF",
      "transitional": 0,
      "vuid": "MT2349X00018VNETS0D0F1",
      "pci_bdf": "23:00.1",
```

```
"pci_vhca_id": "0x1",
"pci_max_vfs": "0",
"enabled_vfs": "0",
"pci_dev_id": "0x1041",
"pci_vendor_id": "0x1af4",
"pci_class_code": "0x20000",
"pci_subsys_id": "0x1041",
"pci_subsys_vendor_id": "0x1af4",
"pci_revision_id": "1",
"device_feature": {
  "value": "0x8930032300ef182f",
  " 0": "VIRTIO_NET_F_CSUM",
  " 1": "VIRTIO_NET_F_GUEST_CSUM",
  " 2": "VIRTIO_NET_F_CTRL_GUEST_OFFLOADS",
  " 3": "VIRTIO_NET_F_MTU",
  " 5": "VIRTIO_NET_F_MAC",
  " 11": "VIRTIO_NET_F_HOST_TS04",
  " 12": "VIRTIO_NET_F_HOST_TS06",
  " 16": "VIRTIO_NET_F_STATUS",
  " 17": "VIRTIO_NET_F_CTRL_VQ",
  " 18": "VIRTIO_NET_F_CTRL_RX",
  " 19": "VIRTIO_NET_F_CTRL_VLAN",
  " 21": "VIRTIO_NET_F_GUEST_ANNOUNCE",
  " 22": "VIRTIO_NET_F_MQ",
  " 23": "VIRTIO_NET_F_CTRL_MAC_ADDR",
  " 32": "VIRTIO_F_VERSION_1",
  " 33": "VIRTIO_F_IOMMU_PLATFORM",
  " 37": "VIRTIO_F_SR_IOV",
  " 40": "VIRTIO_F_RING_RESET",
  " 41": "VIRTIO_F_ADMIN_VQ",
  " 52": "VIRTIO_NET_F_VQ_NOTF_COAL",
  " 53": "VIRTIO_NET_F_NOTF_COAL",
  " 56": "VIRTIO_NET_F_HOST_USO",
  " 59": "VIRTIO_NET_F_GUEST_HDRLEN",
  " 63": "VIRTIO_NET_F_SPEED_DUPLEX"
},
```

```

"driver_feature": {
  "value": "0x8000002300ef182f",
  " 0": "VIRTIO_NET_F_CSUM",
  " 1": "VIRTIO_NET_F_GUEST_CSUM",
  " 2": "VIRTIO_NET_F_CTRL_GUEST_OFFLOADS",
  " 3": "VIRTIO_NET_F_MTU",
  " 5": "VIRTIO_NET_F_MAC",
  " 11": "VIRTIO_NET_F_HOST_TS04",
  " 12": "VIRTIO_NET_F_HOST_TS06",
  " 16": "VIRTIO_NET_F_STATUS",
  " 17": "VIRTIO_NET_F_CTRL_VQ",
  " 18": "VIRTIO_NET_F_CTRL_RX",
  " 19": "VIRTIO_NET_F_CTRL_VLAN",
  " 21": "VIRTIO_NET_F_GUEST_ANNOUNCE",
  " 22": "VIRTIO_NET_F_MQ",
  " 23": "VIRTIO_NET_F_CTRL_MAC_ADDR",
  " 32": "VIRTIO_F_VERSION_1",
  " 33": "VIRTIO_F_IOMMU_PLATFORM",
  " 37": "VIRTIO_F_SR_IOV",
  " 63": "VIRTIO_NET_F_SPEED_DUPLEX"
},
"status": {
  "value": "0xf",
  " 0": "ACK",
  " 1": "DRIVER",
  " 2": "DRIVER_OK",
  " 3": "FEATURES_OK"
},
"reset": "0",
"enabled": "1",
"num_msix": "64",
"num_queues": "63",
"enabled_queues": "63",
"max_queue_size": "256",
"msix_config_vector": "0x0",
"mac": "4E:6A:E1:41:D8:BE",

```

```
"link_status": "1",
"max_queue_pairs": "31",
"mtu": "1500",
"speed": "200000",
"rss_max_key_size": "0",
"supported_hash_types": "0x0",
"ctrl_mac": "4E:6A:E1:41:D8:BE",
"ctrl_mq": "31",
"sf_num": 1000,
"sf_parent_device": "mlx5_0",
"sf_parent_device_pci_addr": "0000:03:00.0",
"sf_rep_net_device": "en3f0pf0sf1000",
"sf_rep_net_ifindex": 12,
"sf_rdma_device": "mlx5_2",
"sf_cross_mkey": "0xC042",
"sf_vhca_id": "0x7E8",
"sf_rqt_num": "0x0",
"aarfs": "disabled",
"dim": "disabled",
"enabled-queues-info": [
  {
    "index": "0",
    "size": "256",
    "msix_vector": "0x1",
    "enable": "1",
    "notify_offset": "0",
    "descriptor_address": "0x10cece000",
    "driver_address": "0x10cecf000",
    "device_address": "0x10cecf240",
    "received_desc": "256",
    "completed_desc": "0",
    "bad_desc_errors": "0",
    "error_cqes": "0",
    "exceed_max_chain": "0",
    "invalid_buffer": "0",
    "batch_number": "64",
```

