—TEMPLATE—

Admin Passthrough Command Run: “sudo nvme admin-passthru <device> –opcode=[0x] –cdw[xx]=[0x]”

NVMe-CLI Command Equivalent: “sudo nvme [COMMAND] <device> [options]”

Admin Passthrough Output:

{HEX DUMP}

NVMe-CLI Command Output:

{CLI Output}

Command works?: [Yes/No]

Command produces expected output?: [Yes/No]

Admin Passthrough Command matches NVMe-CLI Command Output?: [Yes/No]

—TEMPLATE—

Connor (pref: namespace and send and receive => try crypto for sanatize)

**CONFIRMED WORKING (getting minimum out, more to come):**

* get/set feature => Arbitration
  + Get

sudo nvme admin-passthru /dev/nvme0 --opcode=0x0A --cdw10=0x01

* + Set

sudo nvme admin-passthru /dev/nvme0 --opcode=0x09 --cdw10=0x00000001 --cdw11=<arbitration value>

* **Get logs => SMART**

sudo nvme admin-passthru /dev/nvme0 --opcode=0x02 --cdw10=0x00000002 --data-len=512

Read below for further details

# 

# 

# 

# Get/Set Features (0x09/0x0A)

**Feature ID: Arbitration**

## 

NVME cli:

**connor@connorpc**:**~**$ sudo nvme get-feature /dev/nvme0 --feature-id=1

get-feature:0x01 (Arbitration), Current value:0x00000004

**connor@connorpc**:**~**$ sudo nvme set-feature /dev/nvme0 --feature-id=1 --value=0x00010001

set-feature:0x01 (Arbitration), value:0x00010001, cdw12:00000000, save:0

**connor@connorpc**:**~**$ sudo nvme get-feature /dev/nvme0 --feature-id=1

get-feature:0x01 (Arbitration), Current value:0x00010001

**connor@connorpc**:**~**$ sudo nvme set-feature /dev/nvme0 --feature-id=1 --value=0x00000004

set-feature:0x01 (Arbitration), value:0x00000004, cdw12:00000000, save:0

**connor@connorpc**:**~**$ sudo nvme get-feature /dev/nvme0 --feature-id=1

get-feature:0x01 (Arbitration), Current value:0x00000004

**connor@connorpc**:**~**$

1. I am getting the initial value of the arbitration feature
2. overwriting it with a different value,
3. proving it changed
4. finally changing it back
5. confirming reset

Admin passthru:

**connor@connorpc**:**~**$ sudo nvme admin-passthru /dev/nvme0 --opcode=0x0A --cdw10=0x01

Admin Command Get Features is Success and result: 0x00000004

**connor@connorpc**:**~**$ sudo nvme admin-passthru /dev/nvme0 --opcode=0x09 --cdw10=0x00000001 --cdw11=0x00000005

Admin Command Set Features is Success and result: 0x00000000

**connor@connorpc**:**~**$ sudo nvme admin-passthru /dev/nvme0 --opcode=0x0A --cdw10=0x01

Admin Command Get Features is Success and result: 0x00000005

**connor@connorpc**:**~**$ sudo nvme admin-passthru /dev/nvme0 --opcode=0x09 --cdw10=0x00000001 --cdw11=0x00000004

Admin Command Set Features is Success and result: 0x00000000

**connor@connorpc**:**~**$ sudo nvme admin-passthru /dev/nvme0 --opcode=0x0A --cdw10=0x01

Admin Command Get Features is Success and result: 0x00000004

1. I am getting the initial value of the arbitration feature
2. overwriting it with a different value,
3. proving it changed
4. finally changing it back
5. Confirming reset

NOTES:

cdw10 is the feature identifier, in this case it is 01h, which maps to the arbitration feature identifier

cdw11 ***in this case*** seems to be the value it expects for *setting* the value

Because this command does not expect additional payload, data and data-len will cause it to hang (however, in the future if something might require an additional payload, I will consider this)

