NVDIMM DSM Interface Example

V1.2
June 2016

Contents

Contents

Figures

No table of figures entries found.

Tables

1 Introduction

1.1 Document Scope

This document is targeted to writers of BIOS and OS drivers for NVDIMMs whose design adheres to the NFIT Tables in the ACPI V6.1 specification. This document specifically discusses the NVDIMM Device Specific Method (_DSM) example.

1.2 Related Documents

The related documents are ACPI Specification Version 6.1 (http://www.uefi.org/specifications) and NVDIMM Namespace Specification (http://pmem.io/documents).

While the V6.1 ACPI specification describes the root device _DSMs that are supported by the NVDIMM, this specification describes the additional NVDIMM specific leaf node _DSM methods that are supported by the NVDIMM that report the JEDEC NFIT Region Format Interface Code 0x0201 and 0x0301.

1.3 Terminology

Refer to Table 1 for definitions of terms used in this document.

Table 1 – Terminology

Term	Description
NFIT	The NVDIMM Firmware Interface Table defines the ACPI-like information created by the BIOS to inform the OS about NVDIMMs in the system.
NVDIMM	Non-volatile memory in a DIMM form factor.
NVDIMM Namespace Label	Labels, stored at a known location on NVDIMMs, which define the DIMM's contribution to NVDIMM Namespaces. This is a software mechanism; the DIMM itself just sees the labels as part of the overall data stored on the DIMM.
NVDIMM Namespace	Similar to an NVMe Namespace or a Logical Unit (LUN) on a SCSI disk, this is a software mechanism for managing ranges of persistence on NVDIMMs.
Persistent Memory	Byte-addressable memory that retains its contents after power loss.
SPA	System Physical Address. A physical address on the host operating system.

2 NVDIMM Device Specific Method (DSM)

ACPI defines an NVDIMM root device under _SB scope with a _HID of "ACPI0012". The NVDIMM child devices under the NVDIMM root device are defined with _ADR corresponding to the NFIT device handle. The NVDIMM root device and the NVDIMM devices can have device specific methods (_DSM) to provide additional functions specific to a particular NVDIMM implementation.

An example name space is shown below for a platform containing one NVDIMM:

```
Scope (\_SB){
    Device (NVDR) // Root device
    {
        Name (_HID, "ACPI0012")
        Method (_STA) {…}
        Method (_FIT) {…}
        Method (_DSM, …) {…}

        Device (NVD)
        {
            Name(_ADR, h) //where h is NFIT Device Handle for this NVDIMM
            Method (_DSM, …) {…}
        }
    }
}
```

The ACPI 6.1 specification ~~chapter 2 in this document~~ describes ~~the an example~~ _DSM interfaces for NVDIMM Root Device and the chapter **Error! Reference source not found.** in this document describes ~~an example~~leaf node _DSM interfaces for NVDIMM Device with Region Format Interface Code (RFIC) of 0x0201 or 0x0301.

3 _DSM Interface for NVDIMM Device (non-root) - Example

Platforms that have the _DSM interface implemented, as outlined in this section, can support a NVDIMM region with Region Format Interface Code (RFIC) of 0x0201 or 0x0301.

Note that the _DSM methods defined in this section are required to be implemented under NVDIMM devices that are child devices of NVDIMM objects associated with _HID of ACPI0012 in ACPI name space hierarchy.

Arg0 – UUID (set to 4309AC30-0D11-11E4-9191-0800200C9A66)

Arg1 – Revision ID (set to 1)

Arg2 – Function Index

0 – Query command implemented per ACPI Specification

1 – SMART and Health Info

2 – Get SMART Threshold

3 – Get Block NVDIMM Flags

4 – Get Namespace Label Size

5 – Get Namespace Label Data

6 - Set Namespace Label Data

7 - Get Vendor-Specific Command Effect Log Size

8 - Get Vendor-Specific Command Effect Log

9 – Vendor-Specific Command

Arg3 – A package containing parameters for the function specified by the *UUID, Revision ID, and Function Index.* The layout of the package for each command along with the corresponding output are illustrated in the respective *Function Index* description sections. For DSM functions that take an input argument, Arg3 is a package containing a Buffer, list of bytes, value. The output of all functions in the DSM is a Buffer, list of bytes, value.

