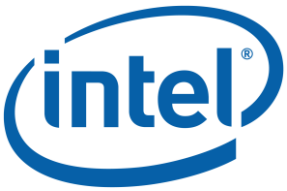


NVDIMM DSM Interface

Revision V1.6

August 9, 2017

The following modifications have been made to this version of the DSM specification:

- *General*
  - *Added two tables of supported Function Ids, Revision Ids and table of returned status values that apply to all DSMs*
  - *Make is clear that all Reserved fields must be written with 0's and read as 0's*
  - *Added FW Update – Theory of Operation section to outline the complete FW Update sequence*
  - *Updated Table 3-B to add deprecated Label API for Revision Id 2*
  - *Updated Table 3-C to clarify the differences between returned status 5 and 8*
- *Get SMART Health Info*
  - *Added missing range for Percentage Used*
  - *Fixed Byte 11 Alarm Trips to fix incorrect limits*
  - *Updated Unsafe Shutdown Count to 4 bytes*
  - *Added AIT DRAM Status byte and validity bit*
  - *Added Current NVDIMM PMIC Temperature bytes and validity bit*
- *Get SMART Threshold*
  - *Threshold Alarm Control is now 2 bytes instead of 1 with no reserved byte after*
- *Get Command Effect Log*
  - *Added list of operations that must be quiesced when handling Quiesce All IO (QIO) effect*
  - *Updated Byte Length field in table for Command Effect Data*
- *Pass-Through Command*
  - *Updated Byte Length field in table for OpCode Parameters Data*
  - *Updated Byte Length field in table for Output Data*
- *Enable Latch System Shutdown Status*
  - *Simplified description, made it clear that the NVDIMM powers up in disabled state and removed disable feature*
- *Renamed Functions (no logic change)*
  - *Get Vendor Specific Command Effect Log Size to Get Command Effect Log Info*
  - *Get Vendor Specific Command Effect Log to Get Command Effect Log*
  - *Vendor Specific Command to Pass-Through Command*
  - *Set Latch System Shutdown Status to Enable Latch System Shutdown Status*

The following additions have been made to this version of the DSM specification:

- *Get Supported Modes*
- *Get FW Info*
- *Start FW Update*
- *Send FW Update Data*
- *Finish FW Update*
- *Query Finish FW Update Status*
- *Set SMART Threshold*
- *Inject Error*

Contents

Contents

1 Introduction 4
1.1 Document Scope 4
1.2 Related Documents 4
1.3 Terminology 4
2 _DSM Interface for NVDIMM ACPI0012 Root Device 5
3 _DSM Interface for the NVDIMM Device 6
3.1 Get SMART and Health Info (Function Index 1) 9
3.2 Get SMART Threshold (Function Index 2) 12
3.3 Get Block NVDIMM Flags (Function Index 3) 14
3.4 Deprecated - Get Namespace Label Size (Function Index 4) 15
3.5 Deprecated - Get Namespace Label Data (Function Index 5) 16
3.6 Deprecated - Set Namespace Label Data (Function Index 6) 17
3.7 Get Command Effect Log Info (Function Index 7) 18
3.8 Get Command Effect Log (Function Index 8) 19
3.9 Pass-Through Command (Function Index 9) 21
3.10 Enable Latch System Shutdown Status (Function Index 10) 22
3.11 Get Supported Modes (Function Index 11) 23
3.12 Get FW Info (Function Index 12) 24
3.13 Start FW Update (Function Index 13) 26
3.14 Send FW Update Data (Function Index 14) 28
3.15 Finish FW Update (Function Index 15) 30
3.16 Query Finish FW Update Status (Function Index 16) 32
3.17 Set SMART Threshold (Function Index 17) 34
3.18 Inject Error (Function Index 18) 36
4 FW Update - Theory of Operation 39

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 V6.2 specification. This document specifically discusses the NVDIMM Device Specific Method (_DSM).

1.2 Related Documents

The related documents are ACPI Specification Version 6.0, 6.1 & 6.2, UEFI 2.7 NVDIMM Label Protocol, UEFI 2.7 NVDIMM BTT Layout (http://www.uefi.org/specifications) and this DSM Specification (http://pmem.io/documents).

1.3 Terminology

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

Table 1-1 – Terminology

Term	Description
NFIT	The NVDIMM Firmware Interface Table defines the ACPI 6.2 specified 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. See the ACPI 6.2 NVDIMM Label additions and the UEFI 2.7 NVDIMM Label Protocol additions to describe this in more detail.
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. See the ACPI 6.2 NVDIMM Label additions, and the UEFI 2.7 NVDIMM Label Protocol additions to describe this in more detail.
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 _DSM Interface for NVDIMM ACPI0012 Root Device

All Root ACPI0012 scoped _DSMs are now found in the following specifications and have been removed from this document, which will now only document the NVDIMM _DSMs.

Please see:
ACPI Specification V6.0 – Initial NVDIMM NFIT additions
ACPI Specification V6.1 – Addition of Common ARS _DSMs, Clear Uncorrectable Error _DSM
ACPI Specification V6.2 – Addition of NVDIMM Label API, ARS Error Injection _DSMs
UEFI Specification V2.7 – See additions of NVDIMM Label Protocol and BTT Layout

3 _DSM Interface for the NVDIMM Device

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 the NVDIMM Root Device associated with _HID of ACPI0012 in ACPI name space hierarchy.

The following tables outlines the required Arg1, Arg2 parameters that are to be utilized for this version of the specification. The platform shall support the Arg1 - Revision Id = 1 and Arg1 - Revision Id = 2 Function Indexes simultaneously as outlined below. No other Arg1 - Revision Id values are supported at this time.

