Swordfish NVMe Model Overview and Mapping Guide

Version: 1.2.4a

Abstract: The Swordfish NVMe Model Overview and Mapping Guide defines the model to manage NVMe and NVMeoF storage systems with Redfish and Swordfish. It provides the detailed mapping information between the NVMe, NVMeoF specifications and the Redfish and Swordfish specifications.

SNIA Approved Publication

This document has been released and approved by the SNIA. The SNIA believes that the ideas, methodologies, and technologies described in this document accurately represent the SNIA goals and are appropriate for widespread distribution. Suggestion for revision should be directed to http://www.snia.org/feedback/.

Last Updated: 12 July 2022

Contents

List of Tables

List of Figures

USAGE

DISCLAIMER

The information contained in this publication is subject to change without notice. The SNIA makes no warranty of any kind with regard to this publication, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The SNIA shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use.

Suggestions for revisions should be directed to http://www.snia.org/feedback/.

Current Revision

SNIA is actively engaged in expanding and refining the Swordfish documentation. The most current revision can be found on the SNIA web site at https://www.snia.org/tech_activities/standards/curr_standards/swordfish.

Contact SNIA

Current SNIA practice is to make updates and other information available through their web site at http://www.snia.org.

FEEDBACK AND INTERPRETATIONS

Requests for interpretation, suggestions for improvement and addenda, or defect reports are welcome. They should be sent via the SNIA Feedback Portal at http://www.snia.org/feedback/ or by mail to the Storage Networking Industry Association, 4360 ArrowsWest Drive, Colorado Springs, Colorado 80907, U.S.A.

INTENDED AUDIENCE

This document is intended for use by individuals and companies engaged in storage management.

VERSIONING POLICY

This document is versioned material. Versioned material shall have a three-level revision identifier, comprised of a version number 'v', a release number 'r' and an errata number 'e'. Future publications of this document are subject to specific constraints on the scope of change that is permissible from one revision to the next and the degree of interoperability and backward compatibility that should be assumed between products designed to this standard. This versioning policy applies to all SNIA Swordfish versioned materials.

Version Number: Versioned material having version number 'v' shall be backwards compatible with all of revisions of that material that have the same version number 'v'. There is no assurance of interoperability or backward compatibility between revisions of a versioned material with different version numbers.

Release Number: Versioned material with a version number 'v' and release number 'r' shall be backwards compatible with previous revisions of the material with the same version number, and a lower release number. A minor revision represents a technical change to existing content or an adjustment to the scope of the versioned material. Each minor revision causes the release number to be increased by one.

Errata Number: Versioned material having version number 'v', a release number 'r', and an errata number 'e' should be backwards compatible with previous revisions of the material with the same version number and release number ("errata versions"). An errata revision of versioned material is limited to minor corrections or clarifications of existing versioned material. An errata revision may be backwards incompatible, if the incompatibility is necessary for correct operation of implementations of the versioned material.

Revision History

Revisions to this document are summarized in Table 1.

Table 1: Revision History

Date	Rev	Notes
18 August 2020	1.2.1	Initial Release
31 October 2020	1.2.1c	Released as SNIA Approved Publication
2 March 2021	1.2.2	Added detailed mapping information to match information in NVMe drive profiles for many new referenced properties, and included guidance for mandatory/recommended implementation as reflected in the profiles. Added sections for firmware update, with details for NVMe Drive implementation requirements. Added cross-references to User's Guide NVMe-specific use cases. Errata fixes – correct diagram, correct table headers.
30 August 2021	1.2.3	Update Mapping Guide with new mapping guidance, corresponding to the Swordfish NVMe Front End profile.
5 December 2021	1.2.3	Release as SNIA Approved Publication
12 April 2022	1.2.4	Release as SNIA Working Draft. Errata fixes throughout. Added section for EBOF configuration, and additions to appendix A to correspond to EBOF, JBOF, and NVMe-oF configurations.
12 July 2022	1.2.4a	Release as SNIA Standard.

About SNIA

The Storage Networking Industry Association (SNIA) is a non-profit organization made up of member companies spanning information technology. A globally recognized and trusted authority, SNIA's mission is to lead the storage industry in developing and promoting vendor-neutral architectures, standards and educational services that facilitate the efficient management, movement and security of information.

Acknowledgements

The SNIA Scalable Storage Management Technical Work Group, which developed and reviewed this work in progress, would like to recognize the significant contributions made by the members listed in Table 2.

Table 2: Contributors

Member	Representatives (* – prior employer)
Broadcom Inc.	Richelle Ahlvers*
Cisco Systems, Inc.	Krishnakumar Gowravaram
Dell Inc.	David Black
	Jim Pendergraft
	Michael Raineri
Hewlett Packard Enterprise	Curtis Ballard
	Jeff Hilland
	Chris Lionetti
Intel Corporation	Richelle Ahlvers
	Rajalaxmi Angadi
	Phil Cayton
	Slawek Putyrski
Kioxia	Mark Carlson
Lenovo	Keith Campbell
NetApp, Inc.	Don Deel
	Fred Knight
Samsung Corporation	Lu Fan
	Bill Martin
	Tom Rainey

Member	Representatives (* – prior employer)
VMware, Inc.	Murali Rajagopal

1 Abstract

The Swordfish NVMe Model Overview and Mapping Guide defines the model to manage NVMe and NVMe-oF storage systems with Redfish and Swordfish. It provides the detailed mapping information between the NVMe, NVMe-oF specifications and the Redfish and Swordfish specifications.

2 Scope

2.1 Document Goals

This document describes how both the NVMe Subsystem model and the NVMe-oF fabric system model should be mapped consistently to Redfish and Swordfish constructs for implementations to be managed within Redfish and Swordfish management environments.

This model and mapping information does not describe or assert any specific implementation recommendation technologies.

This document also provides the mapping information for properties recommended to be implemented in Redfish/Swordfish for NVMe and NVMe-oF devices and the corresponding reference information from the NVMe and NVMe-oF specifications.

2.2 Audience Assumptions

This document assumes that the reader of this document is familiar with NVMe and NVMe-oF technologies and concepts. It also assumes the reader has knowledge of the Redfish and Swordfish concepts.

3 Normative References

3.1 Overview

The documents listed in Table 3 are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

3.2 Approved references

Table 3: Approved normative references

Tag	Title (Version)	A uthor	URL
I SO-8601	Data elements and interchange formats – Information interchange – Representation of dates and times – Part 1: Basic rules	ISO / IEC	<http://www.iso.org/iso/home/store/catalogue_ics/catalogue_detail_ics.htm?csnumber=70907>
ISO -Direct	ISO/IEC Directives, Part 2 Principles and rules for the structure and drafting of ISO and IEC documents (Eigth Edition, 2018)	ISO / IEC	<https://www.iso.org/sites/directives/current/part2/index.xhtml>
Redfish	Redfish Scalable Platforms Management API Specification (v1.11.0)	DMTF	http://www.dmtf.org/sites/default/files/standards/documents/DSP0266_1.4.0.pdf
Sw ordfish	Swordfish Scalable Storage Management API Specification (v1.2.1)	SNIA	<https://www.snia.org/tech_activities/standards/curr_standards/swordfish>

Tag	Title (Version)	A uthor	URL
NVMe	NVMe Spec v1.4a	NVM Ex press	<https://nvmexpress.org/wp-content/uploads/NVM-Express-1_4a-2020.03.09-Ratified.pdf>
NVMe-oF	NVMe-oF Spec v1.1	NVM Ex press	<https://nvmexpress.org/wp-content/uploads/NVMe-over-Fabrics-1.1-2019.10.22-Ratified.pdf>

3.3 References under development

None defined in this document.

3.4 Other references

None defined in this document.

4 NVMe Model Overview

4.1 Introduction

In order to manage NVMe and NVMe-oF devices and systems in a large scale environment, a higher level management ecosystem is needed. The Redfish/Swordfish management specifications are designed to manage multi-system environments, including multiple types of fabrics, covering not only multiple technologies, but also inclusive of system management, storage management and fabric management, making it the ideal ecosystem in which to add not only the integration of NVMe devices for system and storage management, but NVMe-oF for fabric management.

This document describes how both the NVMe Subsystem model and the NVMe-oF fabric system model should be mapped consistently to Redfish and Swordfish constructs for implementations to be managed within Redfish and Swordfish management environments. This model and mapping information does not describe or assert any specific implementation recommendation technologies.

Similar implementations will have similar Redfish and Swordfish constructs. Mockups are used to show static examples of sample representations. Requirements and recommendations for implementations are provided separately through the Swordfish NVMe and NVMe-oF profiles. The profiles use the Redfish interoperability profile schema to specify the required, recommended and optional properties and schema for specific configurations and functionality that correspond to classes of implementations.

4.1.1 Fundamental Model Design Assertions

- There shall be a unified model across all types of NVMe devices.
- There shall not be a different model for “drives” vs other types of NVMe devices
- The model will cover an appropriate level of abstraction for all types of NVMe devices based on modeling and mockups reflected in the documented permutations (e.g., from simple drives through to complex fabric virtual systems)
  - Simple NVMe drives; complex NVMe drives; JBOFs/EBOFs; Arrays/RBOFs
- The logical model for NVMe-oF shall leverage the NVMe Subsystem model
- Logical subsystems, controllers, and namespaces are the same objects with the same relationships as in the NVMe Subystem Model. Unneeded objects are not instantiated (e.g., Endurance Groups, sets)

- The NVMe native model should map to the existing Redfish and Swordfish constructs when and where possible
  - Mapping future NVMe / NVMe-oF functionality should follow this principle when and where possible (e.g., firmware update mapping to the RF update service)

4.2 Overall NVMe Subsystem Model

Key Tenets:

- Model reflects a unified view of all NVMe device types.
- Devices will instantiate an appropriate subset of the model
- The model diagrams do not reflect all available schema elements.
- Model leverages and coarsely maps to existing (Redfish and) Swordfish storage model

4.2.1 Major NVM Objects Mapped to RF/SF

4.2.1.1 NVM Subsystem An NVM subsystem includes one or more controllers, zero or more namespaces, and one or more ports. Examples of NVM subsystems include Enterprise and Client systems that utilize PCI Express based solid state drives and/or fabric connectivity.

4.2.1.2 NVM Controller (IO, Admin and Discovery) The interface between a host and an NVM subsystem

Admin controller: controller that exposes capabilities that allow a host to manage an NVM subsystem

Discovery: controller that exposes capabilities that allow a host to retrieve a Discovery Log Page

I/O: controller that implements I/O queues and is intended to be used to access a non-volatile memory storage medium

4.2.1.3 Namespace A quantity of non-volatile memory that may be formatted into logical blocks. When formatted, a namespace of size n is a collection of logical blocks with logical block addresses from 0 to (n-1).

