\_DSM for Virtual NVDIMMs

Jan 2019

Abstract

This document describes the \_DSM interface associated with virtual NVDIMM ACPI Namespace Devices. The interface specified in this document is exposed by NVDIMM ACPI objects running inside a virtual machine, with a Region Format Interface Code of 0x1901.

(c) 2019 Microsoft Corporation.  All rights reserved.  This document is provided "as-is." Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it. This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may use this document for your internal, reference purposes. You may modify this document for your internal, reference purposes.

Contents

[Change history 3](#_Toc481572386)

[Terms and Definitions 3](#_Toc481572387)

[Acronyms 3](#_Toc481572388)

[Terms 3](#_Toc481572389)

[Introduction 4](#_Toc481572390)

[\_DSM Interface for Virtual NVDIMMs 5](#_Toc481572391)

[Conventions 6](#_Toc481572392)

[Byte Ordering 6](#_Toc481572393)

[\_DSM Method Output 6](#_Toc481572394)

[Query Implemented Functions (Function Index 0) 7](#_Toc481572395)

[Get Health Information (Function Index 1) 8](#_Toc481572396)

[Get Unsafe Shutdown Count (Function Index 2) 9](#_Toc481572397)

[Inject Error (Function Index 3) 10](#_Toc481572398)

[Query Injected Errors (Function Index 4) 11](#_Toc481572399)

# Change history

|  |  |  |
| --- | --- | --- |
| **Version** | **Date** | **Description** |
| 1.0 | 5/3/2017 | * Initial version |
| 1.01 | 1/28/2019 | * Fix incorrect references to status field table. |

# Terms and Definitions

## Acronyms

|  |  |
| --- | --- |
| DIMM | Dual In-line Memory Module |
| NVDIMM | Non-volatile Dual In-line Memory Module |

## Terms

**Virtual NVDIMM**: an NVDIMM that is exposed to the OS running inside of a virtual machine. A virtual NVDIMM is a byte-addressable, persistent storage device.

# Introduction

This document describes the \_DSM interface associated with NVDIMM ACPI Namespace Devices[[1]](#footnote-2) that represent a virtual NVDIMM, that is, an NVDIMM exposed to a virtual machine. Virtual NVDIMMs have a Region Format Interface Code of 0x1901.

# \_DSM Interface for Virtual NVDIMMs

The \_DSM interface defined in this section applies to a virtual NVDIMM with a Region Format Interface Code (RFIC) of 0x1901 and is present under NVDIMM devices that are child devices of the NVDIMM Root Device object with \_HID of ACPI0012 in ACPI name space hierarchy.

Arg0 – UUID (set to 5746C5F2-A9A2-4264-AD0E-E4DDC9E09E80)

Arg1 – Revision ID (set to 1)

Arg2 – Function Index

0 – Query Implemented Functions

1 – Get Health Information

2 – Get Unsafe Shutdown Count

3 – Inject Error

4 – Query Injected Errors

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 functions that don’t take an input argument, Arg3 is an empty package.

The output of all functions in the DSM is a Buffer, list of bytes, value.

All functions in this specification are mandatory and shall be implemented, but some functions may return an error code that indicates the function is not enabled.

# Conventions

## Byte Ordering

Unless specifically stated otherwise, all multi-byte fields shall be represented in a little-endian manner.

## \_DSM Method Output

All methods shall return a buffer of length greater than or equal to 4. The first 4 bytes of the output buffer are structured in the following way:

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Byte Length** | **Byte Offset** | **Description** |
| General Status Code | 2 | 0 | 0 – Success  1 – Not Supported  2 – Invalid Input Parameters   * If a function that does not take an input argument is called with a non-empty package as Arg3, the function shall return General Status Code 2 (Invalid Input Parameters).   3 – Function-Specific Error Code  4 – Vendor-Specific Error Code  5 – 0xFFFF – Reserved |
| Function-Specific Error Code | 1 | 2 | This field contains an error code that is specific to the function that was called. This field only contains valid information if General Status Code is equal to Function-Specific Error Code. |
| Vendor-Specific Error Code | 1 | 3 | This field contains vendor-specific status codes. It only contains valid information if General Status Code is equal to Vendor-Specific Error Code. |

*Table 1 Status field*

Any non-zero General Status Code indicates that the function failed.

No function defined in this version of the specification shall return the “Not Supported” General Status Code.

# Query Implemented Functions (Function Index 0)

This function returns the functions supported by this interface version.

**Input (Arg3)**

None.

**Output**

This function returns an ACPI Buffer containing the byte values {0x1F}.

# Get Health Information (Function Index 1)

Get Health Information function returns health information about the virtual NVDIMM. It conveys information about three possible types of errors: data persistence loss, write persistence loss, and fatal error. These errors have the following meaning:

* Data persistence loss: the OS should assume that all data currently on the virtual NVDIMM will be lost when virtual machine loses power.
* Write persistence loss: the OS should assume that all new data written to the NVDIMM is not persistent after a write persistent lost notification.
* Fatal error: there has been a catastrophic failure on the NVDIMM. Reads may return invalid data and writes may not be persistent.