Arg0 – UUID - *4309AC30-0D11-11E4-9191-0800200C9A66*

Table 3-A Supported Function Index for Arg1 - Revision Id = 1

Arg1 - Revision Id	Arg2 – Function Index	_DSM Function Name
1	0	*Query implemented commands per ACPI Specification (returns the list below based on Arg1 - Revision Id = 1).*
	1	*Get SMART and Health Info*
	2	*Get SMART Threshold*
	3	*Get Block NVDIMM Flags*
	4	*Deprecated - Get Namespace Label Data Size*
	5	*Deprecated - Get Namespace Label Data*
	6	*Deprecated - Set Namespace Label Data*
	7	*Get Command Effect Log Info*
	8	*Get Command Effect Log*
	9	*Pass-Through Command*
	10	*Enable Latch System Shutdown Status*

Table 3-B Supported Function Index for Arg1 - Revision Id = 2

Arg1 - Revision Id	Arg2 – Function Index	_DSM Function Name
2	0	*Query implemented commands per ACPI Specification (returns the list below based on Arg1 - Revision Id = 2).*
	1	*Get SMART and Health Info*
	2	*Get SMART Threshold*
	3	*Deprecated - Get Block NVDIMM Flags*
	4	*Deprecated - Get Namespace Label Data Size*
	5	*Deprecated - Get Namespace Label Data*
	6	*Deprecated - Set Namespace Label Data*
	7	*Get Command Effect Log Info*
	8	*Get Command Effect Log*
	9	*Pass-Through Command*
	10	*Enable Latch System Shutdown Status*
	11	*Get Supported Modes*
	12	*Get FW Info*
	13	*Start FW Update*
	14	*Send FW Update Data*
	15	*Finish FW Update*
	16	*Query Finish FW Update Status*
	17	*Set SMART Threshold*
	18	*Inject Error*

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. For DSM functions that do not take an input parameter, Arg3 is an empty package. The output of all functions in the DSM is a Buffer with a list of bytes. The first four bytes provide Status and Extended Status for the DSM function. Depending on the status code, additional bytes may follow the status bytes. If status bytes signal an error condition, the additional bytes are not present, unless some additional information is explicitly defined for the particular error code. If status bytes signal success, all output bytes defined for the function are present.

The following table outlines the returned Status field common to all of the DSMs defined in this specification. The status 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.

Table 3-C Supported _DSM Return Status Values

Return Status Bytes[1-0] Value	Return Status Meaning
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 - Command Timed Out, Other Command In Progress, Mailbox not Ready*
6	*Failure – Unknown Reason*
7	*Function Specific Error (details in Extended Status Field)*
8	*Failure – Retry Suggested - Out of Resources*
9	*Failure – HW Not Ready*

3.1 Get SMART and Health Info (Function Index 1)

This command requests the device to return Smart and Health information for the requested device.

Function Input

None

Function Output

The following tables outline the expected output payload for this command.

Table 3-1 Get SMART and Health Info – Output Format

Field	Byte Length	Byte Offset	Description
Status	2	0	Defined above in Table 3-C
Extended Status	2	2	Extended Status Field
Smart and Health Data	128	4	Output formatted as shown in Table 3-2.

Table 3-2 SMART and Health Data – Output Format

Field	Byte Length	Byte Offset	Description
Validity Flags	4	0	Validity 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 Remaining 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 Bit[5] – If set to 1, indicates that Unsafe Shutdown Count field is valid Bit[6] – If set to 1, indicates that the AIT DRAM Status field is valid Bit[7] – if set to 1, indicates that Current NVDIMM PMIC Temperature field is valid Bit[8] – Reserved, shall return 0. 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, shall return 0.
Reserved	4	4	Shall return 0.
Health Status	1	8	Health Status (HS): Overall health summary. Normal health is indicated by all HS bits being clear. Only one bit will be set at a time. Bit[0] – if set to 1, indicates Non-Critical condition, maintenance required but no data loss detected Bit[1] – if set to 1, indicates Critical condition, features or performance degraded due to failures but no data loss detected Bit[2] – if set to 1, indicates fatal condition, data loss is detected or is imminent. Bit[7:3] - Reserved, shall return 0.
Spare Blocks Remaining	1	9	Spare Blocks Remaining: 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
Percentage Used	1	10	Percentage Used: Device life span as percentage Valid range 0 to 100. 100 = the warranted life span of the device has been reached.
Alarm Trips	1	11	Alarm Trips: Bits to signify if values have tripped their respective alarm thresholds Bit[0] - Spare Blocks Remaining Trip - If set then the Spare Blocks Remaining value has gone below the pre-programmed threshold limit Bit[1] – NVDIMM Media Temperature Trip - If set then the NVDIMM Media temperature value has gone above the pre-programmed threshold limit Bit[2] – NVDIMM Controller Temperature Trip - If set then the NVDIMM Controller temperature value has gone above the pre-programmed threshold limit Bits[7:3] - Reserved, shall return 0.
Current NVDIMM	2	12	Current Media Temperature: Current temperature of the NVDIMM Media

Media Temperature			Bits[14:0] - Temperature in 0.0625 degree Celsius resolution. Bit[15] – Sign bit for temperature (1 = negative, 0 = positive)
Current NVDIMM Controller Temperature	2	14	Current Controller Temperature: Current temperature of the NVDIMM Controller Bits[14:0] - Temperature in 0.0625 degree Celsius resolution. Bit[15] – Sign bit for temperature (1 = negative, 0 = positive)
Unsafe Shutdown Count	4	16	Unsafe Shutdown Count (USC) – Number of times the NVDIMM Last Shutdown Status (LSS) was non-zero. Incremented anytime Last Shutdown Status (LSS) != 0 & Latch System Shutdown Status is set by host SW (via Enable Latch System Shutdown Status _DSM) . Count wraps back to 0 at overflow.
AIT DRAM Status	1	20	AIT DRAM Status 00h – AIT DRAM is disabled 01h – AIT DRAM is enabled If the AIT DRAM is disabled, it will cause a performance degradation and will trigger a SMART Health Status change to critical state
Current NVDIMM PMIC Temperature	2	21	Current PMIC Temperature: Current temperature of the NVDIMM PMIC Bits[14:0] - Temperature in 0.0625 degree Celsius resolution. Bit[15] – Sign bit for temperature (1 = negative, 0 = positive)
Reserved	8	23	Shall return 0.
Last Shutdown Status	1	31	Last Shutdown Status (LSS): status of last shutdown 00h – Clean shutdown All other Values – Not Clean Shutdown, indicates that there was either a platform or memory device-related failure occurred when saving data targeted for this memory device. Unsafe Shutdown Count (USC) above maintains a count of the number of times a non-clean shutdown occurs. Updated when Latch System Shutdown Status is set by host SW (via Enable Latch System Shutdown Status _DSM)
Size of Vendor Specific Data	4	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.
Vendor Specific SMART Data	92	127-36	Vendor-specific SMART Data