4.2.1.4 Endurance Group A portion of NVM in the NVM subsystem whose endurance is managed as a group

4.2.1.5 NVM Set An NVM Set is a collection of NVM that is separate (logically and potentially physically) from NVM in other NVM Sets.

4.2.1.6 NVM Domain A domain is the smallest indivisible unit that shares state (e.g., power state, capacity information). Domain members can be NVM controllers, endurance groups, sets or namespaces.

4.2.2 Unmapped objects

There are a number of objects that are required for the proper integration of NVMe support within Redfish and Swordfish, but which are not supported by an entity that can be mapped directly from the various NVM Specifications. Information about these related but un-mapped objects are defined elsewhere in the documentation provided with each Swordfish release.

They are summarized in Appendix A.

4.2.3 NVM Subsystem Model

The following diagram reflects the high level mapping of the key NVM objects into Redfish / Swordfish schema objects. These largely follow existing relationships used by the Swordfish storage specification for non-NVMe implementations as well, which provides a great deal of consistency for storage clients, as well as for implementations such as NVMe arrays that may be delivering solutions that combine NVMe and other technologies.

This model covers a wide range of instantiations ranging from individual SSDs, to multi-rack storage systems. All of these can be represented by this NVM Subsystem model, shown in Figure 1.

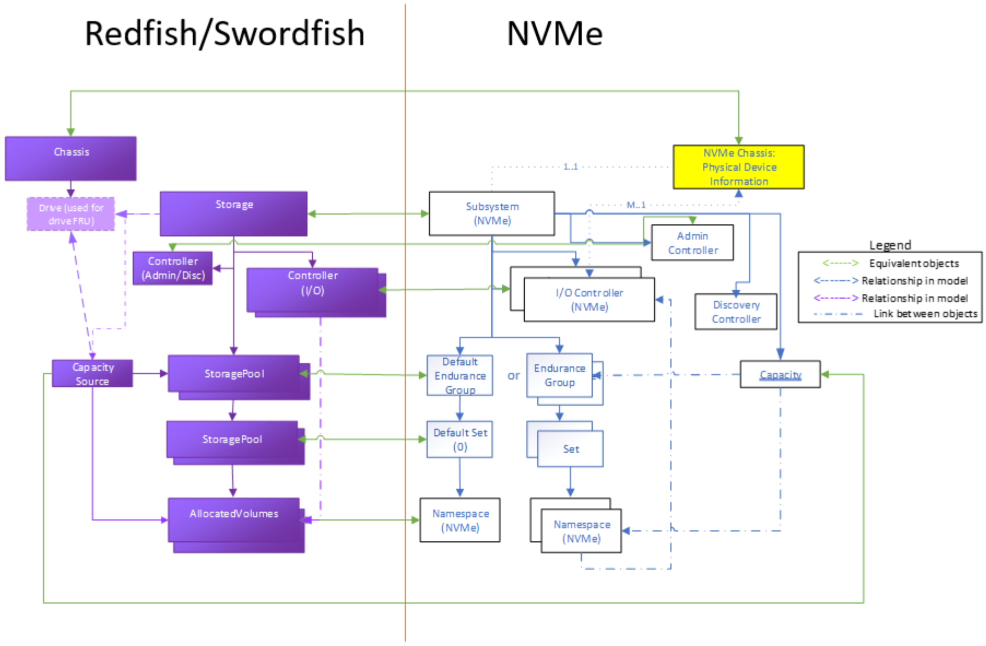


Figure 1: Subsystem model

4.2.4 NVMe-oF Subsystem Model

Figure 2 shows the high level mapping of the key NVMe-oF objects to Redfish / Swordfish schema objects. Following the tenets described in the model overview section, these extend the mapping used in the NVM Subsystem for the logical versions of the objects.

This model also includes the use of the Redfish Fabric model to cover the connectivity aspects of the fabric.

The grey shaded portion of this diagram reflects the logical / exported portion of the NVMe-oF environment represented in Redfish / Swordfish.

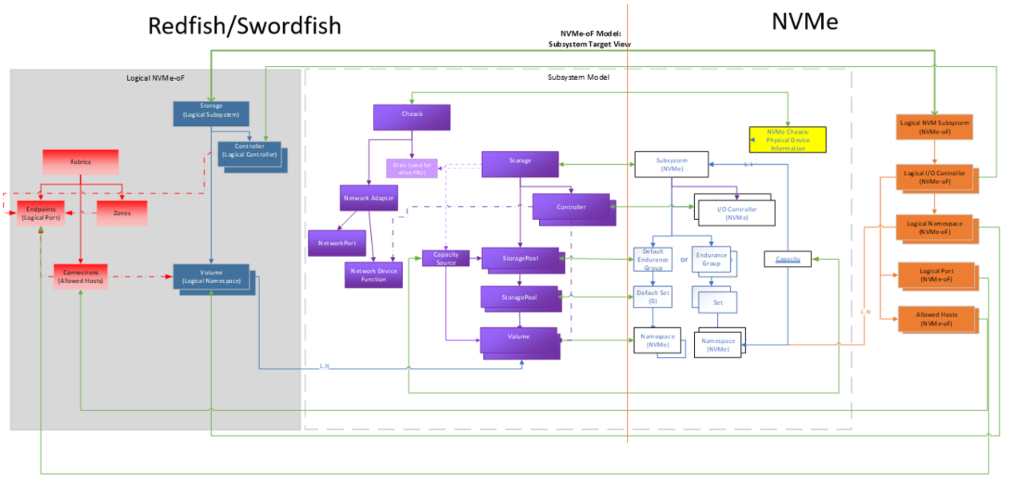


Figure 2: NVMe-oF Subsystem Model

5 Example Instances

5.1 Introduction

This section of the document provides a series of example usages of the model that represent common instantiations of NVMe devices, and how those devices may use the various NVMe, and correspondingly, Redfish and Swordfish objects and schema. This section will not provide comprehensive representations of all potential device types; rather, a representation of several common device types, in order to provide an illustration of the application of the model for those that are unfamiliar with either the NVMe or Redfish / Swordfish ecosystems.

Further, the following sections describe the examples and do not attempt to cover all potential permutations for alternate representations of each device class or possible implementations.

5.2 Simple SSD

5.2.1 Overview

Figure 3 shows a sample representation of a simple NVMe SSD, with a PCIe interface. It is implemented with no endurance group or NVM set functionality; it has only a single namespace capability, and a single IO controller.

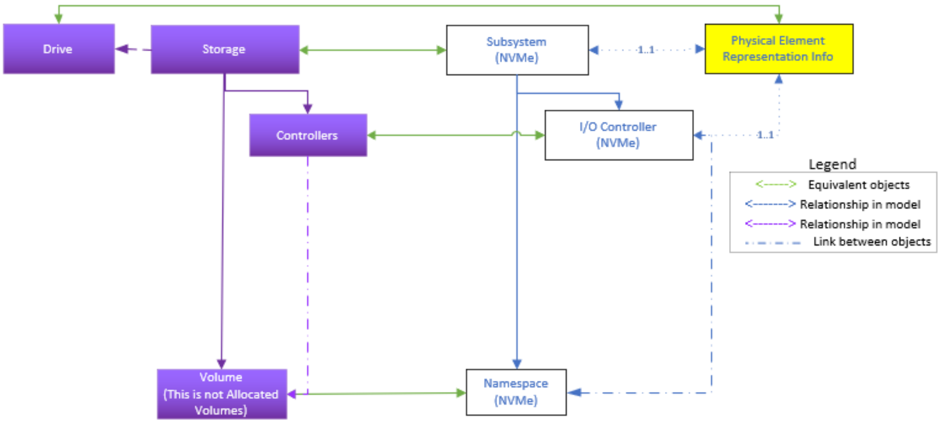

Drive
Storage
Subsystem (NVMe)
1..1
Physical Element Representation Info
Controllers
I/O Controller (NVMe)
1..1
Legend
<-----> Equivalent objects
<-------> Relationship in model
<-------> Relationship in model
Link between objects
Volume (This is not Allocated Volumes)
Namespace (NVMe)

Figure 3: Simple SSD instance diagram

5.2.2 Explanation of Object use

Simple SSDs are SSDs that do not use Endurance Groups or sets. Correspondingly, they only use Storage, Controllers, Volumes (Namespaces), and the Drive schema to represent the fundamental components.

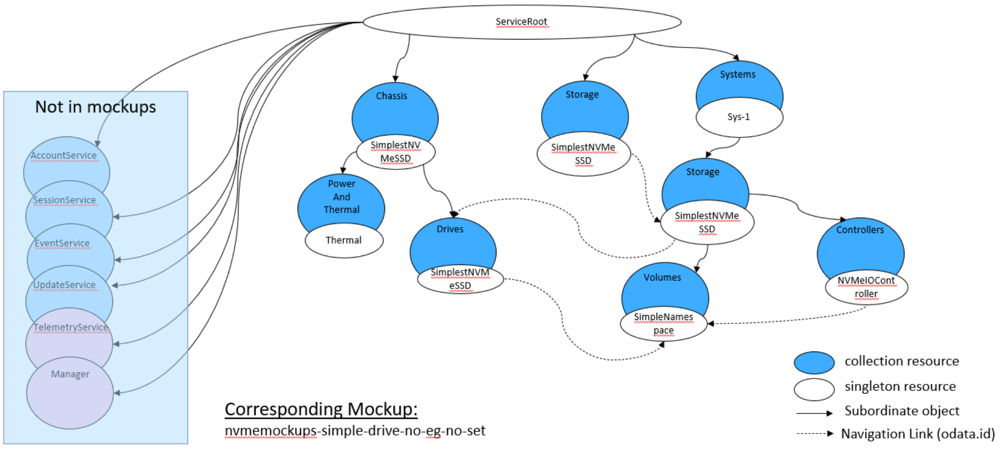
Many SSD implementations support exactly one namespace. These are described by this model, and the mockups reflect this configuration.

This model can also support extensions to cover dual-ported configurations, as well as support for multiple IO controllers per port.

5.2.3 Redfish / Swordfish Object Representation

Figure 4 shows the representation, as expressed in the mockup indicated below, of a sample instantiation using Redfish / Swordfish objects.

Note that this mockup does not represent a complete service instantiation; it contains only objects of interest for this context.



Corresponding Mockup:
nvmemockups-simple-drive-no-eg-no-set

Figure 4: Simple SSD mockup example

5.2.4 Mockup

A corresponding mockup for this configuration can be found at http://swordfishmockups.com/simple-ssd-mockups.