Implementation Note: This section adopts the following conventions for the _DSM function return status codes. This status can always be utilized for the status of each _DSM function, whether the specific status value is defined in the output buffer or not:

Bytes[1-0]

0 – Success

1 – Failure - Function Not Supported

2 – Failure - Non-Existing Memory Device

3 – Failure - Invalid Input Parameters

4 – Failure – HW Error

5 – Failure – Retry Suggested

6 – Failure – Unknown Reason

7 – Vendor Specific Error (details in Extended Status Field)

8-FFFFh Reserved

Bytes[3-2] Extended Status Field (Vendor defined)

3.1 SMART and Health Info (Function Index 1)

This function provides information for the SMART and Health function.

3.1.1 Input (Arg3)

None

3.1.2 Output

The return value for this function is a buffer formatted as shown in Table 3-1.

Table 3-1 SMART and Health Info – Output Format

Field	Byte Length	Byte Offset	Description
Status	4	0	Bytes[1-0] 0 – Success 1 – Not Supported

			2 – Non-Existing Memory Device 3 - Reserved 4 – SMART and Health Info failed due to a HW error 6 – SMART and Health Info failed 7 – Vendor-specific Error(details in Extended Status Field) 8-FFFFh Reserved Bytes[3-2] Extended Status Field (Vendor Defined)
SMART and Health Data	128	4	Output formatted as shown in Table 3-2.

Table 3-2 SMART and Health Data Format

Bytes	**Description**
03-00	Validation Flags – if the corresponding validation flag is not set in this field, it is indication to software that the corresponding field is not valid and must not be interpreted. Bit[0] – if set to 1, indicates that Health Status field is valid Bit[1] – if set to 1, indicates that Spare Blocks field is valid Bit[2] – if set to 1, indicates that Percentage Used field is valid Bit[3] – if set to 1, indicates that Current NVDIMM Media Temperature field is valid Bit[4] – if set to 1, indicates that Current NVDIMM Controller Temperature field is valid Bits[8:5] – Reserved Bit[9] – if set to 1, indicates that Alarm Trips field is valid Bit[10] – if set to 1, indicates that Last Shutdown Status field is valid Bit[11] – if set to 1, indicates that Size of Vendor-specific Data field is valid. If this field is not valid, the software will ignore the vendor-specific data fields. Bits[31:12] – Reserved

07-04	Reserved
08	Health Status (HS): Overall health summary Bit[0] – Non-Critical - if set to 1, indicates a non-critical condition, maintenance required but no data loss detected Bit[1] – Critical - if set to 1, indicates Critical condition, features or performance degraded due to failures but no data loss detected Bit[2] – Fatal - if set to 1, indicates fatal condition, data loss is detected or is imminent Bits[7:3] - Reserved
09	Spare Blocks: Remaining Spare Capacity as % of factory configured space Valid range 0 to 100. 0 = All of the factory configured spare block capacity has been utilized 100 = None of the factory configured spare block capacity has been utilized
10	Percentage Used: Device life span as percentage, 100 = the warranted life span of the device has been reached
11	Alarm Trips: Bits to signify if values have tripped their respective alarm thresholds Bit[0] - Spare Blocks Trip - If set then the spare block value has reached the pre-programmed threshold limit Bit[1] – NVDIMM Media Temperature Trip - If set then the NVDIMM Media temperature value has reached the pre-programmed threshold limit Bit[2] – NVDIMM Controller Temperature Trip - If set then the NVDIMM Controller temperature value has reached the pre-programmed threshold limit Bits[7:3] - Reserved
13-12	Current NVDIMM Media Temperature: Current temperature of the NVDIMM Media Bits[14:0] - Temperature in 1/16th Celsius resolution. Bit[15] – Sign bit for temperature (1 = negative, 0 = positive)
15-14	Current NVDIMM Controller Temperature: Current temperature of the NVDIMM Controller Bits[14:0] - Temperature in 1/16th Celsius resolution. Bit[15] – Sign bit for temperature (1 = negative, 0 = positive)

30-16	Reserved
31	Last Shutdown Status: status of last shutdown 0 – Clean shutdown 1 - 0FFh – Not Clean Shutdown, indicates that there was either a platform or memory device-related failure occurred when saving data targeted for this memory device.
35-32	Size of Vendor-specific Data. If set to 0, indicates that there is no vendor specific data that follows. Otherwise, indicates size of the Vendor-specific data that follows.
127-36	Vendor-specific Data

3.2 Get SMART Threshold (Function Index 2)

This function provides SMART related threshold information.