3.2 Get SMART Threshold (Function Index 2)

This command requests the device to return Smart Threshold values that have been programmed by the platform for the requested device.

Function Input

None

Function Output

The following tables outline the expected output payload for this command.

Table 3-3 Get SMART Threshold – Output Format

Field	Byte Length	Byte Offset	Description
Status	2	0	Defined above in Table 3-C
Extended Status	2	2	Extended Status Field
Smart Threshold Data	8	4	Output formatted as shown in Table 3-4.

Table 3-4 SMART Threshold Data – Output Format

Field	Byte Length	Byte Offset	Description
Threshold Alarm Enable	2	0	Threshold Alarm Control – If a 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] - Spare Blocks Remaining Threshold Alarm Enable Bit[1] – NVDIMM Media Temperature Threshold Alarm Enable Bit[2] – NVDIMM Controller Temperature Threshold Alarm Enable Bit[15:3] - Reserved, shall return 0
Spare Blocks Remaining Threshold	1	2	Spare Blocks Remaining Threshold: Remaining Spare Capacity as % of factory configured space. Valid range 0 to 100. If the *Spare Blocks Remaining Threshold Alarm Enable* bit is set and when the remaining spare block capacity goes below this threshold, the *Spare Blocks Remaining Trip* bit will be set in the SMART and Health Data structure defined in Table 3-2.
NVDIMM Media Temperature Threshold	2	3	Media Temperature Threshold Bit[14:0] – Temperature in 0.0625 degree Celsius resolution. Bit[15] – Sign bit for temperature (1 = negative, 0 = positive) If the *NVDIMM Media Temperature Threshold Alarm Valid* bit is enabled and when the *NVDIMM Media* temperature goes above this value, the *NVDIMM Media Temperature Trip* bit will be set in the SMART and Health Data structure defined in Table 3-2.
NVDIMM Controller Temperature Threshold	2	5	Controller Temperature Threshold Bit[14:0] - Temperature in 0.0625 degree Celsius resolution. Bit[15] - Sign bit for temperature (1 = negative, 0 = positive) If the *NVDIMM Controller Temperature Threshold Alarm Valid* bit is enabled and when the NVDIMM Controller temperature goes above this value, the *NVDIMM Controller Temperature Trip* bit will be set in the SMART and Health Data structure defined in Table 3-2.
Reserved	1	7	Shall return 0.

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). Used by the NVDIMM to report specific features or alternative sequences that need to be implemented by SW drivers.

Warning: This function has been deprecated. It is included here for backwards compatibility with existing Arg1 - Revision Id = 1 implementations.

Function Input

None

Function Output

The following tables outline the expected output payload for this command.

Table 3-5 Get Block NVDIMM Flags - Output Format

Field	Byte Length	Byte Offset	Description
Status	2	0	Defined above in Table 3-C
Extended Status	2	2	Extended Status Field
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 cache lines 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, shall return 0 Byte[3:1] – Reserved, shall return 0

3.4 Deprecated - Get Namespace Label Size (Function Index 4)

This command requests the device to return the size of the Namespace Label storage area for the requested device.

Warning: This function has been deprecated in preference to the ACPI 6.2 _LSI (Label Storage Information) NVDIMM Device Interface and is only supported with Arg1 – Revision Id = 1. It is included here for backwards compatibility with existing Arg1 - Revision Id = 1 implementations.

Function Input

None

Function Output

The following tables outline the expected output payload for this command. See **updated/new additions & clarifications** below for this existing LSM.

Table 3-6 Get Namespace Label Size – Output Format

Field	Byte Length	Byte Offset	Description
Status	2	0	Defined above in Table 3-C
Extended Status	2	2	01h – Extended Success Status - Locked Persistent Memory Region – The PMEM Region is currently in a locked state. This DSM is expected to continue to report a valid namespace label size, returns status success (0) and reports this extended status if the persistent memory region of the NVDIMMs are in a state that requires one or more security keys to be applied before the region is accessible.
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 Deprecated - Get Namespace Label Data (Function Index 5)

This command requests the device to return Namespace Label storage area data based on the requested buffer offset and length for the requested device.

Warning: This function has been deprecated in preference to the ACPI 6.2 _LSR (Label Storage Read) NVDIMM Device Interface and is only supported with Arg1 – Revision Id = 1. It is included here for backwards compatibility with existing Arg1 - Revision Id = 1 implementations.

Function Input

The following tables outline the expected input payload for this command.

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

Function Output

The following tables outline the expected output payload for this command.

Table 3-8 Get Namespace Label Data – Output Format

Field	Byte Length	Byte Offset	Description
Status	2	0	Defined above in Table 3-C 03h – Invalid Input Parameters - Offset + Length is > size of Namespace Label Data Area (Max Namespace Label Data Length from GetNamespaceLabelDataSize LSM) - Length is > maximum amount of data the OSPM can transfer in a single request
Extended Status	2	2	Extended Status Field
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 Deprecated - Set Namespace Label Data (Function Index 6)

This command requests the device to update Namespace Label Data area data based on the requested buffer offset and length for the requested device.

