NVDIMM DSM Interface Example

April 2015

Notices

This document contains information on products, services and/or processes in development. All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest forecast, schedule, specifications and roadmaps.

The products and services described may contain defects or errors known as errata which may cause deviations from published specifications. Current characterized errata are available on request.

Copies of documents which have an order number and are referenced in this document may be obtained by calling 1-800-548-4725 or by visiting www.intel.com/design/literature.htm.

Contents

Contents

1 Introduction 5
1.1 Document Scope 5
1.2 Related Documents 5
1.3 Terminology 5
2 NVDIMM Device Specific Method (DSM) 6
3 _DSM Interface for NVDIMM Root Device - Example 7
3.1 Address Range Scrubbing 7
3.1.1 Query ARS Capabilities (Function Index 1) 8
3.1.2 Start ARS (Function Index 2) 8
3.1.3 Query ARS Status (Function Index 3) 9
4 _DSM Interface for NVDIMM Device (non-root) - Example 13
4.1 SMART and Health Info (Function Index 1) 14
4.1.1 Input (Arg3) 14
4.1.2 Output 14
4.2 Get SMART Threshold (Function Index 2) 15
4.2.1 Input (Arg3) 15
4.2.2 Output 16
4.3 Get Block NVDIMM Flags (Function Index 3) 17
4.3.1 Input (Arg3) 17
4.3.2 Output 17
4.4 Get Namespace Label Size (Function Index 4) 18
4.4.1 Input (Arg3) 18
4.4.2 Output 18
4.5 Get Namespace Label Data (Function Index 5) 19
4.5.1 Input (Arg3) 19
4.5.2 Output 19
4.6 Set Namespace Label Data (Function Index 6) 20
4.6.1 Input (Arg3) 20
4.6.2 Output 21
4.7 Get Vendor-Specific Command Effect Log Size (Function Index 7) 21
4.7.1 Input (Arg3) 21
4.7.2 Output 21
4.8 Get Vendor-Specific Command Effect Log (Function Index 8) 22
4.8.1 Input (Arg3) 22
4.8.2 Output 22
4.9 Vendor-Specific Command (Function Index 9) 25
4.9.1 Input (Arg3) 25

4.9.2 Output 25

Figures

Figure 3-1 Format of ARS Output buffer 11
Figure 3-2 Format of ARS Error Record 12

Tables

Table 1 – Terminology 5
Table 3-1 Query ARS Capabilities – Input Format 8
Table 3-2 Query ARS Capabilities – Output Format 8
Table 3-3 Start ARS – Input 9
Table 3-4 Start ARS – Output Format 9
Table 3-5 Query ARS Status – Output Format 10
Table 3-6 Fields of ARS Output Buffer Format 11
Table 3-7 ARS Error Record Format 12
Table 4-1 SMART and Health Info – Output Format 14
Table 4-2 SMART and Health Data Format 14
Table 4-3 Get SMART Threshold – Output Format 16
Table 4-4 SMART Threshold Data Format 16
Table 4-5 Get Block NVDIMM Flags - Output Format 17
Table 4-6 Get Namespace Label Size – Output Format 18
Table 4-7 Get Namespace Label Data – Input Format 19
Table 4-8 Get Namespace Label Data – Output Format 19
Table 4-9 Set Namespace Label Data – Input Format 20
Table 4-10 Set Namespace Label Data – Output Format 21
Table 4-11 Get Vendor Specific Command Effect Log Size – Output Format 21
Table 4-12 Get Vendor Specific Command Effect Log Size – Output Format 22
Table 4-13 Command Effect Data - Format 23
Table 4-14 Vendor Specific Command – Input Format 25
Table 4-15 Vendor Specific Command – Output Format 25

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 V6.0 specification. This document specifically discusses the NVDIMM Device Specific Method (_DSM) example.

1.2 Related Documents

The related documents are ACPI Specification Version 6.0 (http://www.uefi.org/specifications) and NVDIMM Namespace Specification (http://pmem.io/documents).

