NAME

nvmecontrol - NVM Express control utility

SYNOPSIS

nvmecontrol devlist [**-h**]
nvmecontrol identify [**-v**] [**-x**] [**-n** *nsid*] <*device-id* | *namespace-id*>
nvmecontrol perftest <**-n** *num_threads*> <**-o** *read*|*write*> [**-p**] <**-s** *size_in_bytes*> <**-t** *time_in_sec*> <*namespace-id*>
nvmecontrol reset <*device-id*>
nvmecontrol logpage <**-p** *page_id*> [**-x**] [**-v** *vendor-string*] [**-b**] [**-f** *LSP*] [**-i** *LSI*] [**-r**] <*device-id* | *namespace-id*>
nvmecontrol ns active <*device-id*>
nvmecontrol ns allocated <*device-id*>
nvmecontrol ns attach <**-n** *nsid*> <**-c** *cntid*> <*device-id*>
nvmecontrol ns attached <**-n** *nsid*> <*device-id*>
nvmecontrol ns controllers <*device-id*>
nvmecontrol ns create <**-s** *nsze*> [**-c** *ncap*] [**-f** *lbaf*] [**-m** *mset*] [**-n** *nmic*] [**-p** *pi*] [**-l** *pil*] [**-L** *flbas*] [**-d** *dps*] <*device-id*>
nvmecontrol ns delete <**-n** *nsid*> <*device-id*>
nvmecontrol ns detach <**-n** *nsid*> <**-c** *cntid*> <*device-id*>
nvmecontrol ns identify [**-v**] [**-x**] <**-n** *nsid*> <*device-id*>
nvmecontrol nsid <*device-id* | *namespace-id*>
nvmecontrol resv acquire <**-c** *crkey*> [**-p** *prkey*] <**-t** *rtype*> <**-a** *racqa*> <*namespace-id*>
nvmecontrol resv register [**-i**] [**-c** *crkey*] <**-k** *nrkey*> <**-r** *rrega*> [**-p** *cptpl*] <*namespace-id*>
nvmecontrol resv release <**-c** *crkey*> <**-t** *rtype*> <**-a** *rrela*> <*namespace-id*>
nvmecontrol resv report [**-e**] [**-v**] [**-x**] <*namespace-id*>
nvmecontrol firmware [**-s** *slot*] [**-f** *path_to_firmware*] [**-a**] <*device-id*>
nvmecontrol format [**-f** *fmt*] [**-m** *mset*] [**-o** *pi*] [**-l** *pil*] [**-E**] [**-C**] <*device-id* | *namespace-id*>
nvmecontrol sanitize <**-a** *sanact*> [**-c** *owpass*] [**-d**] [**-p** *ovrpat*] [**-r**] [**-I**] [**-U**] <*device-id*>
nvmecontrol power [**-l**] [**-p -power_state**] [**-w -workload_hint**]
nvmecontrol selftest <**-c** *code*> <*device-id* | *namespace-id*>
nvmecontrol wdc cap-diag [**-o -path_template**] <*device-id*>
nvmecontrol wdc drive-log [**-o -path_template**] <*device-id*>
nvmecontrol wdc get-crash-dump [**-o -path_template**] <*device-id*>
nvmecontrol admin-passthru [args] <*device-id*>
nvmecontrol io-passthru [args] <*namespace-id*>

DESCRIPTION

NVM Express (NVMe) is a storage protocol standard, for SSDs and other high-speed storage devices over PCI Express.

devlist

List all NVMe controllers and namespaces along with their device nodes. With the **-h** argument, use unit suffixes: Byte, Kibibyte, Mebibyte, Gibibyte, Tebibyte and Pebibyte (based on powers of 1024) when showing the disk space. By default, uses Mebibyte.

identify

The identify commands reports information from the drive’s IDENTIFY_CONTROLLER if a *device-id* is specified. It reports IDENTIFY_NAMESPACE data if a *namespace-id* is specified. When used with disk names, the IDENTIFY_NAMESPACE data is reported, unless the namespace *nsid* is overridden with the **-n** flag. Then that namespace’s data is reported, if it exists. The command accepts the following parameters:

-n The namespace <nsid> to use instead of the namespace associated with the device. A *nsid* of "0" is used to retrieve the IDENTIFY_CONTROLLER data associated with that drive.

logpage

The logpage command knows how to print log pages of various types. It also knows about vendor specific log pages from hgst/wdc, samsung and intel. Note that some vendors use the same log page numbers for different data.