5.3 Complex SSD

5.3.1 Overview

Figure 5 shows a sample representation of a complex NVMe SSD, with a PCIe interface. This example shares many similarities to the simple device model, but adds the representation of NVMe Endurance Groups and NVM Sets using the Swordfish StoragePool schema, with additional NVMe specific properties.

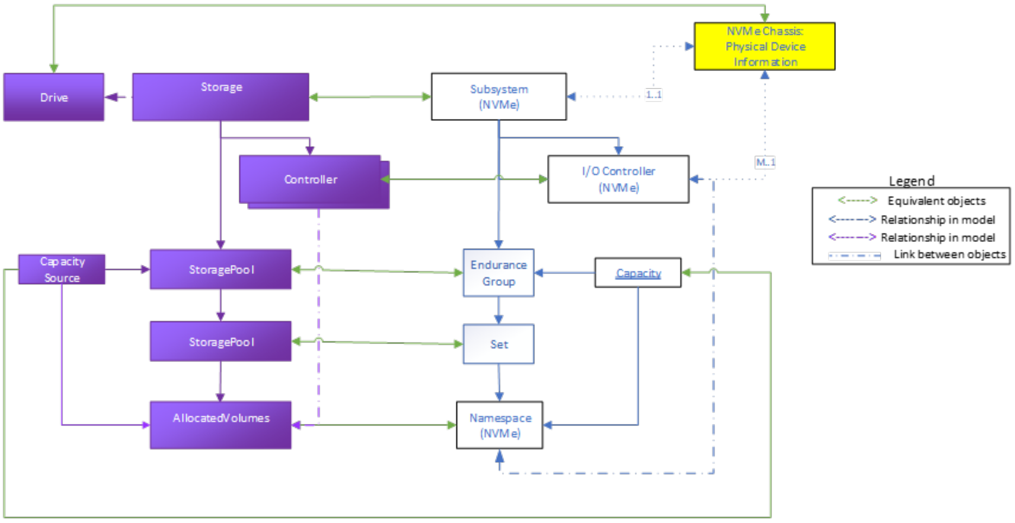


Figure 5: Complex SSD Model

5.3.2 Explanation of Object use

Complex SSDs are SSDs that use Endurance Groups and NVM sets. They also use Storage, Controllers, Volumes (Namespaces), and the Drive schema to represent the fundamental components.

This model can also support extensions to cover dual-ported configurations, as well as support for multiple IO controllers per port.

Endurance Groups divide the media into distinct wear-leveling domains. How this happens is implementation specific.

NVM Sets further subdivide an endurance group in order to limit performance interference within and across these domains.

When this type of device supports dynamic namespace allocation and NVM Sets, the management of the namespaces is done within an NVM Set as the underlying capacity source (e.g., the storage pool).

5.3.3 Redfish / Swordfish Object Representation

Figure 6 shows the representation, as expressed in the mockup indicated below, of a sample instantiation using Redfish / Swordfish objects.

Note that this mockup does not represent a complete service instantiation; it contains only objects of interest for this context.

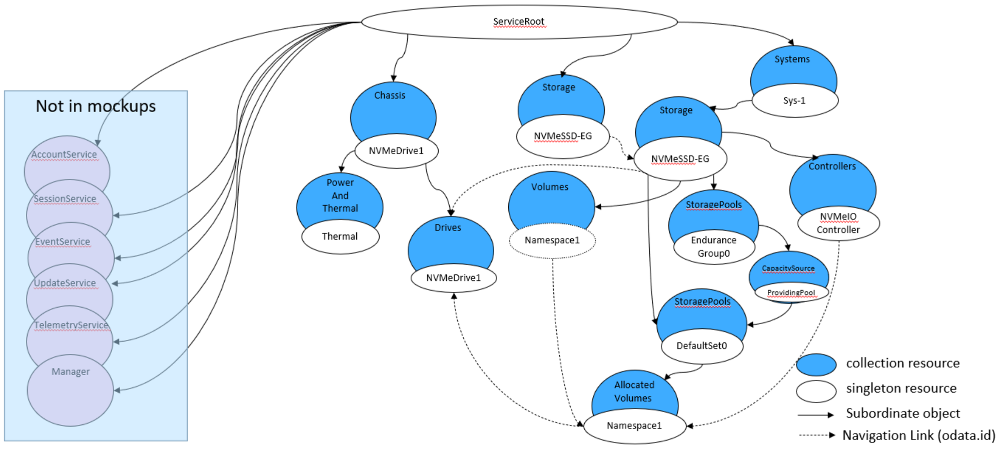


Figure 6: Complex SSD Model

5.3.3.1 Mockup A corresponding mockup for this configuration can be found at http://swordfishmockups.com/simple-ssd-eg-set-mockups.

5.4 Simple SSD with IP (NVMe-oF) Attach

5.4.1 Overview

This example reflects an IP-attached drive configuration, with a single ethernet port configured, as illustrated in Figure 7. It includes a drive configured with a default endurance group and NVM Set, and is instantiated in the Storage Collection off the Service Root.The network configuration is modeled in the Chassis.

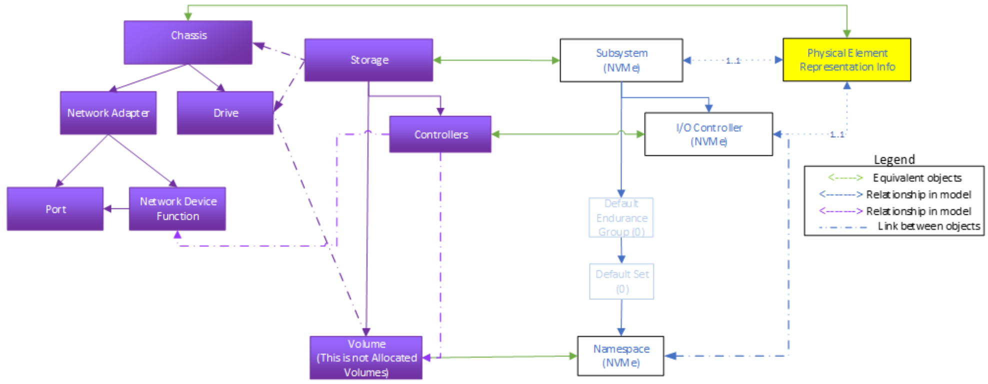


Figure 7: Simple IP-atteched SSD

5.4.2 Explanation of Object use

Simple SSDs with IP attach are also SSDs that do not use Endurance Groups or sets, but that have IP-based network interfaces. Correspondingly, they only use Storage, Controllers, Volumes (Namespaces), and the Drive schema to represent the fundamental components. In addition, they use the Redfish Network Adapter, Port and Network Device Function to model the configuration of the IP interface port(s).

As with the Simple SSD configuration, Many SSD with IP-attach implementations support exactly one namespace. These are described by this model, and the mockups reflect this configuration.

This model can also support extensions to cover multi-ported configurations, as well as support for multiple IO controllers per port.

5.4.3 Redfish / Swordfish Object Representation

Figure 8 shows the representation, as expressed in the mockup indicated below, of a sample instantiation using Redfish / Swordfish objects.

Note that this mockup does not represent a complete service instantiation; it contains only objects of interest for this context.

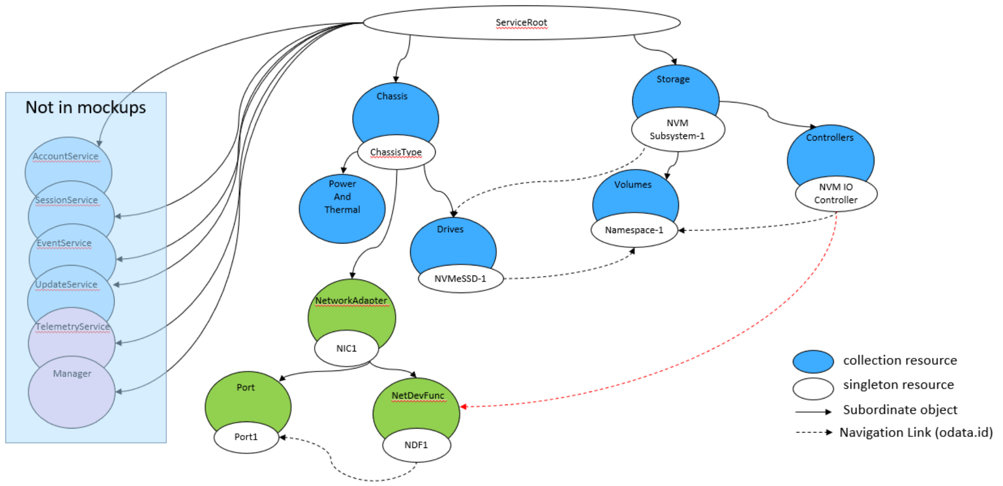


Figure 8: Simple IP-attached SSD mockup

5.4.4 Mockup

A corresponding mockup for this configuration can be found at http://swordfishmockups.com/ethernet-attach-drive-mockups.

5.5 JBOF

5.5.1 Overview

This example covers a representation of a JBOF ("just a bunch of flash") enclosure and contained drives. This mockup reflects a PCIe front-end attach configuration with a set of drives.

Figure 9 shows only the controller object representation for this JBOF configuration. This includes the admin controller function for enclosure management.

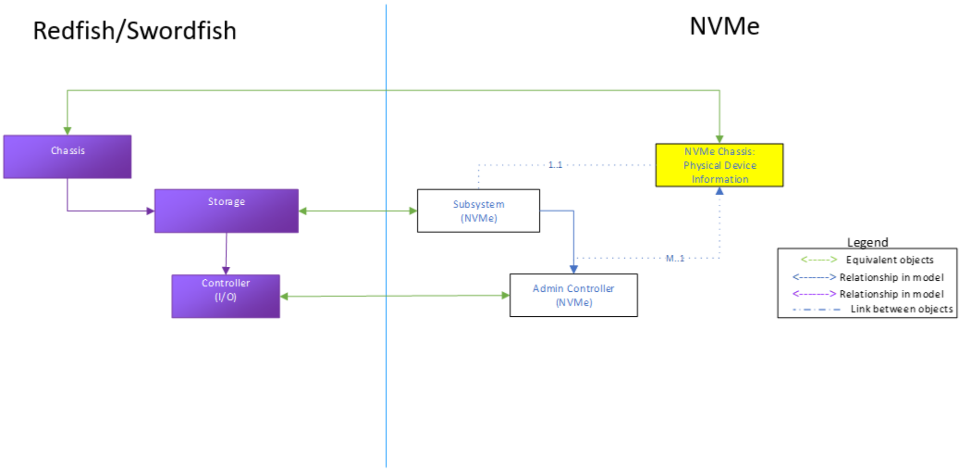


Figure 9: JBOF configuration controller object

Figure 10 shows the combined object representations for the JBOF system, with both the JBOF controller and NVMe drives (using the Simple SSD style drives) represented in the system. Note that the full mockup represented has 7 drives, while this diagram only represents two for the sake of visual clarity.