1.3 Terminology

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

Table 1 – Terminology

Term	Description

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 chapter 2 in this document describes an example _DSM interface for NVDIMM Root Device and the chapter 3 in this document describes an example _DSM interface for NVDIMM Device with Region Format Interface Code (RFIC) of 0x0201.

3 _DSM Interface for NVDIMM Root Device - Example

This chapter describes the device specific method (_DSM) for NVDIMM root devices is described in this section. Note that the _DSM methods defined in this section are required to be implemented only under NVDIMM object with _HID of ACPI0012.

Arg0 *UUID {set to 2f10e7a4-9e91-11e4-89d3-123b93f75cba}*
Arg1 *Revision ID (set to 1)*
Arg2 *Function Index*
0 – Query command implemented per ACPI Specification
1 – Query Address Range Scrub (ARS) Capabilities
2 – Start Address Range Scrub (ARS)
3 – Query Address Range Scrub (ARS) Status
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. The input and output package are a lists of bytes (Buffer).

Implementation Note: This section adopts the following conventions for the _DSM function return status codes:
Bytes[1-0]
0 *Success*
1 *Not Supported*
2 *Invalid Input Parameters*
3 *Function-Specific Error Code*
4 - FFFFh Reserved

Bytes[3-2] Extended Status Field (Function Specific Status Code on Success)

3.1 Address Range Scrubbing

Address Range Scrubbing (ARS) allows platform to communicate persistent memory errors to system software. This capability allows system software to avoid accessing addresses with uncorrectable errors in persistent memory.

The address range scrubbing command is accessed via _DSM interface. The ARS command is system scope and is not specific to a single NVDIMM, i.e., it returns the locations detected to be in error for all the NVDIMMs present in the system.

3.1.1 Query ARS Capabilities (Function Index 1)

This function provides ARS capabilities for a given address range. Its input (Arg3) and the output parameter format are as follows:

3.1.1.1.1 Input (Arg3)

Input is a package containing a single buffer, where the buffer is formatted as shown in Table 3-1.

Table 3-1 Query ARS Capabilities – Input Format

Field	Byte Length	Byte Offset	Description

3.1.1.1.2 Output

The return value for this function is a buffer formatted as shown in Table 3-2.

Table 3-2 Query ARS Capabilities – Output Format

Field	Byte Length	Byte Offset	Description

3.1.2 Start ARS (Function Index 2)

The Start ARS function triggers an Address Range Scrub for the given range of memory. Address scrubbing can be done for volatile memory, persistent memory, or both. Only one scrub can be in progress system wide at any given time. You must first issue a Query ARS Status command and ensure no ARS is in progress before issuing a Start ARS function. When an address range scrub operation is started, the previous ARS data is lost. You must call the Query ARS Status function to retrieve any existing ARS data before calling the Start ARS function.

3.1.2.1 Input (Arg3)

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

Table 3-3 Start ARS – Input

Field	Byte Length	Byte Offset	Description

3.1.2.2 Output

The return value for this function is a buffer formatted as shown in Table 3-4.

Table 3-4 Start ARS – Output Format

Field	Byte Length

3.1.3.1 Input (Arg3)

None

3.1.3.2 Output

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

Table 3-5 Query ARS Status – Output Format

Field	Byte Length	Byte Offset	Description

The output SPA range return indicates the scope of the ARS scrub for the specified type.

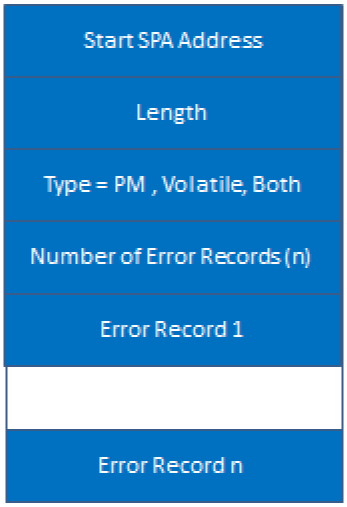

Figure 3-1 Format of ARS Output buffer

Table 3-6 Fields of ARS Output Buffer Format

Field	Byte Length	Byte Offset	Interpretation

Figure 3-2 Format of ARS Error Record

Table 3-7 ARS Error Record Format

Field	Byte Length	Byte Offset	Interpretation

U

U

4 _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.
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. The input and output package are a list of bytes (Buffer).