Page 0x01	Drive Error Log
Page 0x02	Health/SMART Data
Page 0x03	Firmware Information
Page 0x04	Changed Namespace List
Page 0x05	Commands Supported and Effects
Page 0x06	Device Self-test
Page 0x80	Reservation Notification
Page 0x81	Sanitize Status
Page 0xc1	Advanced SMART information (WDC/HGST)
Page 0xc1	Read latency stats (Intel)
Page 0xc2	Wite latency stats (Intel)
Page 0xc5	Temperature stats (Intel)
Page 0xca	Advanced SMART information (Intel)
Page 0xca	Extended SMART information (Samsung)

Specifying **-v help** will list all valid vendors and pages. **-x** will print the page as hex. **-b** will print the binary data for the page. **-s** will set Log Specific Field. **-i** will set Log Specific Identifier. **-r** will set Retain Asynchronous Event.

ns

Various namespace management commands. If namespace management is supported by device, allow list, create and delete namespaces, list, attach and detach controllers to namespaces.

nsid

Reports the namespace id and controller device associated with the <*namespace-id*> or <*device-id*> argument.

resv acquire

Acquire or preempt namespace reservation, using specified parameters:

-a Acquire action:

0	Acquire
1	Preempt
2	Preempt and abort

-c Current reservation key.

-p Preempt reservation key.

-t Reservation type:

1	Write Exclusive
2	Exclusive Access
3	Write Exclusive - Registrants Only
4	Exclusive Access - Registrants Only
5	Write Exclusive - All Registrants
6	Exclusive Access - All Registrants

resv register

Register, unregister or replace reservation key, using specified parameters:

-c Current reservation key.

-k New reservation key.

-r Register action:

0	Register
1	Unregister
2	Replace

-i Ignore Existing Key

-p Change Persist Through Power Loss State:

0	No change to PTPL state
2	Set PTPL state to '0'. Reservations are released and registrants are cleared on a power on.
3	Set PTPL state to '1'. Reservations and registrants persist across a power loss.

resv release

Release or clear reservation, using specified parameters:

-c Current reservation key.

-t Reservation type.

-a Release action:

0	Release
1	Clean

resv report

Print reservation status, using specified parameters:

-x Print reservation status in hex.

-e Use Extended Data Structure.

format

Format either specified namespace, or all namespaces of specified controller, using specified parameters: *fmt* LBA Format, *mset* Metadata Settings, *pi* Protection Information, *pil* Protection Information Location. When formatting specific namespace, existing values are used as defaults. When formatting all namespaces, all parameters should be specified. Some controllers may not support formatting or erasing specific or all namespaces. Option **-E** enables User Data Erase during format. Option **-C** enables Cryptographic Erase during format.

sanitize

Sanitize NVM subsystem of specified controller, using specified parameters:

-a *operation*

Specify the sanitize operation to perform.

overwrite Perform an overwrite operation by writing a user supplied data pattern to the device one or more times. The pattern is given by the **-p** argument.

The number of times is given by the **-c** argument.

block
: Perform a block erase operation. All the device’s blocks are set to a vendor defined value, typically zero.

crypto
: Perform a cryptographic erase operation. The encryption keys are changed to prevent the decryption of the data.

exitfailure
: Exits a previously failed sanitize operation. A failed sanitize operation can only be exited if it was run in the unrestricted completion mode, as provided by the **-U** argument.

-c *passes*
The number of passes when performing an ‘overwrite’ operation. Valid values are between 1 and 16. The default is 1.

-d No Deallocate After Sanitize.

-I When performing an ‘overwrite’ operation, the pattern is inverted between consecutive passes.

-p *pattern*
32 bits of pattern to use when performing an ‘overwrite’ operation. The pattern is repeated as needed to fill each block.

-U Perform the sanitize in the unrestricted completion mode. If the operation fails, it can later be exited with the ‘exitfailure’ operation.

-r Run in "report only" mode. This will report status on a sanitize that is already running on the drive.

power

Manage the power modes of the NVMe controller.

-l List all supported power modes.

-p *mode*
Set the power mode to *mode*. This must be a mode listed with the

```
      nvmecontrol power -l
```

command.

-w *hint* Set the workload hint for automatic power mode control.

0	No workload hint is provided.
1	Extended idle period workload. The device is often idle for minutes at a time. A burst of write commands comes in over a period of seconds. Then the device returns to being idle.
2	Heavy sequential writes. A huge number of sequential writes will be submitted, filling the submission queues.
Other	All other values are reserved and have no standard meaning.

Please see the "NVM Subsystem Workloads" section of the relevant NVM Express Base Standard for details.

selftest

Start the specified device self-test:

-c *code* Specify the device self-test command code. Common codes are:

0x1	Start a short device self-test operation
0x2	Start an extended device self-test operation
0xe	Start a vendor specific device self-test operation
0xf	Abort the device self-test operation

wdc

The various wdc command retrieve log data from the wdc/hgst drives. The **-o** flag specifies a path template to use to output the files. Each file takes the path template (which defaults to nothing), appends the drive’s serial number and the type of dump it is followed by .bin. These logs must be sent to the vendor for analysis. This tool only provides a way to extract them.

passthru

The "admin-passthru" and "io-passthru" commands send NVMe commands to either the administrative or the data part of the device. These commands are expected to be compatible with nvme-cli. Please see the NVM Express Base Standard for details.

-o --opcode *opcode* Opcode to send.

-2 --cdw2 *value* 32-bit value for CDW2.

-3 --cdw3 *value* 32-bit value for CDW3.

-4 --cdw10 *value* 32-bit value for CDW10.

-5 --cdw11 *value* 32-bit value for CDW11.

-6 --cdw12 *value* 32-bit value for CDW12.