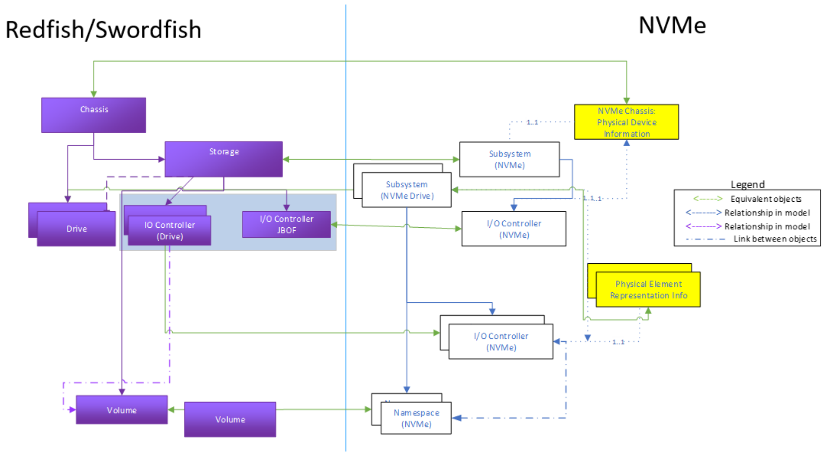


Figure 10: Full JBOF system

5.5.2 Explanation of Object use

This type of JBOF system uses the Chassis, Storage and Controller objects to reflect physical component modeling, Subsystem and Admin Controller functionality.

The Chassis model and Admin controller represent NVM's SES (SCSI enclosure services) usage.

5.5.3 Redfish / Swordfish Object Representation

Figure 11 shows the representation, as expressed in the mockup indicated below, of a sample instantiation using Redfish / Swordfish objects.

Note that this mockup does not represent a complete service instantiation; it contains only objects of interest for this context.

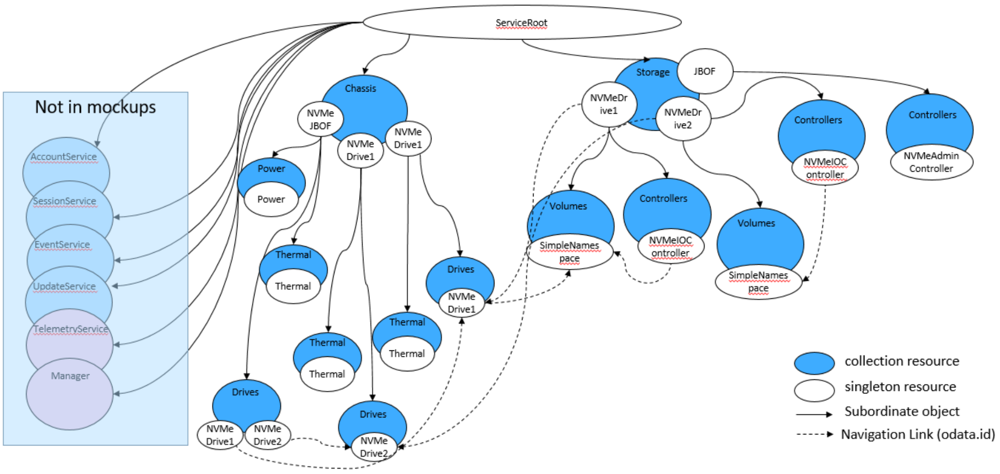


Figure 11: JBOF system instance

5.5.4 Mockup

A corresponding mockup for this configuration can be found at http://swordfishmockups.com/nvme-jbof-mockups

5.6 EBOF

5.6.1 Overview

This example covers a representation of a EBOF ("Ethernet bunch of flash") enclosure and contained drives. This mockup reflects a Ethernet front-end attach enclosure configuration containing a set of Ethernet-attach drives.

Figure 12 shows the combined object representations for the EBOF system, with both the EBOF controller and Ethernet-attach drives (using the Simple SSD with IP drives) represented in the system. Note that the full mockup represented has multiple drives, while this diagram only represents one for the sake of visual clarity.

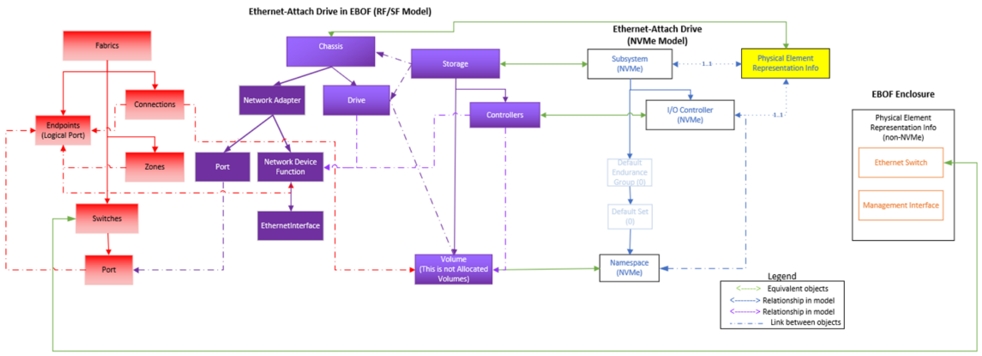

Ethernet-Attach Drive in EBOF (RF/SF Model)
Fabrics
Connections
Endpoints (Logical Port)
Zones
Switches
Port
Chassis
Storage
Network Adapter
Drive
Controllers
Port
Network Device Function
EthernetInterface
Volume (This is not Allocated Volumes)
Ethernet-Attach Drive (NVMe Model)
Subsystem (NVMe)
1..1
Physical Element Representation Info
I/O Controller (NVMe)
1..1
Default Endurance Group (0)
Default Set (0)
Namespace (NVMe)
EBOF Enclosure
Physical Element Representation Info (non-NVMe)
Ethernet Switch
Management Interface
Legend
<-----> Equivalent objects
<-------> Relationship in model
<-------> Relationship in model
Link between objects

Figure 12: Full EBOF system

5.6.2 Explanation of Object use

This type of EBOF system uses the Storage and Controller objects to reflect component modeling of the NVMe Subsystem and controller functionality.

Connectivity is represented through the Chassis model's NetworkAdapter, NetworkPort, NetworkDeviceFunction and EthernetInterface objects, the Fabric model's Switch and Port objects, as well as the Manager model's EthernetInterface and ManagerProtocol. In-band management capability is also modeled via Admin controllers, consistent with other NVMe devices.

5.6.3 Redfish / Swordfish Object Representation

Figure 13 shows the representation, as expressed in the mockup indicated below, of a sample instantiation using Redfish / Swordfish objects.

Note that this mockup does not represent a complete service instantiation; it contains only objects of interest for this context.

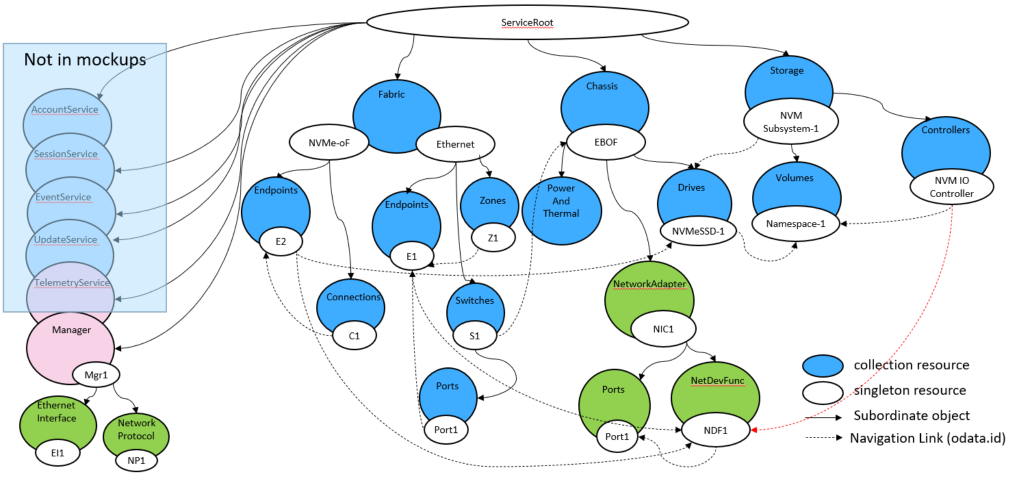


Figure 13: EBOF system instance

5.6.4 Mockup

A corresponding mockup for this configuration can be found at http://swordfishmockups.com/nvme-ebof-mockups

5.7 Opaque Array / NVMe Front End Device

5.7.1 Overview

The "opaque" array reflects a system with an NVMe front end, but the internal implementation is vendor specific, and not necessarily presented by the vendor (aka "opaque"). Figure 14 shows a system that presents an NVMe front-end but also exposes a SATA drive backend. This could be done to support both FRU management and volume/namespace creation.

Requirements for the NVMe portion of this type of configuration is documented in the SwordfishNVMeFrontEnd profile; opaque arrays should implement both traditional Swordfish block profiles, in addition to the SwordfishNVMeFrontEnd profile.

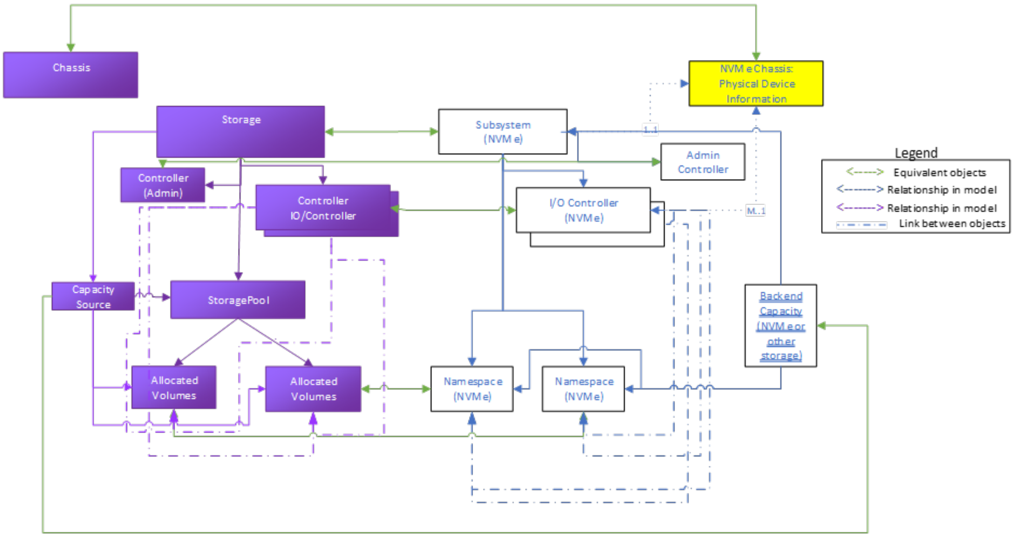


Figure 14: Opaque array example

5.7.2 Explanation of Object use

The opaque array example presents NVMe specific information in Redfish/Swordfish objects, using the Storage, Controller, and Volume objects.