```

        "dma_q_used_number": "6",
        "handler_schd_number": "4",
        "aux_handler_schd_number": "3",
        "max_post_desc_number": "0",
        "total_bytes": "0",
        "rq_cq_max_count": "0",
        "rq_cq_period": "0",
        "rq_cq_period_mode": "1"
    },
    .....
}
]
}
]
}

```

Stats

Tip

This command is recommended for obtaining all packet counter information. The existing packet counter information available using the `virtnet list` and `virtnet query` commands, but will be deprecated in the future.

This command retrieves the packet counters for a specified device, including detailed information for all Rx and Tx virtqueues (VQs).

To enable/disable byte wise packet counters for each Rx queue, use the following command:


```
virtnet modify {[-p PF] [-v VF]} device -pkt_cnt {enable,disable}
```

- When enabled, byte-wise packet counters are initialized to zero.
- When disabled, the previous values are retained for debugging purposes. The command will still return these old, disabled counter values.

Note

Packet counters are attached to an RQ. Thus, RQ must be created first. This means that the virtio-net device should be probed by the driver on the host OS before running the commands above.

Syntax

```
virtnet stats [-h] {[-p PF] [-v VF] | [-u VUID]} [-q QUEUE_ID]
```

Info

The options `--pf`, `--vf`, and `--vuid` are mutually exclusive (except `--pf` and `--vf` which can be used together), but one of them must be applied.

Option	Abbr	Argument Type	Required	Description
<code>--help</code>	<code>-h</code>	N/A	No	Show the help message and exit

Option	Abbr	Argument Type	Required	Description
<code>--pf</code>	<code>-p</code>	Number	No	Unique device ID for the PF. Can be retrieved by using <code>virtnet list</code> .
<code>--vf</code>	<code>-v</code>	Number	No	Unique device ID for the VF. Can be retrieved by using <code>virtnet list</code> .
<code>--vuid</code>	<code>-u</code>	String	No	Unique device SN for the device (PF/VF). Can be retrieved by using <code>virtnet list</code> .
<code>--queue_id</code>	<code>-q</code>	Number	No	Queue index of the device RQs or SQs

Output

The output has two sections.

- The first section wrapped by `device` are device details along with the packet counter statics enable state.

Entry	Type	Description
<code>device</code>	String	Entries under this section is per-device information
<code>pf_id</code>	String	Physical function ID
<code>packet_counters</code>	String	Indicates whether the packet counters feature is enabled or disabled

- The second section wrapped by `queues-stats` are information for each receive VQ.

Entry	Type	Description
<code>VQ Index</code>	Number	The VQ index starts at 0 (the first RQ) and continues up to the last SQ

Entry	Type	Description
rx_64_or_less_octet_packets	Number	The number of packets received with a size of 0 to 64 bytes. Relevant for BlueField-3 RQ.
rx_65_to_127_octet_packets	Number	The number of packets received with a size of 65 to 127 bytes. Relevant for BlueField-3 RQ.
rx_128_to_255_octet_packets	Number	The number of packets received with a size of 128 to 255 bytes. Relevant for BlueField-3 RQ.
rx_256_to_511_octet_packets	Number	The number of packets received with a size of 256 to 511 bytes. Relevant for BlueField-3 RQ.
rx_512_to_1023_octet_packets	Number	The number of packets received with a size of 512 to 1023 bytes. Relevant for BlueField-3 RQ.
rx_1024_to_1522_octet_packets	Number	The number of packets received with a size of 1024 to 1522 bytes. Relevant for BlueField-3 RQ.
rx_1523_to_2047_octet_packets	Number	The number of packets received with a size of 1523 to 2047 bytes. Relevant for BlueField-3 RQ.
rx_2048_to_4095_octet_packets	Number	The number of packets received with a size of 2048 to 4095 bytes. Relevant for BlueField-3 RQ.

Entry	Type	Description
rx_4096_to_8191_octet_packets	Number	The number of packets received with a size of 4096 to 8191 bytes. Relevant for BlueField-3 RQ.
rx_8192_to_9022_octet_packets	Number	The number of packets received with a size of 8192 to 9022 bytes. Relevant for BlueField-3 RQ.
received_desc	Number	Total number of received descriptors by the device on this VQ
complete_desc	Number	Total number of completed descriptors by the device on this VQ
bad_desc_errors	Number	Total number of bad descriptors received on this VQ
error_cqes	Number	Total number of error CQ entries on this VQ
exceed_max_chain	Number	Total number of chained descriptors received that exceed the max allowed chain by device
invalid_buffer	Number	Total number of times the device tried to read or write a buffer which is not registered to the device
batch_number	Number	The number of RX descriptors for the last received packet. Relevant for BlueField-3.

Entry	Type	Description
<code>dma_q_used_number</code>	Number	The DMA q index used for this VQ. Relevant for BlueField-3.
<code>handler_schd_number</code>	Number	Scheduler number for this VQ. Relevant for BlueField-3.
<code>aux_handler_schd_number</code>	Number	Aux scheduler number for this VQ. Relevant for BlueField-3.
<code>max_post_desc_number</code>	Number	Maximum number of posted descriptors on this VQ. Relevant for DPA.
<code>total_bytes</code>	Number	Total number of bytes handled by this VQ. Relevant for BlueField-3.
<code>rq_cq_max_count</code>	Number	Event generation moderation counter of the queue. Relevant for RQ.
<code>rq_cq_period</code>	Number	Event generation moderation timer for the queue in 1 μ sec granularity. Relevant for RQ.
<code>rq_cq_period_mode</code>	Number	Current period mode for RQ <ul style="list-style-type: none"> ◦ 0x0 - <code>default_mode</code> - use device best defaults ◦ 0x1 - <code>upon_event</code> - <code>queue_period</code> timer restarts upon event generation ◦ 0x2 - <code>upon_cqe</code> - <code>queue_period</code> timer restarts upon completion generation

Example

The following is an example of querying the packet statistics information of PF 0 and VQ 0 (i.e., RQ):