Warning: This function has been deprecated in preference to the ACPI 6.2 _LSW (Label Storage Write) NVDIMM Device Interface and is only supported with Arg1 – Revision Id = 1. It is included here for backwards compatibility with existing Arg1 - Revision Id = 1 implementations.

Function Input

The following tables outline the expected input payload for this command.

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.

Function Output

The following tables outline the expected output payload for this command.

Table 3-10 Set Namespace Label Data – Output Format

Field	Byte Length	Byte Offset	Description
Status	2	0	Defined above in Table 3-C 03h – Invalid Input Parameters - Offset + Length is > size of Namespace Label Data Area (Max Namespace Label Data Length from GetNamespaceLabelDataSize LSM) - Length is > maximum amount of data the OSPM can transfer in a single request
Extended Status	2	2	Extended Status Field

3.7 Get Command Effect Log Info (Function Index 7)

This command requests the device to return the Command Effect Log Information for the requested device.

Function Input

None

Function Output

The following tables outline the expected output payload for this command.

Table 3-11 Get Command Effect Log Info – Output Format

Field	Byte Length	Byte Offset	Description
Status	2	0	Defined above in Table 3-C
Extended Status	2	2	Extended Status Field
Max Command Effect Log Data Length	4	8	In bytes, Maximum size of the command effect log data buffer supported by the device

3.8 Get Command Effect Log (Function Index 8)

This command requests the device to return the Command Effect Log associated with the requested device. If the OpCode is not in the Command Effect log, OSPM may block the Pass-Through Command calls for that OpCode.

Function Input

None

Function Output

The following tables outline the expected output payload for this command.

Table 3-12 Get Command Effect Log – Output Format

Field	Byte Length	Byte Offset	Description
Status	2	0	Defined above in Table 3-C
Extended Status	2	2	Extended Status Field
OpCode Count	2	4	Number of OpCode command effect logs returned
Reserved	2	6	Shall return 0.
Command Effect Data	Max Command Effect Log Data Length from Get Command Effect Log Info	8	The command effect data for each OpCode. The Fields in Table 3-8 are repeated *OpCode Count* times.

Table 3-13 Command Effect Data – Output 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 must 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. Operations that must be quiesced include cpu load/store/move/flush memory operations, writes to NFIT Flush Hint Addresses, HW Block aperture programming sequences, in progress long operation sequence including ARSs, and NVDIMM controller mailbox commands. 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. Bit[31:8] – Reserved, shall return 0.

3.9 Pass-Through Command (Function Index 9)

This command requests the device to execute the vendor specific command contained in the input payload for the requested device.

Function Input

The following tables outline the expected input payload for this command.

Table 3-14 Pass-Through 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	OpCode Parameters Data Length	8	Vendor-specific command input data

Function Output

The following tables outline the expected output payload for this command.

Table 3-15 Pass-Through Command – Output Format

Field	Byte Length	Byte Offset	Description
Status	2	0	Defined above in Table 3-C
Extended Status	2	2	Extended Status Field
Output Data Length	4	4	In bytes. If *Status* is not *Success,* output data length returned is 0.
Output Data	Output Data Length	8	The *Output Data* is valid only when the *Output Data Length* is non-zero.

3.10 Enable Latch System Shutdown Status (Function Index 10)

DSM command to allow a SW agent enable the latching of SMART LSS & SMART Unsafe Shutdown Count state of each NVDIMM. By default the NVDIMM powers up assuming that this latch is disabled. When the latch is disabled the NVDIMM will report the previously saved value for the SMART LSS and SMART USC values. Those values will not change again until the next power down sequence following the enable of the latch utilizing this DSM.

Function Input

The following tables outline the expected input payload for this command.

Table 3-16 Enable Latch System Shutdown Status – Input Format

Field	Byte Length	Byte Offset	Description
Latch System Shutdown Status	1	0	Enable System Shutdown Status –Enables latching of SMART Last Shutdown Status (LSS) & SMART Unsafe Shutdown Count in NVDIMM on the next power down event. 01h – Enable the latch. Update SMART LSS & SMART Unsafe Shutdown Count on next power-down, power-up sequence All other values are reserved.

Function Output

The following tables outline the expected output payload for this command.

Table 3-17 Enable Latch System Shutdown Status – Output Format

Field	Byte Length	Byte Offset	Description
Status	2	0	Defined above in Table 3-C
Extended Status	2	2	Extended Status Field

3.11 Get Supported Modes (Function Index 11)

This command requests the platform to return details about the supported Modes of the NVDIMM Interface implementation.

Function Input

None

Function Output

The following tables outline the expected output payload for this command.

Table 3-18 Get Supported Modes – Output Format

Field	Byte Length	Byte Offset	Description
Status	2	0	Defined above in Table 3-C
Extended Status	2	2	Extended Status Field
Supported Modes	2	4	The list of the DIMMs capabilities: Bit[0] – Memory Mode Bit[1] – PMEM Mode supported Bit[2] – Block Aperture Mode supported Bit[15:3] – Reserved, shall return 0.

3.12 Get FW Info (Function Index 12)

This command returns information for the limits utilized for Send FW Update Data function, the running FW image revision, the running FW image Firmware Interface Specification (FIS) version, and the Updated FW Image, if one exists. See FW Update – Theory of Operation section for more information on the complete sequence including implementation of this command in the BIOS.

Function Input

None

Function Output

The following tables outline the expected output payload for this command.