In addition, device management information is presented through StoragePool and Drive objects, providing internal, non-NVMe implementation specific information to the user, for configuration, diagnosis and other storage management functions. (This set of objects is subject to the standard Swordfish specification and profiles.)

5.7.3 Redfish / Swordfish Object Representation

Figure 15 shows the representation, as expressed in the mockup indicated below, of a sample instantiation using Redfish / Swordfish objects.

Note that this mockup does not represent a complete service instantiation; it contains only objects of interest for this context.

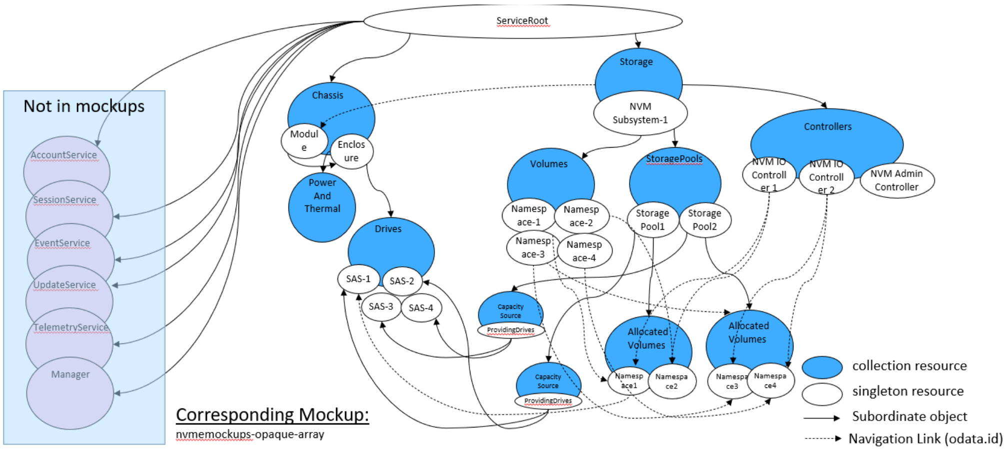


Figure 15: Sample opque system instance

5.7.4 Mockup

A corresponding mockup for this configuration can be found at http://swordfishmockups.com/nvme-opaque-array-mockups.

5.8 Subsystem (Fabric) Model - NVMe-oF: Fabric Attach Subsystem

5.8.1 Overview

Figure 16 shows a logical/exported NVMe-oF subsystem presenting one logical subsystem, one I/O controller, one namespace, one port and representing one allowed host, using the Redfish Fabric model. The fabric model uses the Connection schema to characterize the allowed host information, and Endpoints and Zones to show the network connectivity from the device's perspective.

This example also includes NVMeDomains. NVMeDomains contain a collection of domain members; these can be NVM controllers, endurance groups, NVM sets, or namespaces.

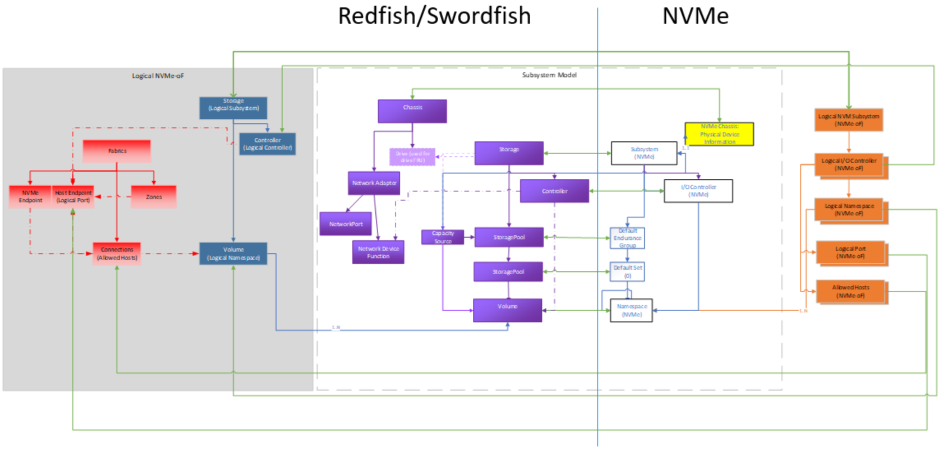


Figure 16: NVMe-OF subsystem example

5.8.2 Explanation of Object use

The fabric attach subsystem example shows the representation of logical, or exported, subsystems, controllers, and namespaces using the Storage, Controller and Volume objects respectively.

It also uses portions of the Redfish fabric model to represent the host attachment information - allowed hosts and logical port - using the Connections and Endpoints objects respectively.

5.8.3 Redfish / Swordfish Object Representation

Figure 17 shows the representation, as expressed in the mockup indicated below, of a sample instantiation using Redfish / Swordfish objects.

Note that this mockup does not represent a complete service instantiation; it contains only objects of interest for this context.

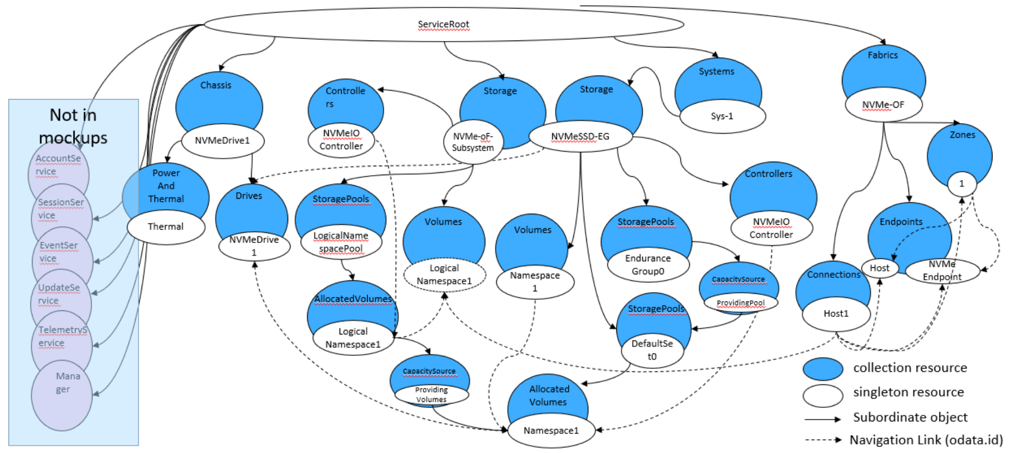


Figure 17: NVMe-oF system instance

5.8.4 Mockup

There are multiple mockups that show the representation for this configuration:

- A mockup for this configuration using an Ethernet attach front-end can be found at http://swordfishmockups.com/nvmeof-mockups.
- A mockup for this configuration using an RDMA attach front-end can be found at http://swordfishmockups.com/nvmeof-RDMA-mockups.

5.9 NVMe Domains

5.9.1 Overview

This example describes NVMeDomains. NVMeDomains contain a collection of domain members; these can be NVM controllers, endurance groups, NVM sets, namespaces, and ports, as illustrated in Figure 18.

Domains are used to subdivide an NVM Subsystem. For example, if there are multiple power sources, the domain is used to represent the scope of each power source.

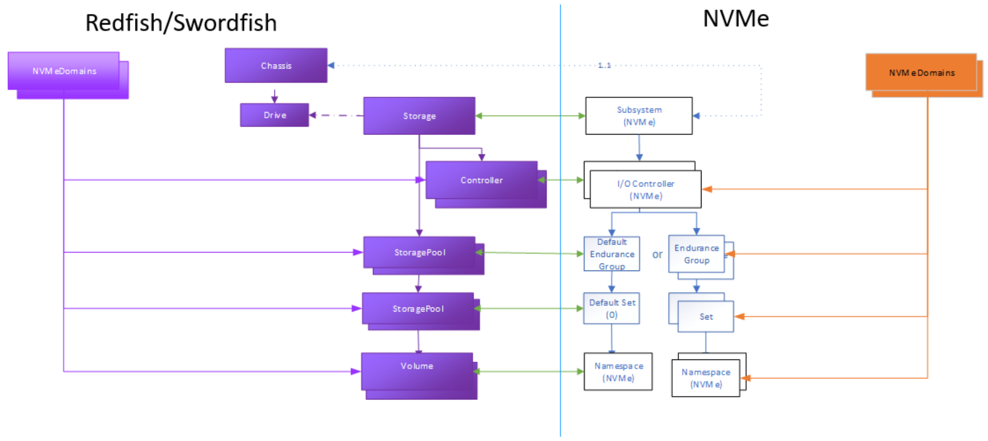


Figure 18: NVMeDomain example

5.9.2 Explanation of Object use

The domain object contains two primary elements: the domain members collection, which contains pointers to the relevant controllers, groups, sets, namespaces, and ports that reflect the appropriate subdivision for the purpose of the domain.

The other primary element includes a set of capacity information properties about this set of domain members.

5.9.3 Mockup

A corresponding mockup for this configuration can be found at http://swordfishmockups.com/nvmeof-mockups.

6 Property Mapping

6.1 Introduction

The property mapping provided defines the preferred translation between the Redfish/Swordfish schema objects and properties, and the corresponding NVMe and NVMe-oF specification properties. The information is ordered within the Redfish/Swordfish objects; each cross-referenced property within the Redfish/Swordfish structure therefore contains a detailed reference to its mapped property in the NVMe ecosystem.

Each section also includes a sample Redfish/Swordfish mockup presenting an example usage for that object.

6.2 Property Mapping Template

Table 4 provides the template and an example for the property mapping provided in the following sections of this document.

For each property (whether a reference, collection, complex type or actual property), there is a comparison between the property in Redfish/Swordfish to the corresponding property in either the NVMe or NVMe-oF specification. The RF/SF property is provided within its schema context; the NVMe/NVMe-oF specification reference is provided within the table, showing both which specification, as well as the section and, if appropriate, the figure in which the property is specified.