```
# virtnet stats -p 0 -q 0
{'pf': '0x0', 'queue_id': '0x0'}
{
  "device": {
    "pf_id": 0,
    "packet_counters": "Enabled",
    "queues-stats": [
      {
        "VQ Index": 0,
        "rx_64_or_less_octet_packets": 0,
        "rx_65_to_127_octet_packets": 259,
        "rx_128_to_255_octet_packets": 0,
        "rx_256_to_511_octet_packets": 0,
        "rx_512_to_1023_octet_packets": 0,
        "rx_1024_to_1522_octet_packets": 0,
        "rx_1523_to_2047_octet_packets": 0,
        "rx_2048_to_4095_octet_packets": 199,
        "rx_4096_to_8191_octet_packets": 0,
        "rx_8192_to_9022_octet_packets": 0,
        "received_desc": "4096",
        "completed_desc": "0",
        "bad_desc_errors": "0",
        "error_cqes": "0",
        "exceed_max_chain": "0",
        "invalid_buffer": "0",
        "batch_number": "64",
        "dma_q_used_number": "0",
        "handler_schd_number": "44",
        "aux_handler_schd_number": "43",
        "max_post_desc_number": "0",
```

```

        "total_bytes": "0",
        "err_handler_schd_num": "0",
        "rq_cq_max_count": "0",
        "rq_cq_period": "0",
        "rq_cq_period_mode": "1"
    }
]
}
}

```

Modify Device

This command modifies the attributes of a given device.

Syntax

```

virtnet modify [-h] [-p PF] [-v VF] [-u VUID] [-a] {device,queue}
...

```

Info

The options `--pf`, `--vf`, `--vuid`, and `--all` are mutually exclusive (except `--pf` and `--vf` which can be used together), but one of them must be applied.

Option	Abbr	Argument Type	Required	Description
<code>--help</code>	<code>-h</code>	N/A	No	Show the help message and exit

Option	Abbr	Argument Type	Required	Description
<code>--all</code>	<code>-a</code>	N/A	No	Modify all available device attributes depending on the selection of <code>device</code> or <code>queue</code>
<code>--pf</code>	<code>-p</code>	Number	No	Unique device ID for the PF. May be retrieved using <code>virtnet list</code> .
<code>--vf</code>	<code>-v</code>	Number	No	Unique device ID for the VF. May be retrieved using <code>virtnet list</code> .
<code>--vuid</code>	<code>-u</code>	String	No	Unique device SN for the device (PF/VF). May be retrieved by using <code>virtnet list</code> .
<code>device</code>	N/A	Number	No	Modify device specific options
<code>queue</code>	N/A	N/A	No	Modify queue specific options

Device Options