Table 3-19 Get FW Info – Output Format

Field	Byte Length	Byte Offset	Description
Status	2	0	Defined above in Table 3-C
Extended Status	2	2	Extended Status Field
Size of FW Update Image Storage Area	4	4	In bytes, Total size of the FW Update Image Storage Area supported by the platform.
Max Send FW Update Data Length	4	8	In bytes, Maximum Length value that can be utilized with each Send FW Update Data command.
Query Finish FW Update Status Polling Interval	4	12	Polling interval in uSecs describing how often software should issue a Query Finish FW Update Status polling command to check for Finish FW Update completion.
Max Time to Query Finish FW Update Status	4	16	Maximum time in uSec software should have to poll for Query Finish FW Update Status on a single NVDIMM.
FW Update Capabilities	1	20	Flags further defining the FW Update capabilities or features of the NVDIMM 00h – FW Update Requires Cold Boot – If set the NVDIMM requires a system cold-boot for the new FW image to become the new executing FW image. This assumes that the FW update sequence has completed successfully. All other values are reserved and read as 0.
Reserved	3	21	Read as 0

Running FW Interface Version	4	24	The current running FW Interface Specification (FIS) revision using the product specific format. -Implementations that do not report a full 4 bytes of Running FW Interface Version information shall fill unused MSB bytes with 0's. Note: This is for informational purposes only and shall not be utilized to determine the command set that is supported by the NVDIMM.
Running FW Revision	8	28	Contains the revision information of the currently running NVDIMM firmware using the product specific format. -Implementations that do not report a full 8 bytes of Running FW Revision information shall fill unused MSB bytes with 0's. -Larger version value indicate newer FW revision.
Updated FW Revision	8	36	Upon successful completion of the Finish FW Update command this field contains the revision information of the updated NVDIMM firmware using the product specific format. -This revision becomes valid after successful completion of a Send FW Update Data & Finish FW Update sequence. This field then becomes invalid after a cold system boot and this revision shall be reported as all 0's at that time. -If no FW image has been sent or an image has been sent but the update has not been finished, or the Finish FW Update fails, then this revision shall be reported as all 0's. -Implementations that do not report a full 8 bytes of Updated FW Revision information shall fill unused MSB bytes with 0's. -Larger version value indicate newer FW revision.

3.13 Start FW Update (Function Index 13)

This command requests the NVDIMM device to start a FW download sequence. The FW download sequence consists of a single Start FW Update, followed by one or more Send FW Update Data commands and completes with a single Finish FW Update command followed by one or more Query Finish FW Update Status to poll for Finish FW Update completion. See FW Update – Theory of Operation section for more information on the complete sequence.

Function Input

None

Function Output

The following tables outline the expected output payload for this command.

Table 3-20 Start FW Update – Output Format

Field	Byte Length	Byte Offset	Description
Status	2	0	Defined above in Table 3-C 00h – Success. FW Update Context field is valid. 05h - Failure – Retry Suggested - ARS in progress. Software shall wait for ARS completion utilizing Get ARS Status, before starting a FW update sequence. 07h – Function Specific Status (see Extended Status below) 08h - Failure – Retry Suggested - Out of Resources. Software may need to complete other outstanding FW update sequences, potentially for other NVDIMM devices before retrying the Start FW Update command. It is also possible to abort other FW update sequences in progress to recover internal platform resources, using the Control Flags in the Finish FW Update input payload.
Extended Status	2	2	Extended Status Field 01h – FW Update already in progress for this NVDIMM device. The FW Update Context field returned is valid and indicates the context for the currently executing FW Update on the NVDIMM device. Software must complete the current FW update sequence with one of the two methods: -Sending a Finish FW Update command and possibly a system cold-boot before another FW update sequence can be started on the same NVDIMM -Using the returned FW Update Context to abort the existing FW Update that is in progress by calling Finish FW Update with the Control Flag set to Abort Existing FW Update Sequence

			02h – FW Update already occurred – A successful FW update sequence has already occurred and another Start FW Update command is being attempted without a system cold-boot.
FW Update Context	4	4	Upon successful completion of the Start FW Update command this field contains a platform implementation specific value that must be passed as an input parameter to Send FW Update Data and Finish FW Update commands.

3.14 Send FW Update Data (Function Index 14)

This command requests the device to update the FW image in the NVDIMMs FW Update Image Storage Area as part of a FW download sequence. The FW download sequence consists of a single Start FW Update, followed by one or more Send FW Update Data commands and completes with a single Finish FW Update command followed by one or more Query Finish FW Update Status to poll for Finish FW Update completion. See FW Update – Theory of Operation section for more information on the complete sequence.

The Offset and Length fields allow software to divide the FW image in to pieces based on the Max Send FW Update Data Length reported in the Get FW Info output payload. There is no ordering restriction regarding how the pieces of the FW image are sent to the NVDIMMs FW Update Image Storage Area.

No validation of the FW image occurs until the FW download sequence is complete. The FW image is considered complete and its validity is verified only after the Finish FW Update command has completed.

If software is aborting a FW Update sequence that is already in progress it can call Finish FW Update directly without issuing any Send FW Update Data commands. See the Control Flags in the Finish FW Update command for details on aborting an outstanding FW Update sequence.

Function Input

The following tables outline the expected input payload for this command.

Table 3-21 Send FW Update Data – Input Format

Field	Byte Length	Byte Offset	Description
FW Update Context	4	0	Platform specific FW update sequence context provided by the platform as part of the Start FW Update output payload.
Offset	4	4	In bytes Indicates the byte offset in the NVDIMMs FW Update Image Storage Area where this portion of the FW Image data will be written
Length	4	8	In bytes Indicates the number of bytes to be written starting at the Offset specified above
FW Image Data	Length	12	FW Image data to be written at the starting Offset for Length bytes

Function Output

The following tables outline the expected output payload for this command.