3.2.1 Input (Arg3)

None

3.2.2 Output

The return value for this function is a buffer formatted as shown in Table 3-3.

Table 3-3 Get SMART Threshold – Output Format

Field	Byte Length	Byte Offset	Description
Status	4	0	Bytes[1-0] 0 – Success 1 – Not Supported 2 – Non-Existing Memory Device 3 – Reserved 4 – Vendor-specific Error(details in Extended Status Field) 5–FFFFh Reserved

			Bytes[3-2] Extended Status Field (Vendor-defined)
SMART Threshold Data	8	4	Output formatted as shown in Table 3-4.

Table 3-4 SMART Threshold Data Format

Bytes	**Description**
0	Threshold Alarm Control – If this bit is set to 1, the specific alarm is enabled and the corresponding Alarm Trip bit in the SMART Health Status output payload will be set when a specific threshold outlined below has been reached. Bit[0] – Temperature Threshold Alarm Bit[1] – Spare Block Threshold Alarm Bits[7:2] – Reserved
1	Reserved
3-2	Temperature Threshold Bits[14:0] – Temperature in 1/16th Celsius resolution. Bit[15] – Sign bit for temperature (1 = negative, 0 = positive) If the *Temperature Threshold Alarm* bit is enabled and when the temperature goes above this value, the *Temperature Trip* bit will be set in the SMART and Health Data structure defined in Table 3-2.
4	Spare Block Threshold: Remaining spare capacity as % of factory configured space. Valid range 0 to 100. If the *Spare Block Threshold Alarm* bit is enabled and when the space block capacity goes below this threshold, the *Spare Blocks Trip* bit will be set in the SMART and Health Data structure defined in Table 3-2.
7-5	Reserved

3.3 Get Block NVDIMM Flags (Function Index 3)

This function that is only applicable if block mode is enabled in the NVDIMM (i.e., the Number of Block Control Windows field set is set to a non-zero value in the NVDIMM Control Region Structure).

3.3.1 Input (Arg3)

None

3.3.2 Output

The return value for this function is a buffer formatted as shown in Table 3-5.

Table 3-5 Get Block NVDIMM Flags - Output Format

Field	Byte Length	Byte Offset	Description
Status	4	0	Bytes[1-0] 0 – Success 1 – Not Supported 2 – Non-Existing Memory Device 3 – Reserved 6 – Get Block NVDIMM Flags failed 7 – Vendor-specific Error(details in Extended Status Field) 8-FFFFh Reserved Bytes[3-2] Extended Status Field (Vendor-defined)
NVDIMM Flags	4	4	Byte[0] Bit[0] – Block Data Window Invalidation Required – If this bit is set to 1, indicates that the NVDIMM requires the driver to flush previous data from cache lines that will be moved through the Block Data Window, before re-using the Block Data Window for read. If set to '0', flushing of previous data from cachelines that will be moved through the Block Data Window are handled by the

			platform or VMM. Typical usage of this flag is in a virtualized environment. Bit[1] – Command Register in Block Control Window Latch – If this bit is set to 1, indicates that after a write to the Command Register in Block Control Windows, the NVDIMM requires the software to read the same Command Register to ensure that the command is latched before reading contents from Block Data Window. If this bit is set to 0, software is allowed to read the contents of the Block Data Window immediately after writing to the Command Register of Block Control Window. Bits[7:2] – Reserved Note: If this command is not implemented, then the software should assume bit[0] and bit[1] are clear. Bytes[3-1] – Reserved

3.4 Get Namespace Label Size (Function Index 4)

The usage of this function is detailed in *NVDIMM Namespace Specification*.

3.4.1 Input (Arg3)

None

3.4.2 Output

The return value for this function is a buffer formatted as shown in Table 3-6.