```
virtnet modify device [-h] [-m MAC] [-t MTU] [-e SPEED] [-l LINK]
                    [-s STATE] [-f FEATURES]
                    [-o SUPPORTED_HASH_TYPES] [-k
RSS_MAX_KEY_SIZE]
                    [-r RX_MODE] [-n MSIX_NUM] [-q
MAX_QUEUE_SIZE]
                    [-d DST_PORT] [-b RX_DMA_Q_NUM]
                    [-dc {enable,disable}] [-pkt_cnt
{enable,disable}]
                    [-aarfs {enable,disable}] [-qp
MAX_QUEUE_PAIRS] [-dim {enable,disable}]
```


Option	Abbr	Argument Type	Required	Description
<code>--help</code>	<code>-h</code>	String	No	Show the help message and exit
<code>--mac</code>	<code>-m</code>	Number	No	The virtio-net device MAC address
<code>--mtu</code>	<code>-t</code>	Number	No	The virtio-net device MTU
<code>--speed</code>	<code>-e</code>	Number	No	The virtio-net device link speed in Mb/s
<code>--link</code>	<code>-l</code>	Number	No	The virtio-net device link status <ul style="list-style-type: none"> • 0 – down • 1 – up
<code>--state</code>	<code>-s</code>	Number	No	The virtio-net device status field bit masks according to the virtio spec: <ul style="list-style-type: none"> • <code>ACKNOWLEDGE</code> (bit 0) • <code>DRIVER</code> (bit 1) • <code>DRIVER_OK</code> (bit 2) • <code>FEATURES_OK</code> (bit 3) • <code>DEVICE_NEEDS_RESET</code> (bit 6) • <code>FAILED</code> (bit 7)
<code>--features</code>	<code>-f</code>	Hex Number	No	The virtio-net device feature bits according to the virtio spec

Option	Abbr	Argument Type	Required	Description
<code>--supported-hash-types</code>	<code>-o</code>	Hex Number	No	Supported hash types for this device in hex. Only applicable when <code>VIRTIO_NET_F_HASH_REPORT</code> is enabled. <ul style="list-style-type: none"> <code>VIRTIO_NET_HASH_TYPE_IPv4</code> (bit 0) <code>VIRTIO_NET_HASH_TYPE_TCPv4</code> (bit 1) <code>VIRTIO_NET_HASH_TYPE_UDPv4</code> (bit 2) <code>VIRTIO_NET_HASH_TYPE_IPv6</code> (bit 3) <code>VIRTIO_NET_HASH_TYPE_TCPv6</code> (bit 4) <code>VIRTIO_NET_HASH_TYPE_UDPv6</code> (bit 5)
<code>--rss_max_key_size</code>	<code>-k</code>	Number	No	The maximum supported length of RSS key. Only applicable when <code>VIRTIO_NET_F_RSS</code> or <code>VIRTIO_NET_F_HASH_REPORT</code> is enabled.
<code>--rx_mode</code>	<code>-r</code>	Hex Number	No	The RX mode exposed to the driver: <ul style="list-style-type: none"> 0 – promisc 1 – all-multi 2 – all-uni 3 – no-multi 4 – no-uni 5 – no-broadcast
<code>--msix_num</code>	<code>-n</code>	Number	No	Maximum number of VQs (both data and ctrl/admin VQ). It is bound by the cap of <code>max_virt_queues</code> at the controller level (<code>virtnet list</code>).
<code>--max_queue_size</code>	<code>-q</code>	Number	No	Maximum number of buffers in the VQ. The queue size value is always a power of 2. The maximum queue size value is 32768.

Option	Abbr	Argument Type	Required	Description
<code>--max_queue_pairs</code>	<code>-qp</code>	Number	No	Number of data VQ pairs. One VQ pair has one TX queue and one RX queue. Control or admin VQs are not counted. From the host side, it appears as <code>Pre-set maximums->Combined</code> in <code>ethtool -l <virtio-dev></code> .
<code>--dst_port</code>	<code>-d</code>	Hex number	No	Modify IPv4 <code>dst_port</code> rules. Note Will be deprecated in the future.
<code>--rx_dma_q_num</code>	<code>-b</code>	Number	No	Modify max RX DMA queue number
<code>--drop_counter</code>	<code>-dc</code>	String	No	Enable/disable virtio-net drop counter
<code>--packet_counter</code>	<code>-pkt_cnt</code>	String	No	Enable/disable virtio-net device packet counter stats
<code>--aarfs_config</code>	<code>-aarfs</code>	String	No	Enable/disable auto-AARFS. Only applicable for PF devices (static PF and hotplug PF).

Option	Abbr	Argument Type	Required	Description
<code>--dim_config</code>	<code>-dim</code>	String	No	Enable/disable dynamic interrupt moderation (DIM)

(i) Note

The following `modify` options require unbinding the virtio device from virtio-net driver in the guest OS:

- `mac`
- `mtu`
- `features`
- `msix_num`
- `max_queue_size`
- `max_queue_pairs`

For example:

1. On the guest OS:

```
[host]# echo "bdf of virtio-dev" >
/sys/bus/pci/drivers/virtio-pci/unbind
```

2. On the DPU side:

1. Modify the max queue size of device:

```
[dpu]# virtnet modify -p 0 -v 0 device -q  
2048
```

2. Modify the MSI-X number of VF device:

```
[dpu]# virtnet modify -p 0 -v 0 device -n  
8
```

3. Modify the MAC address of virtio physical device ID 0 (or with its "VUID string", which can be obtained through virtnet list/query):

```
[dpu]# virtnet modify -p 0 device -m  
0C:C4:7A:FF:22:93
```

4. Modify the maximum number of queue pairs of VF device:

```
[dpu]# virtnet modify -p 0 -v 0 device -  
qp 2
```

3. On the guest OS:

```
[host]# echo "bdf of virtio-dev" >  
/sys/bus/pci/drivers/virtio-pci/bind
```

Queue Options

```
virtnet modify queue [-h] -e {event,cqe} -n PERIOD -c MAX_COUNT
```

Option	Abbr	Argument Type	Required	Description
<code>--help</code>	<code>-h</code>	String	No	Show the help message and exit
<code>--period-mode</code>	<code>-e</code>	String	No	RQ period mode: <code>event</code> or <code>cqe</code> . Default is selected by device for the best result.
<code>--period</code>	<code>-n</code>	Number	No	The event generation moderation timer for the queue in 1 μ sec granularity
<code>--max-count</code>	<code>-c</code>	Number	No	The max event generation moderation counter of the queue

Output

Entry	Type	Description
<code>errno</code>	Number	Error number: <ul style="list-style-type: none">• 0 – success• Non-0 – failed
<code>errstr</code>	String	Explanation of the error number

Example

To modify the link status of the first VF on the first PF to be down:

```
# virtnet modify -p 0 device -l 0
{'pf': '0x0', 'all': '0x0', 'subcmd': '0x0', 'link': '0x0'}
{
  "errno": 0,
  "errstr": "Success"
}
```

Log

This command manages the log level of virtio-net-controller.

Syntax

```
virtnet log [-h] -l {info,err,debug}
```

Option	Abbr	Argument Type	Required	Description
<code>--help</code>	<code>-h</code>	N/A	No	Show the help message and exit
<code>--level</code>	<code>-l</code>	String	Yes	Change the log level of <code>virtio_net_controller</code> from the journal. Default is DEBUG.

Output

Entry	Type	Description
<code>Stdout</code>	String	Success or failed with message

Example

To change the log level to info:

```
# virtnet log -l info
{'level': 'info'}
"Success"
```

To monitor current log output of the controller service with the latest 100 lines printed out:

```
$ journalctl -u virtio-net-controller -f -n 100
```

Validate

This command validates configurations of virtio-net-controller.

Syntax

```
virtnet validate [-h] -f PATH_TO_FILE
```

Option	Abbr	Argument Type	Required	Description
<code>--help</code>	<code>-h</code>	N/A	No	Show the help message and exit
<code>--file</code>	<code>-f</code>	String	No	Validate the JSON format of the <code>virtnet.conf</code> file of the <code>virtio_net_controller</code>

Output

Entry	Type	Description
Stdout	String	Success or failed with message

Example

To check if `virtnet.conf` is a valid JSON file:

```
# virtnet validate -f /opt/mellanox/mlnx_virtnet/virtnet.conf  
/opt/mellanox/mlnx_virtnet/virtnet.conf is valid
```

Version

This command prints current and updated version of virtio-net-controller.

Syntax

```
virtnet version [-h]
```

Option	Abbr	Argument Type	Required	Description
<code>--help</code>	<code>-h</code>	N/A	No	Show the help message and exit

Output

Entry	Type	Description
Original Controller	String	The original controller version

Entry	Type	Description
Destination Controller	String	The to be updated controller version

Example

Check current and next available controller version:

```
# virtnet version
[
  {
    "Original Controller": "v24.10.17"
  },
  {
    "Destination Controller": "v24.10.19"
  }
]
```

Update

This command performs a [live update](#) to another version installed on the OS. Instead of a complete shutdown and recreating all existing devices, this procedure updates to the new version with minimal down time.

Syntax

```
virtnet update [-h] [-s | -t]
```

Option	Abbr	Argument Type	Required	Description
--help	-h	N/A	No	Show the help message and exit
--start	-s	N/A	No	Start live update virtio-net-controller

Option	Abbr	Argument Type	Required	Description
<code>--status</code>	<code>-t</code>	N/A	No	Check live update status

Output

Entry	Type	Description
<code>stdout</code>	String	If the update started successfully

Example

To start the live update process, run:

```
# virtnet update -s
{'start': '0x1'}
"Update started, use 'virtnet update -t' or check logs for
status"
```

To check the update status during the update process:

```
# virtnet update -t
{'status': '0x1'}
{
  "status": "inactive",
  "last live update status": "success",
  "time_used (s)": 0.604152
}
```

Restart

This command performs a fast restart of the virtio-net-controller service. Compared to regular restart (using `systemctl restart virtio-net-controller`) this command has shorter down time per device.

Syntax

```
virtnet restart [-h]
```

Option	Abbr	Argument Type	Required	Description
<code>--help</code>	<code>-h</code>	N/A	No	Show the help message and exit

Output

Entry	Type	Description
<code>stdout</code>	String	If the fast restart finishes successfully <ul style="list-style-type: none">• SUCCESS• Failed to fast restart

Example

To start the live update process, run:

```
# virtnet restart  
SUCCESS
```

Health

This command shows health information for given devices.

i Note

The virtio-net driver must be loaded for this command to show valid information.

Syntax

```
virtnet health [-h] {[-a] | [-p PF] [-v VF] | [-u VUID]} [show]
```

i Info

The options `--pf`, `--vf`, `--vuid`, and `--all` are mutually exclusive (except `--pf` and `--vf` which can be used together), but one of them must be applied.

Option	Abbr	Argument Type	Required	Description
<code>--help</code>	<code>-h</code>	N/A	No	Show the help message and exit
<code>--all</code>	<code>-a</code>	N/A	No	Query all the detailed information for all available devices. It can be time consuming if a large number of devices is available.
<code>--pf</code>	<code>-p</code>	Number	No	Unique device ID for the PF. Can be retrieved by using <code>virtnet list</code> .

Option	Abbr	Argument Type	Required	Description
<code>--vf</code>	<code>-v</code>	Number	No	Unique device ID for the VF. Can be retrieved by using <code>virtnet list</code> .
<code>--vuid</code>	<code>-u</code>	String	No	Unique device SN for the device (PF/VF). Can be retrieved by using <code>virtnet list</code> .

Sub-command	Required	Description
<code>show</code>	Yes	Show health information for given devices

Output

Entry	Type	Description
<code>pf_id</code>	Number	Physical function ID
<code>type</code>	String	Function type: Static PF, hotplug PF, VF
<code>vuid</code>	String	Unique device SN, it can be used as an index to query/modify/unplug a device
<code>dev_status</code>	String	Device status field bit masks according to the virtio spec: <ul style="list-style-type: none"> <code>ACKNOWLEDGE</code> (bit 0) <code>DRIVER</code> (bit 1) <code>DRIVER_OK</code> (bit 2) <code>FEATURES_OK</code> (bit 3) <code>DEVICE_NEEDS_RESET</code> (bit 6) <code>FAILED</code> (bit 7)
<code>health_status</code>	String	<ul style="list-style-type: none"> Good Fatal
	Number	The number of recoveries has been performed

Entry	Type	Description
health_recover_counter		
dev_health_details	Dictionary	<p>Two types of health information are included: <code>control_plane_errors</code> and <code>data_plane_errors</code>, where <code>control_plane_errors</code> has following specific errors reported, with value either 0 or 1:</p> <ul style="list-style-type: none"> • <code>sf_rqt_update_err</code> • <code>sf_drop_create_err</code> • <code>sf_tir_create_err</code> • <code>steer_rx_domain_err</code> • <code>steer_rx_table_err</code> • <code>sf_flows_apply_err</code> • <code>aarfs_flow_init_err</code> • <code>vlan_flow_init_err</code> • <code>drop_cnt_config_err</code> <p>and <code>data_plane_errors</code> has following specific errors reported, with value either 0 or 1:</p> <ul style="list-style-type: none"> • <code>sq_stall</code> • <code>dma_q_stall</code> • <code>spurious_db_invoke</code> • <code>aux_not_invoked</code> • <code>dma_q_errors</code> • <code>host_read_errors</code> <p>Detailed descriptions of each error can be found in Health Statistics.</p>

Example

The following is an example of showing the information of the first PF:

```
# virtnet health -p 0 show
```

```

{'pf': '0x0', 'all': '0x0', 'subcmd': '0x0'}
{
  "pf_id": 0,
  "type": "static PF",
  "vuid": "MT2306XZ00BPVNETS0D0F1",
  "dev_status": {
    "value": "0xf",
    " 0": "ACK",
    " 1": "DRIVER",
    " 2": "DRIVER_OK",
    " 3": "FEATURES_OK"
  },
  "health_status": "Good",
  "health_recover_counter": 0,
  "dev_health_details": {
    "control_plane_errors": {
      "sf_rqt_update_err": 0,
      "sf_drop_create_err": 0,
      "sf_tir_create_err": 0,
      "steer_rx_domain_err": 0,
      "steer_rx_table_err": 0,
      "sf_flows_apply_err": 0,
      "aarfs_flow_init_err": 0,
      "vlan_flow_init_err": 0,
      "drop_cnt_config_err": 0
    },
    "data_plane_errors": {
      "sq_stall": 0,
      "dma_q_stall": 0,
      "spurious_db_invoke": 0,
      "aux_not_invoked": 0,
      "dma_q_errors": 0,
      "host_read_errors": 0
    }
  }
}
}

```


Error Code

CLI commands will return non-0 error code upon failure. All error numbers are negative. When there is error happening from log, it could return error number as well.

If the error number is greater than `-1000`, it's standard error. Please refer to Linux error code at [errno](#)

If the error number is less or equal `-1000`, please refer to the table below for the explanation.

Errno	Error Name	Error Description
-1000	VIRTNET_ERR_DEV_FEATURE_VALIDATE	Failed to validate device feature
-1001	VIRTNET_ERR_DEV_NOT_FOUND	Failed to find device
-1002	VIRTNET_ERR_DEV_NOT_PLUGGED	Failed - Device is not hotplugged
-1003	VIRTNET_ERR_DEV_NOT_STARTED	Failed - Device did not start
-1004	VIRTNET_ERR_DRIVER_PROBED	Failed - Virtio driver should not be loaded
-1005	VIRTNET_ERR_EPOLL_ADD	Failed to add epoll
-1006	VIRTNET_ERR_ID_OUT_OF_RANGE	Failed - ID input exceeds the max range
-1007	VIRTNET_ERR_VUID_INVALID	Failed - VUID is invalid
-1008	VIRTNET_ERR_MAC_INVALID	Failed - MAC is invalid
-1009	VIRTNET_ERR_MSIX_INVALID	Failed - MSIX is invalid

Errno	Error Name	Error Description
-1010	VIRTNET_ERR_MTU_INVALID	Failed - MTU is invalid
-1011	VIRTNET_ERR_PORT_CONTEXT_NOT_FOUND	Failed to find port context
-1012	VIRTNET_ERR_REC_CONFIG_LOAD	Failed to load config from recovery file
-1013	VIRTNET_ERR_REC_CONFIG_SAVE	Failed to save config into recovery file
-1014	VIRTNET_ERR_REC_FILE_CREATE	Failed to create recovery file
-1015	VIRTNET_ERR_REC_MAC_DELETE	Failed to delete MAC in recovery file
-1016	VIRTNET_ERR_REC_MAC_LOAD	Failed to load MAC from recovery file
-1017	VIRTNET_ERR_REC_MAC_SAVE	Failed to save MAC into recovery file
-1018	VIRTNET_ERR_REC_MQ_SAVE	Failed to save MQ into recovery file
-1019	VIRTNET_ERR_REC_PFNUM_LOAD	Failed to load PF number from recovery file
-1020	VIRTNET_ERR_REC_RX_MODE_SAVE	Failed to save RX mode into recovery file
-1021	VIRTNET_ERR_REC_SF_SAVE	Failed to save PF and SF number into recovery file
-1022	VIRTNET_ERR_REC_SFNUM_LOAD	Failed to load SF number from recovery file
-1023	VIRTNET_ERR_SF_MAC_FLOW_APPLY	Failed to apply MAC flow by SF
-1024	VIRTNET_ERR_SF_MQ_UPDATE	Failed to update MQ by SF

Errno	Error Name	Error Description
-10 25	VIRTNET_ERR_SF_RX_MODE_SET	Failed to set RX mode by SF
-10 26	VIRTNET_ERR_SNAP_NET_CTRL_OPEN	Failed to open SNAP device control
-10 27	VIRTNET_ERR_SNAP_CROSS_MKEY_CREATE	Failed to create SNAP cross mkey
-10 28	VIRTNET_ERR_SNAP_DMA_Q_CREATE	Failed to create SNAP DMA Q
-10 29	VIRTNET_ERR_SNAP_NET_DEV_QUERY	Failed to query SNAP device
-10 30	VIRTNET_ERR_SNAP_NET_DEV_MODIFY	Failed to modify SNAP device
-10 31	VIRTNET_ERR_SNAP_PF_HOTPLUG	Failed to hotplug SNAP PF
-10 32	VIRTNET_ERR_VQ_PERIOD_UPDATE	Failed to update VQ period