Table 3-22 Send FW Update Data – Output Format

Field	Byte Length	Byte Offset	Description
Status	2	0	Defined above in Table 3-C 03h – Invalid Input Parameters - Offset + Length is > Size of FW Update Image Storage Area reported in the Get FW Info command - Length is > Max Send FW Update Data Length reported in the Get FW Info command 05h - Failure - ARS in progress. Software shall wait for ARS completion utilizing Get ARS Status, before starting retrying the FW update sequence. 07h – Function Specific Status (see Extended Status below) 08h - Failure – Out of Resources. Software may need to complete other outstanding FW update sequences, potentially for other NVDIMM devices before retrying the Start FW Update command. It is also possible to abort other FW update sequences in progress to recover internal platform resources, using the Control Flags in the Finish FW Update input payload.
Extended Status	2	2	Extended Status Field 01h – FW Update Context invalid

3.15 Finish FW Update (Function Index 15)

This command requests the NVDIMM device to begin the process of finishing a FW download sequence. The FW download sequence consists of a single Start FW Update, followed by one or more Send FW Update Data commands and completes with a single Finish FW Update command followed by one or more Query Finish FW Update Status to poll for Finish FW Update completion. See FW Update – Theory of Operation section for more information on the complete sequence.

Upon successful completion of this command, the NVDIMM has begun the process of finishing the FW update process. This consists of decrypting the FW image header, verifying header information including checksum, and saving the FW image in the internal NVDIMM FW Image Storage Area. This can take seconds to complete, requiring the use of the Query Finish FW Update Status so that applications can poll for Update FW completion without waiting for the update to be completed by the NVDIMM.

Software must issue the Query Update FW Status command to poll for Update FW completion. The Update FW image sequence is not complete until the query command returns proper status indicating the Update FW process is complete.

The Control Flags allow software to abort an existing FW Download instead of completing the sequence. Aborting a FW download sequence results in no change to the NVDIMM FW image. If aborting a FW Update sequence, software does not send the Query Finish FW Update command.

Function Input
The following tables outline the expected input payload for this command.

Table 3-23 Finish FW Update – Input Format

Field	Byte Length	Byte Offset	Description
Control Flags	1	0	Finish FW Update Control Flags 00h – Finish the FW Update sequence. Once software instructs the platform to finish the FW Update, it is not possible to abort the Finish FW Update sequence at a later date. Software needs to wait for the FW Update to complete using the Query Finish FW Update Status. 01h – Abort Existing FW Update Sequence. The FW Update Context describes an existing FW Download sequence that should be aborted without updating the FW image on the NVDIMM. When aborting a FW update sequence software does not call Query Finish FW Update Status. All other values are reserved.
Reserved	3	1	Must be 0

FW Update Context	4	4	Platform specific FW update sequence context provided by the platform as part of the Start FW Update output payload.

Function Output

The following tables outline the expected output payload for this command.

Table 3-24 Finish FW Update – Output Format

Field	Byte Length	Byte Offset	Description
Status	2	0	Defined above in Table 3-C 00h – Success – The Finish FW Update sequence has started. Software shall call Query Finish FW Update Status command to poll for FW Update sequence completion 05h - Failure – ARS in progress. Software shall wait for ARS completion utilizing Get ARS Status, before retrying a FW update sequence. 07h – Function Specific Status (see Extended Status below) 08h - Failure – Out of Resources. Software may need to complete other outstanding FW update sequences, potentially for other NVDIMM devices before retrying the Start FW Update command. It is also possible to abort other FW update sequences in progress to recover internal platform resources, using the Control Flags in the Finish FW Update input payload.
Extended Status	2	2	Extended Status Field - Any non-zero value returned here means the FW Update sequence is not active. Software does not need to call Query Finish FW Update Status for any of these cases. 01h – FW Update Context invalid 02h – FW Update already occurred – A successful FW update sequence has already occurred and another Finish FW Update command is being attempted without a system cold-boot. 03h – Current updated FW Image failed authentication checks – fallback to prior FW image 04h – FW update sequence successfully aborted. Only returned if the caller requested a FW Update sequence to be aborted by setting Control Flags to Abort Existing FW Update Sequence.

3.16 Query Finish FW Update Status (Function Index 16)

This command allows software to poll for completion of the FW download sequence. The FW download sequence consists of a single Start FW Update, followed by one or more Send FW Update Data commands and completes with a single Finish FW Update command followed by one or more Query Finish FW Update Status to poll for Finish FW Update completion. See FW Update – Theory of Operation section for more information on the complete sequence.

Finish FW Update consists of decrypting the FW image header, verifying header information including checksum, and saving the FW image in the internal FW Image Storage Area. This can take seconds to complete requiring the use of the Query Finish FW Update Status so that applications can poll for completion without the BIOS blocking in SMM waiting for the update to be completed by the NVDIMM. The Query Finish FW Update Status Polling Interval returned in the Get FW Info command specifies what frequency software should utilize when polling for Finish FW Update completion using the Query Finish FW Update Status command.

Upon successful completion of this command, the updated FW image will become the new executing FW image on the next cold-boot, replacing the currently executing FW image.

Sending a Finish FW Update followed by one or more Query Finish FW Update Status commands completes the FW download sequence and requests the NVDIMM to verify the Updated FW Image and report the revision information for the Updated FW Image. If no updated FW image is sent or the updated FW image is incomplete, Query Finish FW Update Status command will return an appropriate error and the Updated FW Image Revision will be reported as all 0's.

Only a single FW Update sequence can be handled per NVDIMM per system cold-boot sequence. Once successful status is returned for Query Finish FW Update Status, the system must be go through a cold-boot cycle before another FW Update sequence can be executed on that same NVDIMM. Multiple NVDIMMs can have FW images updated and utilize a single system cold-boot to activate the new FW image on all NVDIMMs.

Function Input
The following tables outline the expected input payload for this command.

Table 3-25 Query Finish FW Update Status – Input Format

Field	Byte Length	Byte Offset	Description
FW Update Context	4	0	Platform specific FW update sequence context provided by the platform as part of the Start FW Update output payload.

Function Output
The following tables outline the expected output payload for this command.

