

Managing Device Power with Set Features

Using the SANBlaze **io** program to execute
Set Features 02h and Set Features 0Ch



Author:

Zachery Allen, Software Engineer
SANBlaze

Table of Contents

Introduction	2
Using the IO Program from the CLI	2
Set Features 02h - Power Management.....	3
Set Features 0Ch - Autonomous Power State Transition	4

Introduction

Many NVMe devices support multiple internal power states which can be selected by the user. Some devices provide additional functionality for configuring automatic transitions between those power states. These device capabilities are managed via the Set Features Admin Command.

In this white paper, we examine how to use the Set Features command on your SANBlaze system to configure device power. We will be using Feature Identifier 02h (Power Management) to select power states, and Feature Identifier 0Ch (Autonomous Power State Transition) to configure transition behavior.

Using the IO Program from the CLI

The Set Features command can be issued to a device by using the io program from the CLI.

Running `io -io` on the CLI will provide a list of all available commands.

Running `io -help` on the CLI will give the command structure and a list of all of the built-in flags that are available.

Commands are structured like so. The virtual file `/iportX/targetY` points to the device on the system:
`io /iportX/targetY command [flags]`

Issuing the command with the `-help` flag will show its usage and available flags.

Here we can see the `SetFeatures` command using the device at `/iport0/target100`.

```
[root@vlun]# io /iport0/target100 SetFeatures -help
io /iport(n)/target(n)lun(n) SetFeatures -nsid -fid -d
```

`SetFeatures` has the following flags available by default: `-nsid -fid -d`

There are other flags not listed which are important to know about:

`-w` is needed any time the command writes data to the drive (like `SetFeatures`).

`-v` enables verbose output, which provides details about the command and response.

`-x` enables output formatted as XML.

`-data` is used whenever a data buffer is to be sent with the command.

Note that data sent using this flag must be *big-endian*, unlike other flags which use little-endian format.

`-file` can be used in place of `-data` to send the contents of a file.

`-tt` enables usage of the *translation table*, which contains additional flags available for some commands.

The translation table can be found at `/virtualun/commands/translation_table`

For `SetFeatures` 02h and 0Ch, we will be using the flag `-CDW11` from the translation table to set the contents of Command Dword 11.

For Feature 02h (Power Management), this specifies the PS and WH fields (Note: a Workload Hint of 000b is assumed for the example below). For Feature 0Ch (Autonomous Power State Transition) this specifies the APSTE bit.

For **SetFeatures**, the **-d** flag (as seen in the results of **-help**) can also be used to specify the contents of Command Dword 11, but here we will use **-CDW11** to provide an example of translation table usage.

Set Features 02h - Power Management

Here, we use **SetFeatures** with Feature Identifier 02h (Power Management) to put the device in power state 1. Because **-CWD11** is used, the **-tt** flag is needed at the end:

```
io /iport0/target100 SetFeatures -fid 2 -CDW11 1 -tt -w -v
```

```
Allocate write buffer of 200h bytes
Issuing CDB: 09 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 02 00 00 00 01 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00
Data (showing 512 bytes)...
00000000: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000010: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000020: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
...
000001d0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000001e0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000001f0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Actual CQE/SQE:
CQE: 00000000: 00 00 00 00 00 00 00 00 00 1b 00 00 00 fd 00 01 00
SQE: 00000000: 09 00 fd 00 00 00 00 00 00 00 00 00 00 00 00 00
SQE: 00000010: 00 00 00 00 00 00 00 00 00 10 b9 e0 0f 00 00 00
SQE: 00000020: 00 00 00 00 00 00 00 00 00 02 00 00 00 01 00 00 00
SQE: 00000030: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Received CQE: 00 00 00 00 00 00 00 00 1b 00 00 00 fd 00 01 00
NVMe CQE Dword 0: 00000000
NVMe command SetFeatures succeeded, 512 bytes, NVMe Status=0h (SUCCESS) (528 usec)
Successful Write
```

If we then issue **GetFeatures**, we can see that it has been set successfully:

```
io /iport0/target100 GetFeatures -fid 2 -v
```

```
Allocate read buffer of 1000h bytes
Issuing CDB: 0a 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 02 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00
```

```

Actual CQE/SQE:
CQE: 00000000: 01 00 00 00 00 00 00 00 1e 00 00 00 43 00 01 00
SQE: 00000000: 0a 00 43 00 00 00 00 00 00 00 00 00 00 00 00 00
SQE: 00000010: 00 00 00 00 00 00 00 00 10 a2 99 0f 00 00 00
SQE: 00000020: 00 00 00 00 00 00 00 00 02 00 00 00 00 00 00 00
SQE: 00000030: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Received CQE: 01 00 00 00 00 00 00 00 1e 00 00 00 43 00 01 00
NVMe CQE Dword 0: 00000001
Data (showing 4 of 4 bytes)...
00000000: 01 00 00 00
NVMe command GetFeatures succeeded, 4 bytes, NVMe Status=0h (SUCCESS) (534 usec)
Successful Read

```

Set Features 0Ch - Autonomous Power State Transition

Here, we use **SetFeatures** with Feature Identifier 0Ch (Autonomous Power State Transition) to configure power state transition behavior. The drive being used for this example supports 5 power states; hence the five 8-byte segments of the data being sent (colored for clarity):

```

io /iport0/target100 SetFeatures -fid c -CDW11 1 -data
20d007000000000020d007000000000020d00700000000020d007000000000020d00700000000000 -tt -w
-v

```

```

Allocate write buffer of 100h bytes
Issuing CDB: 09 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0c 00 00 00 01 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00
Data (showing 256 bytes)...
00000000: 20 d0 07 00 00 00 00 00 20 d0 07 00 00 00 00 00
00000010: 20 d0 07 00 00 00 00 00 20 d0 07 00 00 00 00 00
00000020: 20 d0 07 00 00 00 00 00 00 00 00 00 00 00 00 00
00000030: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000040: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000050: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000060: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000070: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000080: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000090: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000a0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000b0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000c0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000d0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000e0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000f0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

```

Actual CQE/SQE:

CQE: 00000000: 00 00 00 00 00 00 00 00 22 00 00 00 d5 00 01 00

SQE: 00000000: 09 00 d5 00 00 00 00 00 00 00 00 00 00 00 00 00

SQE: 00000010: 00 00 00 00 00 00 00 00 00 60 8d fe 1f 00 00 00

SQE: 00000020: 00 00 00 00 00 00 00 00 0c 00 00 00 01 00 00 00

SQE: 00000030: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Received CQE: 00 00 00 00 00 00 00 00 22 00 00 00 d5 00 01 00

NVMe CQE Dword 0: 00000000

NVMe command SetFeatures succeeded, 256 bytes, NVMe Status=0h (SUCCESS) (150328 usec)

Successful Write

After setting this configuration, the drive will transition to power state 4 from all other power states after 2 seconds of idle time (Note that we are setting the APSTE field with **-CDW11 1** in the command, which enables the feature).

Each 8-byte segment of the sent data is as follows. The state to enter (ITPS, bits 07:03) is in orange, and the transition time (ITPT, bits 31:08) is in blue:

20d0070000000000

Recall that the **-data** flag is used with big-endian format.

Note that neither of these features apply to specific namespaces, which is why the **-nsid** flag was not used.