-10 33	VIRTNET_ERR_QUEUE_SIZE_INVALID	Failed - Queue size is invalid
-10 34	VIRTNET_ERR_SF_PORT_ADD	Failed to add SF port
-10 35	VIRTNET_ERR_WQ_WORKQUEUE_ALLOC	Failed to alloc workqueue
-10 36	VIRTNET_ERR_ETH_VQS_OPERATION_ALLOC	Failed to alloc eth VQS operation
-10 37	VIRTNET_ERR_ETH_VQS_OPERATION_COMP	Failed to complete eth VQS operation
-10 38	VIRTNET_ERR_JSON_OBJ_NOT_EXIST	Failed - JSON obj does not exist
-10 39	VIRTNET_ERR_DEV_LOAD_PREP	Failed to prepare device load

Errno	Error Name	Error Description
-1040	VIRTNET_ERR_DEV_SW_MIGRATION	Failed to sw migrate a device
-1041	VIRTNET_ERR_DEV_IS_SW_MIGRATING	Failed - Device is migrating
-1042	VIRTNET_ERR_MAX_QUEUE_SIZE	Error - queue size must be greater than 2 and is power of 2
-1043	VIRTNET_ERR_MSIX_LESS_EQUAL_THREE	Warning - this device won't function, don't try to probe with virtio driver
-1044	VIRTNET_ERR_SF_POOL_CREATING	SF pool is creating try again later
-1045	VIRTNET_ERR_DST_PORT	Failed to set dst port rule
-1046	VIRTNET_ERR_INVALID_OPTION	Option is not supported
-1047	VIRTNET_ERR_SF_CREATE	Failed to create SF
-1048	VIRTNET_ERR_DEV_SF_NUM_OUT_OF_RANGE	SF number for hotplug device should be between 2000 and 2999
-1049	VIRTNET_ERR_DEV_SF_NUM_USED	SF number is already used
-1050	VIRTNET_ERR_QUEUE_NUMBER_INVALID	Queue index is invalid
-1051	VIRTNET_ERR_SPEED_INVALID	Invalid speed please check help menu for supported link speeds
-1052	VIRTNET_ERR_SUPPORTED_HASH_TYPES_INVALID	Invalid hash types please check help menu for supported hash types
-1053	VIRTNET_ERR_RSS_MAX_KEY_SIZE_INVALID	Invalid rss max key size supported key size is 40
-1054	VIRTNET_ERR_REC_OFFLOADS_SAVE	Failed to save OFFLOADS into recovery file

Errno	Error Name	Error Description
-1055	VIRTNET_ERR_SF_OFFLOADS_UPDATE	Failed to update OFFLOADS by SF
-1056	VIRTNET_ERR_READ_LINK	Failed to readlink
-1057	VIRTNET_ERR_PATH_FORMAT	Error - Path format is invalid
-1058	VIRTNET_ERR_Q_COUNTER_ALLOC	Failed to alloc q counter
-1059	VIRTNET_ERR_REC_DIRTY_LOG_SAVE	Failed to save dirty log
-1060	VIRTNET_ERR_REC_DIRTY_LOG_DEL	Failed to delete dirty log
-1061	VIRTNET_ERR_REC_LM_STATUS_SAVE	Failed to save LM status
-1062	VIRTNET_ERR_REC_LM_STATUS_REC	Failed to found LM status record
-1063	VIRTNET_ERR_REC_DEV_MODE_SAVE	Failed to save dev mode
-1064	VIRTNET_ERR_REC_DEV_MODE_REC	Failed to found dev mode record
-1065	VIRTNET_ERR_UNPLUG_NOT_READY	Error - Device is not ready to be unplugged please check host and retry
-1066	VIRTNET_ERR_REC_MAC_TABLE_DEL	Failed to delete MAC table in recovery file
-1067	VIRTNET_ERR_REC_MAC_TABLE_LOAD	Failed to load MAC table from recovery file
-1068	VIRTNET_ERR_REC_MAC_TABLE_SAVE	Failed to save MAC table into recovery file
-1069	VIRTNET_ERR_REC_HASH_CFG_DEL	Failed to delete hash cfg in recovery file

Errno	Error Name	Error Description
-1070	VIRTNET_ERR_REC_HASH_CFG_LOAD	Failed to load hash cfg from recovery file
-1071	VIRTNET_ERR_REC_HASH_CFG_SAVE	Failed to save hash cfg into recovery file
-1072	VIRTNET_ERR_DEV_VF_GET	Failed to get VF device
-1073	VIRTNET_ERR_MAX_QUEUES_INVALID	Failed - QUEUES is invalid
-1074	VIRTNET_ERR_DEBUGFS_SAVE	Failed to save into debugfs file
-1075	VIRTNET_ERR_DEBUGFS_DEL	Failed to delete from debugfs file

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.

Trademarks

NVIDIA and the NVIDIA logo are trademarks and/or registered trademarks of NVIDIA Corporation in the U.S. and other countries. Other company and product names may be trademarks of the respective companies with which they are associated.

© Copyright 2024, NVIDIA. PDF Generated on 11/12/2024