Table 3-26 Query Finish FW Update Status – Output Format

Field	Byte Length	Byte Offset	Description
Status	2	0	Defined above in Table 3-C 00h – Success – The Update FW sequence has completed successfully. Authentication checks passed. Updated FW Revision field is valid. The Updated FW Image will be loaded on the next system cold-boot. 07h – Function Specific Status (see Extended Status below) 08h - The Finish FW Update sequence timed out
Extended Status	2	2	Extended Status Field 01h – FW Update Context invalid 02h – FW Update in progress 03h – Current updated FW Image failed authentication checks – fallback to prior FW image 04h – Sequencing Error – Query Finish FW Update Status called without first calling Finish FW Update
Updated FW Revision	8	4	Upon successful completion of the Finish FW Update command this field contains the revision information of the updated NVDIMM firmware using the product specific format. -This becomes valid after successful completion of a Send FW Update Data & Finish FW Update sequence. This field then becomes invalid after a cold system boot. -If no FW image has been updated or the updated FW image is invalid, or the Finish FW Update fails, then this revision shall be reported as all 0's. -Implementations that do not report a full 8 bytes of Updated -FW Revision information shall fill unused MSB bytes with 0's. -Larger version value indicates newer FW revision.

3.17 Set SMART Threshold (Function Index 17)

This command requests the device to simultaneously enable specific SMART Threshold Alarm Triggers and set the SMART Threshold Alarm Trigger values for the device. Parameter values are verified first before any enable/disable state or threshold values are updated.

Function Input

The following tables outline the expected input payload for this command.

Table 3-27 Set SMART Threshold – Input Format

Field	Byte Length	Byte Offset	Description
Threshold Alarm Enable	2	0	Threshold Alarm Control - If a 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] - Spare Blocks Remaining Threshold Alarm Enable Bit[1] – NVDIMM Media Temperature Threshold Alarm Enable Bit[2] – NVDIMM Controller Temperature Threshold Alarm Enable Bit[15:3] - Reserved, shall be 0
Spare Blocks Remaining Threshold	1	2	Remaining Spare Capacity Alarm - A % of factory configured spare blocks. Values 0 & 100 are not valid and will result in an error. If the *Spare Blocks Remaining Threshold Alarm Enable* bit is set and when the spare block capacity goes below this threshold, the *Spare Blocks Remaining Trip* bit will be set in the SMART and Health Data structure defined in Table 3-2. This field is ignored if the *Spare Blocks Remaining Threshold Alarm Enable* bit above is cleared to 0.
NVDIMM Media Temperature Threshold	2	3	Media Temperature Alarm Bit[14:0] – Temperature in 0.0625 degree Celsius resolution. Bit[15] – Sign bit for temperature (1 = negative, 0 = positive) If the *NVDIMM Media Temperature Threshold Alarm Valid* bit is enabled and when the *NVDIMM Media* temperature goes above this value, the *NVDIMM Media Temperature Trip* bit will be set in the SMART and Health Data structure defined in Table 3-2. This field is ignored if the *NVDIMM Media Temperature Threshold Alarm Enable* bit above is cleared to 0.

NVDIMM Controller Temperature Threshold	2	5	Control Temperature Alarm Bit[14:0] - Temperature in 0.0625 degree Celsius resolution. Bit[15] - Sign bit for temperature (1 = negative, 0 = positive) If the *NVDIMM Controller Temperature Threshold Alarm Valid* bit is enabled and when the NVDIMM Controller temperature goes above this value, the *NVDIMM Controller Temperature Trip* bit will be set in the SMART and Health Data structure defined in Table 3-2. This field is ignored if the *NVDIMM Controller Temperature Threshold Alarm Enable* bit above is cleared to 0.

Function Output

The following tables outline the expected output payload for this command.

Table 3-28 Set SMART Threshold – Output Format

Field	Byte Length	Byte Offset	Description
Status	2	0	Defined above in Table 3-C 03h – Invalid Input Parameters Returned If any threshold value requested to be enabled is invalid. No changes are made to any previously set threshold enable/disable state and no changes are made to any previously set threshold values.
Extended Status	2	2	Extended Status Field

3.18 Inject Error (Function Index 18)

Inject NVDIMM specific errors not covered by the ACPI ARS Error Inject function. None of the injected errors are persistent across power cycles or reboots unless otherwise stated below. An error will stay injected until disabled using this command or the system is restarted, unless otherwise stated below.

Function Input

The following tables outline the expected input payload for this command.

Table 3-29 Inject Error - Input Format

Field	Byte Length	Byte Offset	Description
Error Inject Validity Flags	8	0	Valid Fields – 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 all Media Temperature Error Inject fields are valid Bit[1] – if set to 1, indicates that all Spare Blocks Remaining Trigger fields are valid Bit[2] – if set to 1, indicates that all Fatal Error Trigger fields are valid Bit[3] – if set to 1, indicates that all Unsafe Shutdown Error Trigger fields are valid Bit[63:4] – Reserved, shall be 0
Media Temperature Error Inject	3	8	Media Temperature Error Inject fields - This will override the NVDIMM from reading the actual temperature of the media device and spoof a media temperature reading of the injected value instead. Byte[0] Bit[0] – Enable If 0, injecting Media Temperature Errors is disabled. If 1, the Media Temperature specified will be injected. Bit[7:1] - Reserved, shall be 0. Byte[2:1] - Media Temperature to Inject Bit[14:0] – Temperature in Celsius with 0.0625 resolution Bit[15] – Sign Bit, if 1 the Temperature is negative, if 0 the temperature is positive Note: Although actions taken due to the Media Temperature injected may cause adverse effects on the NVDIMM, including IO throttling, the media temperature injected is an artificial temperature and will not cause harm to the NVDIMM. If the