Status:

Command works?: yes

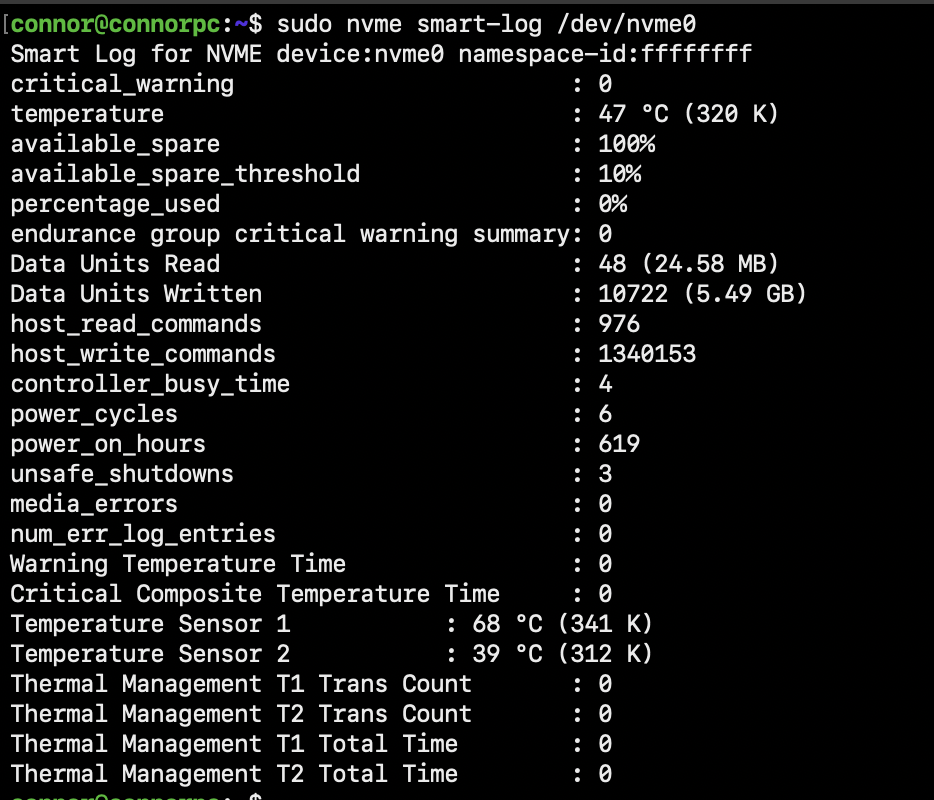
Command produces expected output?: yes

Admin Passthrough Command matches NVMe-CLI Command Output?:yes

# Get logs (0x02)

Log Page ID: SMART logs (02h)

NVME cli:



Admin Passthru:

Remember our original issue, the smart log admin passthru was not properly returning all data, HOWEVER, I stumbled upon something interesting

**connor@connorpc**:**~**$ sudo nvme admin-passthru /dev/nvme0 --opcode=0x02 --cdw10=0x00000002 --data-len=512

Admin Command Get Log Page is Success and result: 0x00000000

0 1 2 3 4 5 6 7 8 9 a b c d e f

0000: 00 40 01 64 00 00 00 00 00 00 00 00 00 00 00 00 ".@.d............"

…. ***The rest are 0s***

**connor@connorpc**:**~**$ sudo nvme admin-passthru /dev/nvme0 --opcode=0x02 --cdw10=0x00000002 --cdw11=0x01 --data-len=512

Admin Command Get Log Page is Success and result: 0x00000000

0 1 2 3 4 5 6 7 8 9 a b c d e f

0000: 00 40 01 64 0a 00 00 00 00 00 00 00 00 00 00 00 ".@.d............"

0010: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 "................"

0020: 30 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 "0..............."

0030: e2 29 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ".).............."

0040: d0 03 00 00 00 00 00 00 00 00 00 00 00 00 00 00 "................"