Table 3-6 Get Namespace Label Size – Output Format

Field	Byte Length	Byte Offset	Description
Status	4	0	Bytes[1-0] 0 – Success 1 – Not Supported 2 – Non-Existing Memory Device 3 - Reserved 6 – Get Namespace Label Size failed 7 – Vendor-specific Error(details in Extended Status Field) 8-FFFFh Reserved Bytes[3-2] Extended Status Field (Vendor-defined)
Size of Namespace Label Area	4	4	Size returned in bytes
Max Namespace Label Data Length	4	8	In bytes, Maximum size of the namespace label data length supported by the platform in *Get/Set Namespace Label Data* functions

3.5 Get Namespace Label Data (Function Index 5)

The usage of this function is detailed in *NVDIMM Namespace Specification*.

3.5.1 Input (Arg3)

The input is a package containing a single buffer, where the buffer is formatted as shown in Table 3-7.

Table 3-7 Get Namespace Label Data – Input Format

Field	Byte Length	Byte Offset	Description
Offset	4	0	In bytes

			Indicates the offset in the namespace label data area, to which the namespace label data is to be read from the target NVDIMM
Length	4	4	In bytes

3.5.2 Output

The return value for this function is a buffer formatted as shown in Table 3-8.

Table 3-8 Get Namespace Label Data – Output Format

Field	Byte Length	Byte Offset	Description
Status	4	0	Bytes[1-0] 0 – Success 1 – Not Supported 2 – Non-Existing Memory Device 3 – Invalid Input Parameters(Offset + Length is > size of Namespace Label Data Area) 4 – Get Namespace Label Data failed due to a HW error 6 – Get Namespace Label Data failed 7 – Vendor-specific Error(details in Extended Status Field) 8-FFFFh Reserved Bytes[3-2] Extended Status Field (Vendor-defined)
Namespace Label Data	Varies	4	The size of the output is equal to input's *Length* if *Status* is Success; otherwise, the contents of rest of the output buffer are not valid.

3.6 Set Namespace Label Data (Function Index 6)

The usage of this function is detailed in *NVDIMM Namespace Specification*.

3.6.1 Input (Arg3)

Input is a package containing a single buffer, where the buffer is formatted as shown in Table 3-9.

Table 3-9 Set Namespace Label Data – Input Format

Field	**Byte Length**	**Byte Offset**	**Description**
Offset	4	0	In bytes Indicates the offset in the namespace label data area, to which the *Namespace Label Data* is to be written to the target NVDIMM
Length	4	4	In bytes
Namespace Label Data	Varies	8	Namespace label data. Size of the namespace label data is as indicated by *Length* field above.

3.6.2 Output

The return value for this function is a buffer formatted as shown in Table 3-10.

Table 3-10 Set Namespace Label Data – Output Format

Field	**Byte Length**	**Byte Offset**	**Description**
Status	4	0	Bytes[1-0] 0 – Success 1 – Not Supported 2 – Non-Existing Memory Device 3 – Invalid Input Parameters(Offset + Length is > size of Namespace Label Data Area) 4 – Set Namespace Label Data failed due to a HW error

			6 – Set Namespace Label Data failed 7 – Vendor-specific Error(details in Extended Status Field) 8-FFFFh Reserved Bytes[3-2] Extended Status Field (Vendor-defined)

3.7 Get Vendor-Specific Command Effect Log Size (Function Index 7)

This function returns the maximum data size of output buffer for retrieving the Vendor-Specific Command Effect Log.

3.7.1 Input (Arg3)

None

3.7.2 Output

The return value for this function is a buffer formatted as shown in Table 3-11.

Table 3-11 Get Vendor Specific Command Effect Log Size – Output Format

Field	Byte Length	Byte Offset	Description
Status	4	0	Bytes[1-0] 0 – Success 1 – Not Supported 2 – Non-Existing Memory Device 3 - Reserved 6 - Get Vendor Specific Command Effect Log Size failed 7 – Vendor-specific Error(details in Extended Status Field) 8-FFFFh Reserved

			Bytes[3-2] - Extended Status Field (Vendor-defined)
Max Command Effect Log Data Length	4	8	In bytes, Maximum size of the Vendor-specific command effect log data buffer supported by the platform

3.8 Get Vendor-Specific Command Effect Log (Function Index 8)

This function returns the Command Effect Log for all of the Vendor-Specific Commands. If the OpCode is not in the Command Effect Log, OSPM may block the Vendor-Specific call for that OpCode.