-7 --cdw13 *value* 32-bit value for CDW13.

-8 --cdw14 *value* 32-bit value for CDW14.

-9 --cdw15 *value* 32-bit value for CDW15.

-l --data-len Length of the data for I/O (bytes).

-m --metadata-len Length of the metadata segment for command (bytes). This is ignored and not implemented in nvme(4).

-f --flags Nvme command flags.

-n --namespace-id Namespace ID for command (Ignored).

-p --prefill Value to prefill payload with.

-b --raw-binary Output in binary format (otherwise a hex dump is produced).

-d --dry-run Do not actually execute the command, but perform sanity checks on it.

-r --read Command reads data from the device.

-s --show-command
Show all the command values on stdout.

-w --write Command writes data to the device.

Send arbitrary commands to the device. Can be used to extract vendor specific logs. Transfers to/from the device possible, but limited to MAXPHYS bytes. Commands either read data or write it, but not both. Commands needing metadata are not supported by the nvme(4) drive.

DEVICE NAMES

Where <*namespace-id*> is required, you can use either the *nvmeXnsY* device, or the disk device such as *ndaZ* or *nvdZ*. The leading */dev/* is omitted. Where <*device-id*> is required, you can use either the *nvmeX* device, or the disk device such as *ndaZ* or *nvdZ*. For commands that take an optional <nsid> you can use it to get information on other namespaces, or to query the drive itself. A <nsid> of "0" means query the drive itself.

EXAMPLES

```
nvmecontrol devlist
```

Display a list of NVMe controllers and namespaces along with their device nodes.

```
nvmecontrol identify nvme0
nvmecontrol identify -n 0 nvd0
```

Display a human-readable summary of the nvme0 IDENTIFY_CONTROLLER data. In this example, nvd0 is connected to nvme0.

```
nvmecontrol identify -x -v nvme0ns1
nvmecontrol identify -x -v -n 1 nvme0
```

Display an hexadecimal dump of the nvme0 IDENTIFY_NAMESPACE data for namespace 1.

```
nvmecontrol perftest -n 32 -o read -s 512 -t 30 nvme0ns1
```

Run a performance test on nvme0ns1 using 32 kernel threads for 30 seconds. Each thread will issue a single 512 byte read command. Results are printed to stdout when 30 seconds expires.

```
nvmecontrol reset nvme0
nvmecontrol reset nda4
```

Perform a controller-level reset of the nvme0 controller. In this example, nda4 is wired to nvme0.

```
nvmecontrol logpage -p 1 nvme0
```

Display a human-readable summary of the nvme0 controller’s Error Information Log. Log pages defined by the NVMe specification include Error Information Log (ID=1), SMART/Health Information Log (ID=2), and Firmware Slot Log (ID=3).

```
nvmecontrol logpage -p 0xc1 -v wdc nvme0
```

Display a human-readable summary of the nvme0’s wdc-specific advanced SMART data.

```
nvmecontrol logpage -p 1 -x nvme0
```

Display a hexadecimal dump of the nvme0 controller’s Error Information Log.

```
nvmecontrol logpage -p 0xcb -b nvme0 > /tmp/page-cb.bin
```

Print the contents of vendor specific page 0xcb as binary data on standard out. Redirect it to a temporary file.

```
nvmecontrol firmware -s 2 -f /tmp/nvme_firmware nvme0
```

Download the firmware image contained in "/tmp/nvme_firmware" to slot 2 of the nvme0 controller, but do not activate the image.

```
nvmecontrol firmware -s 4 -a nvme0
```

Activate the firmware in slot 4 of the nvme0 controller on the next reset.

```
nvmecontrol firmware -s 7 -f /tmp/nvme_firmware -a nvme0
```

Download the firmware image contained in "/tmp/nvme_firmware" to slot 7 of the nvme0 controller and activate it on the next reset.

```
nvmecontrol power -l nvme0
```

List all the current power modes.

```
nvmecontrol power -p 3 nvme0
```

Set the current power mode.

```
nvmecontrol power nvme0
```

Get the current power mode.

```
nvmecontrol identify -n 0 nda0
```

Identify the drive data associated with the *nda0* device. The corresponding *nvmeX* devices is used automatically.

```
nvmecontrol identify nda0
```

Get the namespace parameters associated with the *nda0* device. The corresponding *nvmeXnsY* device is used automatically.

DYNAMIC LOADING

The directories */lib/nvmecontrol* and */usr/local/lib/nvmecontrol* are scanned for any .so files. These files are loaded. The members of the *top* linker set are added to the top-level commands. The members of the *logpage* linker set are added to the logpage parsers.

SEE ALSO

The NVM Express Base Specification,
https://nvmexpress.org/wp-content/uploads/NVM-Express-1_4-2019.06.10-Ratified.pdf, June 10, 2019.

HISTORY

The **nvmecontrol** utility appeared in FreeBSD 9.2.

AUTHORS

nvmecontrol was developed by Intel and originally written by Jim Harris <*jimharris@FreeBSD.org*>.

This man page was written by Jim Harris <*jimharris@FreeBSD.org*>.