0050: f9 72 14 00 00 00 00 00 00 00 00 00 00 00 00 00 ".r.............."

0060: 04 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 "................"

0070: 06 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 "................"

0080: 6b 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 "k..............."

0090: 03 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 "................"

00a0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 "................"

00b0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 "................"

00c0: 00 00 00 00 00 00 00 00 55 01 38 01 00 00 00 00 "........U.8....."

… ***Again all 0s***

01f0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 "................"

Notes:

cdw10 used for log page identifier, 0x00000002 is SMART logs (02h)

cdw11 used for dwords returned, log specific identifier, (might be the solution to our problem regarding smart logs not returning enough info)

* Interestingly, 0 returns not enough info, 0x 1-3 return all info and 0x 4+ seems to error out
* intuition/instinct would tell me 0x01 means get all? While 0x00 or non specified means get minimal?

(optional formatting and references)

cdw13 upper offset

cdw14 command set ID and offset type

I am going with the assumption that all hex data numbers are little endian in the order I read them. For the sake of keeping the document to the point, I will exclude little endian hex to decimal conversion work out of confirmations below, but reference the address that stores the values according to spec.

Referencing smart log (converting the byte number as decimal count to hex count for analysis)

I confirm the following appear to match or are expected compared to nvme cli return:

* Critical warning 0x00
* temperature (in little endian) 0x02 -> 0x01
* available spare 0x03
* Available spare threshold 0x04
* Percentage used 0x05
* Endurance group summary 0x06
* Data units read 0x20 -> 0x2F
* Data units written 0x30 -> 0x3F
* Host read commands 0x40 -> 0x4F
* Host write commands 0x50 -> 0x5F
* Controller busy time 0x60-> 0x6F
* Power cycles 0x70-> 0x7F
* Power on hours 0x80 -> 0x8F
* Unsafe shutdowns 0x90->0x9F (no idea how you'd get that many unsafe shutdowns)
* Media errors 0xA0 -> 0XAF
* Num error log entries 0xB0 -> 0xBF
* warning temp time 0xC0 -> 0xC3
* Critical temp time 0xC4 -> 0xC7
* Temp sensor 1 0xC8 -> 0xC9
* Temp sensor 2 0xC10 -> 0xC11
* Thermal management T1 trans count 0xD8 -> 0xDB
* Thermal management T2 trans count 0xDC -> 0xDF
* Thermal management T1 total time 0xE0 -> 0xE3
* Thermal management T2 total time 0xE4 -> 0xE7

✅✅🥳🥳all cli and admin hex dump indicates logs match

Status:

Command works?: yes

Command produces expected output?: yes

Admin Passthrough Command matches NVMe-CLI Command Output?:yes

Charles (pref: logging)

Namespace management

Chas

**Abort**

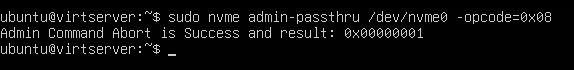
Command Used :

sudo nvme admin-passthru /dev/nvme0 –opcode=0x08

NVMe-CLI Command Equivalent:

I was unable to find the command equivalent

Admin Passthrough Output:



NVMe-CLI Command Output:

Command Works:

Command produces expected output?:

Carter

**Format NVMe**

Admin Passthrough Command Run: sudo nvme admin-passthru dev/nvme0n1 –opcode=0x80 ==data-len=0

//check progress sudo nvme get-log /dev/nvme0n1 --log-id=0x01

NVMe-CLI Command Equivalent: sudo nvme format /dev/nvme0n1 –ses=0

-ses secure erase

* 0: no secure erase
* 1: user data erase
* 2: cryptographic erase (if supported)

Admin Passthrough Output:

{HEX DUMP}

NVMe-CLI Command Output:

{CLI Output}

Command works?: No

Command produces expected output?: No

Admin Passthrough Command matches NVMe-CLI Command Output?: No