			critical shutdown temperature, or higher, is injected, the NVDIMM may shutdown in order to preserve the part and data.
Spare Blocks Remaining Inject	2	11	Spare Blocks Remaining Trigger - This will spoof the NVDIMM to trigger either: -User Configured Spare Blocks Remaining Alarm for a previously set value using the Set SMART Threshold function -SMART Health Change Notification for Health Status Non-Critical or Critical Byte[0] Bit[0] – Enable If 0, injecting Spare Blocks Remaining is disabled If 1, the Spare Blocks Remaining will be injected Bit[7:1] – Reserved, shall be 0. Byte[1] – Spare Blocks Remaining to inject. Valid values are 0-99. All other values are reserved and will result in returned Status of Invalid Input Parameters. Note: For this trigger to inject a User Configured Spare Block Alarm Threshold Trigger requires the Spare Block Alarm Threshold to be previously enabled using the Set SMART Threshold function. If the Spare Block Alarm Threshold has not been enabled, this function will inject SMART Health Change notification ACPI Notification 0x81 as follows: Spare Blocks Remaining of 1% - Causes Health Status to change to Non-Critical Spare Blocks Remaining of 0% - Causes Health Status to change to Critical
Fatal Error Inject	1	13	Fatal Error Trigger – This trigger will spoof the NVDIMM to trigger a fatal NVDIMM error. Injecting this error will result in a change to the SMART Health Info – Health Status of fatal. Bit[0] – Enable If 0, injecting Fatal Error Trigger is disabled If 1, a Fatal Error Trigger will be injected Bit[7:1] – Reserved, shall be 0
Unsafe Shutdown Error Inject	1	14	Unsafe Shutdown Error Trigger – This trigger will spoof an ADR or system shutdown failure on the next power down as follows: -Enable SMART Last Shutdown Status (LSS) and Unsafe Shutdown Count (USC) increment via the Enable Latch System Shutdown Status DSM with Bit[0] - Enable System Shutdown Status set -Power down the system – The device spoofs a failure and latches SMART LSS, increments SMART USC

			-Power the system up – SMART Health Change is reported with non-zero LSS ad incremented USC Bit[0] – Enable If 0, injecting ADR Failure is disabled If 1, an ADR Failure will be injected Bit[7:1] – Reserved, shall be 0

Function Output

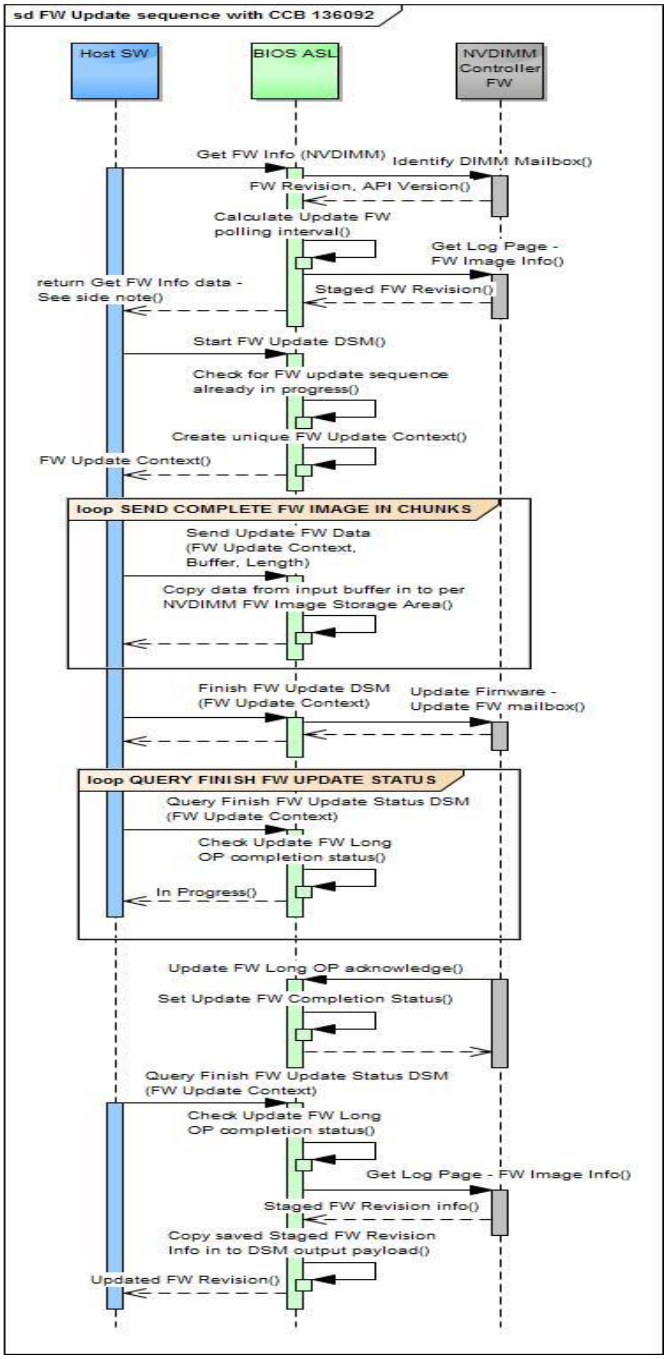
The following tables outline the expected output payload for this command.

Table 3-28 Inject Error Data – Output Format

Field	Byte Length	Byte Offset	Description
Status	2	0	Defined above in Table 3-C 03h – Invalid Input Parameters Returned If any Error Inject parameter value requested is invalid. No changes are made to any previous enable/disable Error Injection state and no changes are made to any previously set Error Inject values.
Extended Status	2	2	Extended Status Field 01h – Platform not enabled for error injection. Error Injection must be enabled on the platform before attempting to inject NVDIMM specific errors.

4FW Update - Theory of Operation

The following figure outlines the basic sequence for the native DSM Update FW support. It outlines the sequence of messages between a SW application, the Platform BIOS, and the NVDIMM Controller FW.



Basic Update FW Execution Flow