Similarly, the type of each property is correspondingly specified. The RF/SF type is specified, and the NVM Spec property type is shown, as well as, where appropriate, any additional identifying information, such as byte offset and data structure.

The Mandatory field is used to specify whether properties are Mandatory or Optional on the NVMe specification side, and in the rare instance where properties are Mandatory in the Redfish/Swordfish schema. (Recommended / required properties for specific implementation types in Redfish/Swordfish will be done separately, through the use of profiles.)

The Notes field can / will be used to include any relevant information about either the purpose of the property, additional context, or other useful information to implementers, such as inter-relationships with other properties.

Table 4: Property Mapping Template and Example

	Redfish/Swordfish	NVMe / NVMe-oF
Property	**Redfish / Swordfish Schema Property:** Reco mmendedArbitrationBurstSize	**NVM Spec Property / Field:** Recom mendedArbitrationBurst(RAB) **NVM Spec: Section:Figure** NVMe 1.4a: Section 5.15.2.2, Figure 249
Type	**Redfish / Swordfish Schema Type:** String	**NVM Spec Property Type:** Power of 2: 2ˆn **Additional NVM Spec Identifying Information:** ByteOffset: 72, IdentifyController data structure
Des cription	The Recommended Arbitration Burst Size indicates the maximum number of commands that the controller may launch at one time from a particular Submission Queue.	This is the recommended Arbitration Burst size. The value is in commands and is reported as a power of two (2ˆn). This is the same units as the Arbitration Burst size.
LongDes cription	This property shall contain the Recommended Arbitration Burst Size indicates the maximum number of commands that the controller may launch at one time from a particular Submission Queue. The value is expressed as a power of two (e.g., 000b indicates one, 011b indicates eight). A value of 111b indicates no limit.	
M andatory		Mandatory
Notes		

6.3 NVM subsystem

The Redfish/Swordfish Storage schema is used to represent an NVM Subsystem.

6.3.1 Mockup

The following mockup shows a sample representation of the Storage schema used to represent an NVM Subysystem.

```
{
  "@Redfish.Copyright": "Copyright 2014-2020 SNIA. All rights reserved.",
  "@odata.id": "/redfish/v1/Storage/NVMe-oF-Subsystem",
  "@odata.type": "#Storage.v1_9_0.Storage",
  "Id": "1",
  "Name": "NVMe-oF Logical NVM Fabric System",
  "Description": "Mockup of NVMe-oF Logical NVM Fabric System with 1 Logical
  ↪  Subsystem, 1 Logical I/O Controller and 1 Logical port and 1 allowed
  ↪  host.",
  "Status": {
    "State": "Enabled",
    "Health": "OK",
    "HealthRollup": "OK"
  },
  "Identifiers": [{
    "DurableNameFormat": "NQN",
    "DurableName":
    ↪  "nqn.2014-08.org.nvmexpress:uuid:6c5fe566-10e6-4fb6-aad4-8b4159f50245"
  }],
  "Controllers": {
    "@odata.id": "/redfish/v1/Storage/NVMe-oF-Subsystem/Controllers"
  },
  "Volumes": {
    "@odata.id":
    ↪  "/redfish/v1/Storage/NVMe-oF-Subsystem/Volumes/LogicalNamespace1"
  }
}
```

6.3.2 Property Mapping

6.3.2.1 Actions.#StorageController.SetEncryptionKey The mapping for `Actions.#StorageController.SetEncryptionKey` is summarized in Table 5.

Table 5: Actions.#StorageController.SetEncryptionKey mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Actions.#Storage Controller.SetEncryptionKey	N/A
Type	Action (Special form of POST)	
Des cription	The available OEM-specific actions for this resource.	
LongDes cription	This property shall contain the available OEM-specific actions for this resource.	
M andatory	Do not implement (for NVMe Drives, or for devices with an NVMe front end interface, e.g., opaque arrays).	
Notes		N/A for NVMe (drives or for devices with an NVMe front end interface). Drives will generate their own key for CryptoErase, this requires passing a new key.

6.3.2.2 Controllers

The mapping for `Controllers` is summarized in Table 6.

Table 6: Controllers mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Controllers	Controllers
Type	S torageControllerCollection.	Controller list.
Des cription	The set of controllers instantiated by this storage subsystem.	A list of controller identifiers in the NVM subsystem that may or may not be attached to namespace(s)
LongDes cription	This property shall contain a link to a Resource of type StorageControllerCollection that contains the set of storage controllers allocated to this instance of an storage subsystem.	A Controller List (refer to NVMe Bese Specification section 4.11) of up to 2,047 controller identifiers is returned containing a controller identifier greater than or equal to the value specified in the Controller Identifier (CDW10.CNTID) field. The list contains controller identifiers in the NVM subsystem that may or may not be attached to namespace(s).
M andatory	Yes. Implement (for NVMe Drives, or for devices with an NVMe front end interface, e.g., opaque arrays).	No (see note)
Notes	This is a collection of StorageControllers. Refer to the StorageController schema for details of the instance information. These are used to provide information on NVM IO, Admin and Discovery controllers.	This property is only mandatory for controllers that support the Namespace Management capability - reference NVMe Base Specification section 5.15.2.9 Controller list (CNS 13h)

6.3.2.3 Description The mapping for `Description` is summarized in Table 7.

Table 7: Description mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Description	N/A
Type	String	N/A
Des cription	The description of this resource.	N/A
LongDes cription	This object represents the description of this resource. The resource values shall comply with the Redfish Specification-described requirements.	N/A
M andatory	Yes	N/A
Notes	In Redfish, Description is a read-only field.	Return the common description: "An NVM Express Subsystem is an NVMe device that contains one or more NVM Express controllers and may contain one or more namespaces."

6.3.2.4 Drives The mapping for `Drives` is summarized in Table 8.

Table 8: Drives mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Drives	**NVM Spec Property / Field:** **NVM Spec: Section:Figure**
Type	Collection(Drive.Drive)	
Des cription	The set of drives attached to the storage controllers that this resource represents.	N/A
LongDes cription	This property shall contain a set of the drives attached to the storage controllers that this resource represents.	
M andatory	Required (for NVMe Drives). Optional to implement for devices with an NVMe front end interface, e.g., opaque arrays).	
Notes	For NVMe Drive implementation, this links to "Drive" object, which contains the physical representation of NVMe Drive information.	

6.3.2.5 Identifiers

The mapping for `Identifiers` is summarized in Table 9.

Table 9: Identifiers mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Identifiers	NVM Subsystem NVMe Qualified Name (SUBNQN)
Type	Coll ection(Resource.Identifier)	An array of identifiers
Des cription	The Durable names for the subsystem.	An array of identifiers
LongDes cription	This property shall contain a list of all known durable names for the associated subsystem.	This specifies the NVM Subsystem NVMe Qualified Name as a UTF-8 null-terminated string. Refer to NVMe Base Specification, section 7.9, for the definition of NVMe Qualified Name. Support for this field is mandatory if the controller supports revision 1.2.1 or later as indicated in the Version register (refer to section 3.1.2).
M andatory	Yes	Yes
Notes	This is an array of unique identifiers for the NVM Subsystem.	There will only be one instance in this array for Subsystem. Refer to the Identify Controller data structure (CNS 01h) bits 1023:768 in figure 249 (Identify – Identify Controller Data Structure) of the NVMe Base Specification.

6.3.2.6 Identifiers.DurableNameFormat The mapping for `Identifiers.DurableNameFormat` is summarized in Table 10.

Table 10: Identifiers.DurableNameFormat mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Id entifiers.DurableNameFormat	NVM Subsystem NVMe Qualified Name (SUBNQN)
Type	Resourc e.v1_1_0.DurableNameFormat	There is a single value for this array in Subsystem. The property type is of type NVMe Qualified Name (NQN).
Des cription	The format of the Durable names for the subsystem.	NVM Subsystem NVMe Qualified Name (SUBNQN)
LongDes cription	This specifies the format of the associated NVM Subsystem NVMe Qualified Name of type NQN. Support for this field is mandatory if the controller supports revision 1.2.1 or later as indicated in the Version register (refer to section 3.1.2).	
M andatory	No	Yes
Notes	This is an enum with multiple potential values. For this particular usage in Subsystem, there will only be one instance populated, of type NQN.	There will only be one instance in this array for Subsystem. Refer to the Identify Controller data structure (CNS 01h) bits 1023:768 in figure 249 (Identify – Identify Controller Data Structure) of the NVMe Base Specification.

6.3.2.7 Identifiers.DurableName The mapping for `Identifiers.DurableName` is summarized in Table 11.

Table 11: Identifiers.DurableName mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Identifiers.DurableName	NVM Subsystem NVMe Qualified Name (SUBNQN)
Type	Edm.String	The NVM Subsystem NVMe Qualified Name as a UTF-8 null-terminated string
Des cription	The format of the Durable names for the subsystem.	NVM Subsystem NVMe Qualified Name (SUBNQN)
LongDes cription	This specifies the NVM Subsystem NVMe Qualified Name as a UTF-8 null-terminated string. Refer to NVMe Base Specification, section 7.9, for the definition of NVMe Qualified Name. Support for this field is mandatory if the controller supports revision 1.2.1 or later as indicated in the Version register (refer to section 3.1.2).	
M andatory	No	Yes
Notes	For this particular usage in Subsystem, there will only be one instance populated in the identifiers array.	There will only be one instance in this array for Subsystem. Refer to the Identify Controller data structure (CNS 01h) bits 1023:768 in figure 249 (Identify – Identify Controller Data Structure) of the NVMe Base Specification.