Implementation Note: This section adopts the following conventions for the _DSM function return status codes:
Bytes[1-0]
0 Success
1 Not Supported
2 Non-Existing Memory Device
3 Invalid Input Parameters
4 Vendor Specific Error (details in Extended Status Field)
5-FFFFh Reserved

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

4.1 SMART and Health Info (Function Index 1)

This function provides information for the SMART and Health function.

4.1.1 Input (Arg3)

None

4.1.2 Output

The return value for this function is a buffer formatted as shown in Table 4-1.

Table 4-1 SMART and Health Info – Output Format

Field	Byte Length	Byte Offset	Description

Table 4-2 SMART and Health Data Format

Bytes	Description

4.2 Get SMART Threshold (Function Index 2)

This function provides SMART related threshold information.

4.2.1 Input (Arg3)

None

4.2.2 Output

The return value for this function is a buffer formatted as shown in Table 4-3.

Table 4-3 Get SMART Threshold – Output Format

Field	Byte Length	Byte Offset	Description

Table 4-4 SMART Threshold Data Format

Bytes	Description
	h U m U h *h U*
	U m U h *U m*

4.3 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).

4.3.1 Input (Arg3)

None

4.3.2 Output

The return value for this function is a buffer formatted as shown in Table 4-5.

Table 4-5 Get Block NVDIMM Flags - Output Format

Field	Byte Length	Byte Offset	Description

4.4 Get Namespace Label Size (Function Index 4)

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

4.4.1 Input (Arg3)

None

4.4.2 Output

The return value for this function is a buffer formatted as shown in Table 4-6.

Table 4-6 Get Namespace Label Size – Output Format

Field	Byte Length	Byte Offset	Description

			Uh mU U U U

4.5 Get Namespace Label Data (Function Index 5)

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

4.5.1 Input (Arg3)

The input is a package containing a single buffer, where the buffer is formatted as shown in Table 4-7.

Table 4-7 Get Namespace Label Data – Input Format

Field	Byte Length	Byte Offset	Description

4.5.2 Output

The return value for this function is a buffer formatted as shown in Table 4-8.

Table 4-8 Get Namespace Label Data – Output Format

Field	Byte Length	Byte Offset	Description

			U m

4.6 Set Namespace Label Data (Function Index 6)

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

4.6.1 Input (Arg3)

Input is a package containing a single buffer, where the buffer is formatted as shown in Table 4-9.

Table 4-9 Set Namespace Label Data – Input Format

Field	Byte Length	Byte Offset	Description
			Uh mU U U U

4.6.2 Output

The return value for this function is a buffer formatted as shown in Table 4-10.

Table 4-10 Set Namespace Label Data – Output Format

Field	Byte Length	Byte Offset	Description

4.7 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.

4.7.1 Input (Arg3)

None

4.7.2 Output

The return value for this function is a buffer formatted as shown in Table 4-11.

Table 4-11 Get Vendor Specific Command Effect Log Size – Output Format

Field	Byte Length	Byte Offset	Description

4.8 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.

4.8.1 Input (Arg3)

None

4.8.2 Output

The return value for this function is a buffer formatted as shown in Table 4-12.

Table 4-12 Get Vendor Specific Command Effect Log Size – Output Format

Field	Byte Length	Byte Offset	Description

Table 4-13 Command Effect Data - Format

Field	Byte Length	Byte Offset	Description

4.9 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.

4.9.1 Input (Arg3)

Input is a package containing a single buffer, where the buffer is formatted as shown in Table 4-14.

Table 4-14 Vendor Specific Command – Input Format

Field	Byte Length	Byte Offset	Description

4.9.2 Output

The return value for this function is a buffer formatted as shown in Table 4-15.

Table 4-15 Vendor Specific Command – Output Format

Field	Byte Length	Byte Offset	Description
			U m mm
			U U U U