3.8.1 Input (Arg3)

None

3.8.2 Output

The return value for this function is a buffer formatted as shown in Table 3-12.

Table 3-12 Get Vendor Specific Command Effect Log Size – Output Format

Field	Byte Length	Byte Offset	Description
Status	4	0	Bytes[1-0] 0 – Success 1 – Not Supported 2 – Non-Existing Memory Device 3 - Reserved

			4 - Get Vendor Specific Command Effect Log failed due to a HW error 6 - Get Vendor Specific Command Effect Log failed 7 – Vendor-specific Error(details in Extended Status Field) 8-FFFFh Reserved Bytes[3-2] Extended Status Field (Vendor-defined)
OpCode Count	2	4	Number of OpCode command effect logs returned
Reserved	2	6	
Command Effect Data	Varies	8	The command effect data for each OpCode. The Fields in Table 3-13 are repeated *OpCode Count* times.

Table 3-13 Command Effect Data - Format

Field	Byte Length	Byte Offset	Description
OpCode	4	0	OpCode representing a Vendor-specific command
OpCode Command Effect	4	4	Bit[0] – No Effects (NE) If set to 1, execution of this OpCode does not change DIMM state. If this bit is set, all the following bits should be clear. Bit[1] – Security State Change (SSC) If set to 1, execution of this Opcode results in immediate security state change of the NVDIMM.

			Bit[2] – DIMM Configuration Change after Reboot (DCC) If set to 1, execution of this Opcode results in change to the configuration of the NVDIMM or data contained within persistent memory regions of the NVDIMM. The change does not take effect until the system reboots. Bit[3] – Immediate DIMM Configuration Change (IDCC) If set to 1, execution of this Opcode results in immediate change to the configuration of the NVDIMM or data contained within persistent memory regions of the NVDIMM. Bit[4] – Quiesce All IO (QIO) If set to 1, execution of this Opcode may disrupt on-going operations of the memory region covered by this NVDIMM. The outstanding IO operations corresponding to this NVDIMM must be quiesced before executing this command; otherwise, undefined system behavior will result. Bit[5] - Immediate DIMM Data Change (IDDC) If set to 1, execution of this Opcode results in immediate change to the data written to the NVDIMM. Bit[6] – Test Mode (TM) If set to 1, execution of this Opcode activates a test feature that may disrupt on-going operations. This may result in errors or error recovery operations.

			Bit[7] – Debug Mode (DM) If set to 1, execution of this Opcode activates a debug feature that is non-disruptive, but may alter performance characteristics of the NVDIMM. Bits[31:8] – Reserved

3.9 Vendor-Specific Command (Function Index 9)

This function provides access to the vender specific commands. Refer to the vendor specific document for the format of the input and output data buffers.

3.9.1 Input (Arg3)

Input is a package containing a single buffer, where the buffer is formatted as shown in Table 3-14.

Table 3-14 Vendor Specific Command – Input Format

Field	Byte Length	Byte Offset	Description
OpCode	4	0	Vendor-specific command OpCode
OpCode Parameters Data Length	4	4	In bytes Length of OpCode parameters data
OpCode Parameters Data	Varies	8	Vendor-specific command input data

3.9.2 Output

The return value for this function is a buffer formatted as shown in Table 3-15.

Table 3-15 Vendor Specific Command – Output Format

Field	**Byte Length**	**Byte Offset**	**Description**
Status	4	0	Bytes[1-0] 0 – Success 1 – Not Supported 2 – Non-Existing Memory Device 3 – Invalid Input Parameters 4 – Vendor Specific Command failed due to a HW error 5 – Vendor Specific Command failed - Retry Suggested 6 – Vendor Specific Command failed 7 – Vendor-specific Error(details in Extended Status Field) 8-FFFFh Reserved Bytes[3-2] Extended Status Field (Vendor-defined)
Output Data Length	4	4	In bytes. If *Status* is not *Success,* output data length returned is 0.
Output Data	Varies	8	The *Output Data* is valid only when the *Output Data Length* is non-zero.