6.3.2.8 Links.Enclosures The mapping for `Links.Enclosures` is summarized in Table 12.

Table 12: Links.Enclosures mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Links.Enclosures	N/A
Type	Collection(Chassis.Chassis)	N/A
Des cription	An array of links to the chassis to which this storage subsystem is attached.	N/A
LongDes cription	This property shall contain an array of links to resources of type Chassis that represent the physical containers attached to this resource.	N/A
M andatory	Required	N/A
Notes	For NVMe Drive implementation, this links to a chassis collection that contains the subsystem's "Drive" object, which contains the physical representation of NVMe Drive information. For devices with an NVMe front end interface, e.g., opaque arrays), this refers to the appropriate chassis instance for the device/system; there may be multiple chassis instances, reflecting different physical entities in the system, such as controllers, drive enclosures, racks, etc).	The functionality comes from the underlying implementation and does not originate in the NVMe specs

6.3.2.9 Links.Enclosures@odata.count The mapping for `Links.Enclosures@odata.count` is summarized in Table 13.

Table 13: Links.Enclosures@odata.count mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Lin ks.Enclosures@odata.countt	N/A
Type	(odata property)int64	N/A
Des cription	Count of the number of items in the Links.Enclosures array.	N/A
LongDes cription		N/A
M andatory	Required	N/A
Notes		The functionality comes from the underlying implementation and does not originate in the NVMe specs

6.3.2.10 Links.HostingStorageSystems The mapping for `Links.HostingStorageSystems` is summarized in Table 14.

Table 14: Links.Enclosures mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Links.HostingStorageSystems	N/A
Type	Collection(Com puterSystem.ComputerSystem)	
Des cription	The storage systems that host this storage subsystem.	N/A
LongDes cription	This property shall contain an array of links to resources of type ComputerSystem that represent the storage systems that host this storage subsystem. The members of this array shall be in the StorageSystems resource collection off the service root.	N/A
M andatory	Recommended for devices with an NVMe front end interface such as opaque arrays.	N/A
Notes	For devices with an NVMe front end interface, e.g., opaque arrays), this refers to the ComputerSystem instances providing the NVMe front end, modeling the device’s controller(s).	The functionality comes from the underlying implementation and does not originate in the NVMe specs

6.3.2.11 Links.SimpleStorage The mapping for `Links.SimpleStorage` is summarized in Table 15.

Table 15: Links.SimpleStorage mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Links.SimpleStorage	
Type	SimpleStorage.SimpleStorage	
Des cription	The link to the simple storage instance that corresponds to this storage.	
LongDes cription	This property shall contain a link to a resource of type SimpleStorage that represents the same storage subsystem as this resource.	
M andatory	Do not implement	
Notes		

6.3.2.12 Name The mapping for `Name` is summarized in Table 16.

Table 16: Name mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Name	NVM Subsystem NVMe Qualified Name (SUBNQN)
Type	String	String
Des cription	The name of the resource or array member.	Uniquely describes the NVM subsystem.

	Redfish/Swordfish	NVMe / NVMe-oF
LongDescription	This object represents the name of this resource or array member. The resource values shall comply with the Redfish Specification-described requirements. This string value shall be of the 'Name' reserved word format.	The NVM Subsystem NVMe Qualified Name is a UTF-8 null-terminated string used (e.g., by host software) as the unique identifier for the NVM subsystem
M andatory	Yes	Yes (see note)

	Redfish/Swordfish	NVMe / NVMe-oF
Notes	In Redfish, Name is a read-only field.	Support for this field is mandatory if the controller supports revision 1.2.1 or later as indicated in the Version register (refer to section 3.1.2). Reported in the NVM Subsystem NVMe Qualified Name field of the Identify Controller data structure, bytes 1023:768 (refer to figure 249 in section 5.15.2.1 of the NVMe Base Specification). If the NVM Subsystem NVMe Qualified Name field of the Identify Controller data structure is not supported, then all bytes of this field shall be cleared to 0h. Refer to NVMe Base Specification section 7.9 for the definition of NVMe Qualified Name. Refer to NVMe Base Specifiction section 7.11 for details on the Unique Identifier, including compatibility with older versions of NVMe Controllers that do not support NVM Subsystem NQNs.

6.3.2.13 Status.State The mapping for `Status.State` is summarized in Table 17.

Table 17: Status.State mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Status.State	N/A
Type	Resource.State (enum)	N/A
Des cription	The known state of the resource, such as, enabled.	N/A
LongDes cription	This property shall indicate whether and why this component is available. Enabled indicates the resource is available. Disabled indicates the resource has been intentionally made unavailable but it can be enabled. Offline indicates the resource is unavailable intentionally and requires action to make it available. InTest indicates that the component is undergoing testing. Starting indicates that the resource is becoming available. Absent indicates the resource is physically unavailable.	
M andatory	Optional for NVMe drives; recommended to implement for NVMe front end devices such as opaque arrays.	No

	Redfish/Swordfish	NVMe / NVMe-oF
Notes	Possible values: Enabled / Disabled / StandbyOffline / StandbySpare / InTest / Starting / Absent / UnavaialableOffline / Deferring / Quiesced / Updating / Qualified	In general, there is no simple corresponding property or mappable set of information at this time from the NVMe Specifications. Current guidance for NVMe drives: do not implement this property. Guidance will be added in a future version of this document as this is an important concept for clients and for consistency with traditional storage devices. For opaque arrays and other similar devices with an NVMe front end, support and map these properties to the device's concepts of Enabled/Disabled/etc.

6.3.2.14 Status.Health The mapping for `Status.Health` is summarized in Table 18.

Table 18: Status.Health mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Status.Health	Critical Warning Condition
Type	Resource.Health	Boolean
Des cription	The health state of this resource in the absence of its dependent resources.	Indicates the NVM subsystem has detected a condition that causes at least one of bits 0 to 4 in the Critical Warning field of the SMART / Health Information log (refer to NVMe Base Specification section 5.14.1.2) to be set to one.
LongDes cription	This property shall represent the health state of the resource without considering its dependent resources. The values shall conform to those defined in the Redfish Specification.	Bits in this field represent the associated state at the time of this event. The Additional Hardware Error Information field shall be set at the time of the event using the same format as is specified for the Critical Warning field of the SMART / Health Information.
M andatory	Yes	Yes

	Redfish/Swordfish	NVMe / NVMe-oF
Notes	Possible Values: OK / Warning / Critical	Returned as a Critical Warning Condition (code 06h) in the NVM Subsystem Hardware Error Event data (bytes 01:00) of an NVM Subsystem Hardware Error Event (Event Type 05h) in the Persistent Event Log. Reference NVMe Base Specification 5.14.1.13.1.5 NVM Subsystem Hardware Error Event (Event Type 05h), Figure 221 and Figure 222. Implementations of more complex systems, such as opaque arrays and other similar devices with an NVMe front end, may also map this property to the device's concepts of OK/Warning/Critical.

6.3.2.15 Status.HealthRollup The mapping for `Status.HealthRollup` is summarized in Table 19.

Table 19: Status.HealthRollup mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Status.HealthRollup	Critical Warning
Type	Resource.Health	Boolean
Des cription	The overall health state from the view of this resource.	Indicates the NVM subsystem reliability has been degraded due to significant media related errors or any internal error that degrades NVM subsystem reliability.
LongDes cription	This property shall represent the health state of the resource and its dependent resources. The values shall conform to those defined in the Redfish Specification.	Indicates if the NVM subsystem reliability has been degraded due to significant media related errors or any internal error that degrades NVM subsystem reliability. Critical warnings regarding the health of the NVM subsystem may be indicated via an asynchronous event notification to the host. The warnings that results in an asynchronous event notification to the host are configured using the Set Features command; refer to section 5.21.1.11.
M andatory	Yes	Yes

	Redfish/Swordfish	NVMe / NVMe-oF
Notes	Possible Values: OK / Warning / Critical	Returned in byte 00, bit 1 of the Get Log Page – SMART / Health Information Log. Reference the NVMe Base Specification section 5.14.1.2 - SMART / Health Information (Log Identifier 02h), Figure 196. Implementations of more complex systems, such as opaque arrays and other similar devices with an NVMe front end, may also map this property to the device’s concepts of OK/Warning/Critical.

6.3.2.16 StorageControllers The mapping for `StorageControllers` is summarized in Table 20.

Table 20: StorageControllers mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	StorageControllers	**NVM Spec Property / Field: NVM Spec: Section:Figure**
Type	Storage.StorageControllers	
Des cription	The set of storage controllers that this resource represents.	N/A
LongDes cription	This property shall contain a set of the storage controllers that this resource represents.	
M andatory	Do not implement	
Notes	Deprecated for NVMe use - replaced by Controllers (type StorageCon troller.StorageController).	

6.3.2.17 StorageGroups The mapping for `StorageGroups` is summarized in Table 21.

Table 21: StorageGroups mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	StorageGroups	**NVM Spec Property / Field: NVM Spec: Section:Figure**
Type	StorageGroup.StorageGroup	
Des cription	All of the storage groups, each of which contains a set of volumes and endpoints that are managed as a group for mapping and masking, that belong to this storage subsystem.	N/A
LongDes cription	This property shall contain a link to a resource collection of type StorageGroupsCollection. This property shall be used when implementing mapping and masking.	
M andatory	Do not implement	
Notes	N/A for NVMe use cases. Deprecated by Connections.	

6.3.2.18 Volumes The mapping for `Volumes` is summarized in Table 22.

Table 22: Volumes mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Volumes	Allocated Namespace ID list
Type	Volume Collection.VolumeCollection	List of namespace IDs
Des cription	The set of volumes instantiated by this storage subsystem.	A list of Allocated Namespaces for this Subsystem
LongDes cription	This property shall contain a link to a Resource of type VolumeCollection that contains the set of storage volumes allocated to this instance of an storage subsystem.	A list of namespace IDs is returned to the host containing allocated NSIDs in increasing order.
M andatory	Required for NVMe drives, as well as opaque arrays and other similar devices with an NVMe front end.	No
Notes	This is a collection of Namespaces that belong to this NVM Subsystem. Refer to the Volume schema for details of the instance information.	Reference NVMe Base Specification section 5.15.2.6 Allocated Namespace ID list (CNS 10h).

6.4 NVM Controllers

The Redfish/Swordfish StorageControllers schema is used to represent an NVM Controller.

There are three different types of NVM Controllers: Admin, Discovery, and IO.