This \_DSM function returns information about runtime errors only. Data loss errors are reported in the Get Unsafe Shutdown Count function.

**Input (Arg3)**

None.

**Output**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Byte Length** | **Byte Offset** | **Description** |
| Status | 4 | 0 | See Table 1. |
| Health Status | 4 | 4 | A bitmask that represents the health condition of the virtual NVDIMM:  [0] – Data persistence loss  [1] – Write persistence loss  [2] – Fatal error  [3] – Data persistence loss imminent warning  [4] – Write persistence loss imminent warning  [5] – Fatal error imminent warning  [31:6] – Reserved  If all bits are 0, the virtual NVDIMM is healthy. |

*Table 2 Get Health Information – Output*

# Get Unsafe Shutdown Count (Function Index 2)

This function returns the unsafe shutdown count, which is the number of times that the virtual NVDIMM was shut down in a way that can result in data loss.

The unsafe shutdown count (USC) must be interpreted in a relative way. The OS should keep a record of the USC that was last returned by the NVDIMM. During system initialization, the OS should call this function and check if the returned USC is greater than the count that was stored. If it is, there might have been a data loss on the virtual NVDIMM, and the OS can choose to alert applications of that fact.

The OS shall safely handle cases where the USC returned by this function is less than the value stored. This can happen, for example, if the physical device on which the virtual NVDIMM is hosted changes and the virtualization platform is not able to detect whether the virtual NVDIMM might have lost data. The OS can choose to treat cases when the USC is less than the stored value as potential data loss cases.

**Input (Arg3)**

None.

**Output**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Byte Length** | **Byte Offset** | **Description** |
| Status | 4 | 0 | See Table 1. |
| Unsafe Shutdown Count | 4 | 4 | The number of times that this virtual NVDIMM has seen an unsafe shutdown.  If the Unsafe Shutdown Count reaches the value 0xFFFFFFFF and another unsafe shutdown event occurs, the Unsafe Shutodwn Count shall remain at 0xFFFFFFFF. |

# Inject Error (Function Index 3)

This function lets the operating system inject a failure on the virtual NVDIMM. The purpose of this function is to enable software validation.

If the platform supports virtual NVDIMM NFIT Health Event notifications, any call to this function that results in a change of the virtual NVDIMM’s health state shall trigger a notification.

The platform may choose to disable this function in certain circumstances. If error injection is disabled, this function shall return the Function-Specific Error Code 1.

**Input (Arg3)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Byte Length** | **Byte Offset** | **Description** |
| Errors | 4 | 0 | A bitmask that describes all the errors being injected:  [0] – Inject a data persistence loss error  [1] – Inject a write persistence loss error  [2] – Inject a fatal error  [3] – Inject a data persistence loss imminent warning  [4] – Inject a write persistence loss imminent warning  [5] – Inject a fatal error imminent warning  [6] – If 1, set the Unsafe Shutdown Count to the value in Injected Unsafe Shutdown Count  [31:7] – Reserved  Note: If a bit is 0, the platform shall clear the corresponding error. To clear all error injections, the OS calls this function with the Errors field set to 0. |
| Injected Unsafe Shutdown Count | 4 | 4 | The Unsafe Shutdown Count to be returned by Get Unsafe Shutdown Count. This field is valid when bit 6 of the Errors field is 1. |

*Table 3 Inject Error – Input*

**Output**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Byte Length** | **Byte Offset** | **Description** |
| Status | 4 | 0 | See Table 1. This function can return the following Function-Specific Error Code:  1: Error injection is disabled. |

*Table 4 Inject Error – Output*

# Query Injected Errors (Function Index 4)

This function returns information about errors currently being injected.

**Input (Arg3)**

None.

**Output**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Byte Length** | **Byte Offset** | **Description** |
| Status | 4 | 0 | See Table 1. |
| Is Error Injection Enabled | 1 | 4 | 0 – Error injection is not enabled by the platform.  1 – Error injection is enabled by the platform.  Note: if this field is 0, then Injected Errors shall be 0. |
| Injected Errors | 4 | 5 | A bitmask that represents the errors currently injected:  [0] – Data persistence loss error  [1] – Write persistence loss error  [2] – Fatal error  [3] – Data persistence loss imminent warning  [4] – Write persistence loss imminent warning  [5] – Fatal error imminent warning  [6] – An Unsafe Shutdown Count is injected  [31:2] – Reserved |
| Injected Unsafe Shutdown Count | 4 | 9 | The Unsafe Shutdown Count being returned by Get Unsafe Shutdown Count. This field is valid when bit 6 of the Injected Errors field is 1. |

*Table 5 Query Injected Errors – Output*

1. See ACPI 6.1 section 9.20.3 for the definition of NVDIMM ACPI Namespace Device. [↑](#footnote-ref-2)