6.4.1 Admin Controller

6.4.1.1 Mockup The following mockup shows a sample representation of the StorageController schema used to represent an Admin Controller.

```
{
  "@Redfish.Copyright": "Copyright 2014-2020 SNIA. All rights reserved.",
  "@odata.id":
  ↪  "/redfish/v1/Storage/OpaqueArray/Controllers/NVMeAdminController",
  "@odata.type": "#StorageController.v1_0_0.StorageController",

  "Name": "NVMe Admin Controller",
  "Description": "Single NVMe Admin Controller for in-band admin command
  ↪  access.",
  "Status": {
    "State": "Enabled",
    "Health": "OK"
  },
  "Id": "NVMeAdminController",
  "Manufacturer": "Best NVMe Vendor",

  "Model": "NVMe Connect Array",
  "SerialNumber": "NVME123456",
  "PartNumber": "NVM44",
  "FirmwareVersion": "1.0.0",
  "SupportedControllerProtocols": [
    "PCIe"
  ],
  "NVMeControllerProperties": {
    "ControllerType": "Admin",
    "NVMeVersion": "1.3",
    "NVMeControllerAttributes": {
      "SupportsSQAssociations": false,
      "SupportsTrafficBasedKeepAlive": false,
      "SupportsExceedingPowerOfNonOperationalState": false,
      "Supports128BitHostId": false
    }
  }
}
```

6.4.1.2 Property Mapping

6.4.1.2.1 Assembly The mapping for `Assembly` is summarized in Table 23.

Table 23: Assembly mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Des cription	The Assembly schema defines an assembly. Assembly information contains details about a device, such as part number, serial number, manufacturer, and production date. It also provides access to the original data for the assembly.	N/A
LongDes cription	This Resource shall represent an assembly for a Redfish implementation. Assembly information contains details about a device, such as part number, serial number, manufacturer, and production date. It also provides access to the original data for the assembly.	N/A
M andatory	Do Not Implement for NVMe drives, or devices with NVMe front ends, such as opaque arrays.	
Notes		

6.4.1.2.2 AssetTag The mapping for `AssetTag` is summarized in Table 24.

Table 24: Assembly mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	AssetTag	**NVM Spec Property / Field:** N/A **NVM Spec: Section:Figure** N/A
Type	Edm.String	N/A
Des cription	The user-assigned asset tag for this storage controller.	N/A
LongDes cription	This property shall track the storage controller for inventory purposes.	N/A
M andatory	Do Not Implement for NVMe Drives, or devices with NVMe front ends, such as opaque arrays.	
Notes		

6.4.1.2.3 CacheSummary The mapping for `CacheSummary` is summarized in Table 25.

Table 25: CacheSummary mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	CacheSummary	N/A
Type	ComplexType	N/A
Des cription	The cache memory of the storage controller in general detail.	N/A
LongDes cription	This property shall contain properties that describe the cache memory for this resource.	N/A
M andatory	Do Not Implement	
Notes		This property exists for hw cache reporting in other RF/SF use cases. Not used in NVMe controllers.

6.4.1.2.4 ControllerRates The mapping for `ControllerRates` is summarized in Table 26.

Table 26: ControllerRates mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	ControllerRates	N/A
Type	ComplexType	N/A
Des cription	This property describes the various controller rates used for processes such as volume rebuild or consistency checks.	N/A
LongDes cription	This object shall contain all the rate settings available on the controller.	N/A
M andatory	Do Not Implement	
Notes		

6.4.1.2.5 Description The mapping for `Description` is summarized in Table 27.

Table 27: Description mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Description	N/A
Type	String	
Des cription	The description of this resource.	
LongDes cription	This object represents the description of this resource. The resource values shall comply with the Redfish Specification-described requirements.	
M andatory	Optional	N/A
Notes	In Redfish, Description is a read-only field.	Return the common description: “An NVM Admin Controller exposes capabilities that allow a host to manage an NVM subsystem. Admin controllers support commands providing management capabilities but does not provide IO access.”

6.4.1.2.6 FirmwareVersion The mapping for `FirmwareVersion` is summarized in Table 28.

Table 28: FirmwareVersion mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	FirmwareVersion	**NVM Spec Property / Field:** IdentifyController: Firmare Revision (FR) **NVM Spec: Section:Figure** 249: 71:64
Type	String	
Des cription	The firmware version of this storage controller.	
LongDes cription	This property shall contain the firmware version as defined by the manufacturer for the associated storage controller.	
M andatory	Required.	Required
Notes		Return the currently active firmware revision information.

6.4.1.2.7 Identifiers The mapping for `Identifiers` is summarized in Table 29.

Table 29: Identifiers mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Identifiers	N/A
Type	Coll ection(Resource.Identifier)	N/A
Des cription	The Durable names for the storage controller.	N/A
LongDes cription	This property shall contain a list of all known durable names for the associated storage controller.	N/A
M andatory	Recommend not implementing. There isn't a good mapping for these in the NVMe spec to a property that has an appropriate / mapping to a durable name format.	No
Notes		N/A

6.4.1.2.8 Identifiers.DurableName The mapping for `Identifiers.DurableName` is summarized in Table 30.

Table 30: Identifiers.DurableName mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Identifiers.DurableName	N/A
Type	Variable - see notes	N/A
Des cription	The Durable names for the storage controller.	N/A
LongDes cription	T	N/A
M andatory	Recommend not implementing.	No
Notes	Recommend not implementing. There isn't a good mapping for these in the NVMe spec to a property that has an appropriate / mapping to a durable name format.	N/A

6.4.1.2.9 Identifiers.DurableNameFormat The mapping for `Identifiers.DurableNameFormat` is summarized in Table 31.

Table 31: Identifiers.DurableNameFormat mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Id entifiers.DurableNameFormat	N/A
Type	enum (DurableNameFormat)	N/A
Des cription	The Durable names for the storage controller.	N/A
LongDes cription	This property shall contain a list of the types for all known durable names for the associated storage controller. The type determines the length of the corresponding Namespace ID	N/A
M andatory	Recommend not implementing.	No
Notes	Recommend not implementing. There isn’t a good mapping for these in the NVMe spec to a property that has an appropriate / mapping to a durable name format.	N/A

6.4.1.2.10 Links.AttachedVolumes The mapping for `Links.AttachedVolumes` is summarized in Table 32.

Table 32: Links.AttachedVolumes mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Links.AttachedVolumes	N/A
Type	Collection(Volume.Volume)	N/A
Des cription	An array of links to volumes that are attached to this controller instance.	
LongDes cription	This property shall contain a link to the Resources of type Volume that are attached to this instance of storage controller.	
M andatory	Do not implement (for admin controllers).	N/A
Notes		

6.4.1.2.11 Links.NetworkDeviceFunctions

The mapping for `Links.NetworkDeviceFunctions` is summarized in Table 33.

Table 33: Links.NetworkDeviceFunctions mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	L inks.NetworkDeviceFunctions	N/A
Type	C ollection(NetworkDeviceFunction.NetworkDeviceFunction)	
Description	The network device functions that provide connectivity to this controller.	
LongDescription	This property shall contain an array of links to resources of type NetworkDeviceFunction that represent the devices that provide connectivity to this controller.	
M andatory	Recommended to implement for more complex devices with NVMe front ends, such as opaque arrays.	
Notes	For NVMe-oF configurations.	

6.4.1.2.12 Location The mapping for `Location` is summarized in Table 34.

Table 34: Location mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Location	**NVM Spec Property / Field:** N/A **NVM Spec: Section:Figure** N/A
Type	Co llection(Resource.Location)	
Des cription	The location of the storage controller.	
LongDes cription	This property shall contain location information of the associated storage controller.	N/A
M andatory	Do Not Implement for NVM Drives or more complex devices with an NVMe front end, such as opaque arrays.	
Notes		

6.4.1.2.13 Manufacturer The mapping for `Manufacturer` is summarized in Table 35.

Table 35: Manufacturer mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Manufacturer	NVM Spec Property / Field:** IdentifyController / PCI Vendor ID (VID) **NVM Spec: Section: Figure** 249 byte 00:01
Type	String	
Des cription	The manufacturer of this storage controller.	The company vendor identifier
LongDes cription	This property shall contain the name of the organization responsible for producing the storage controller. This organization might be the entity from whom the storage controller is purchased, but this is not necessarily true.	The company vendor identifier that is assigned by the PCI SIG. This is the same value as reported in the ID register
M andatory	Recommended	
Notes		End clients expect to see the name of the company (e.g,; Contoso, BestVendor). While the value may be filled from the IdentifyController PCI Vendor ID or SubsystemID field, it would be preferable to have this filled with the actual string value of the company name.

6.4.1.2.14 Model The mapping for `Model` is summarized in Table 36.

Table 36: Model mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Model	**NVM Spec Property / Field:** IdentifyController / Model Number (MN) **NVM Spec: Section: Figure** 249 byte 24:63
Type	String	
Des cription	The model number for the storage controller.	
LongDes cription	This property shall contain the name by which the manufacturer generally refers to the storage controller.	Contains the model number for the NVM subsystem that is assigned by the vendor as an ASCII string. Refer to section 7.10 for unique identifier requirements. Refer to section 1.5 for ASCII string requirements
M andatory	Recommended	
Notes		

6.4.1.2.15 Name The mapping for `Name` is summarized in Table 37.

Table 37: Name mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Name	**NVM Spec Property / Field:** Controller ID (CNTLID) **NVM Spec: Section:Figure** NVMe 1.4a: Section 5.15.2.2 (IdentifyController), Figure 249

	Redfish/Swordfish	NVMe / NVMe-oF
Type	String	**NVM Spec Property Type:** 16-bit hex value **Additional NVM Spec Identifying Information:** ByteOffset: 79:78, IdentifyController data structure
Des cription	The name of the resource or array member.	
LongDes cription	This object represents the name of this resource or array member. The resource values shall comply with the Redfish Specification-described requirements. This string value shall be of the 'Name' reserved word format.	
M andatory	Required	Mandatory
Notes	In Redfish, Name is a read-only field.	Map the CNTLID field to a string with the format: "0xABCD"

6.4.1.2.16 NVMeControllerProperties.ControllerType The mapping for `NVMeControllerProperties.ControllerType` is summarized in Table 38.

Table 38: NVMeControllerProperties.ControllerType mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeControll erProperties.ControllerType	**NVM Spec Property / Field:** Controller Type (CNTRL_TYPE) **NVM Spec: Section:Figure** Section 5.15.2.2 (IdentifyController), Figure 249 Byte 111
Type	StorageController .v1_0_0.NVMeControllerType	Hex value
Des cription	This property specifies the type of NVMe Controller.	Controller Type
LongDes cription	This property shall specify the type of NVMe Controller.	This field specifies the controller type. A value of 0h indicates that the controller type is not reported.
M andatory	Required	Required
Notes	This property must be used to specify the type of NVMe Controller. For an admin controller, set to Admin.	For Admin Controller - value in Identify Controller is ‘03h’. Return “Admin”

6.4.1.2.17 NVMeControllerProperties.NVMeVersion

The mapping for `NVMeControllerProperties.NVMeVersion` is summarized in Table 39.

Table 39: NVMeControllerProperties.NVMeVersion mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeContr ollerProperties.NVMeVersion	**NVM Spec Property / Field:** Version (VER) **NVM Spec: Section:Figure** NVMe 1.4a: Section 5.15.2.2 (IdentifyController), Figure 249
Type	String	**NVM Spec Property Type:** Maps from register 3.1.2. **Additional NVM Spec Identifying Information:** ByteOffset: 83:90
Des cription	The version of the NVMe Base Specification supported.	
LongDes cription	This property shall specify the type of NVMe Controller.	
M andatory		Mandatory
Notes		Implementations compliant to revision 1.2 or later of this specification shall report a non-zero value in this field. Map from corresponding value in register 3.1.2 to string as “1.0”, “1.1”, “1.2”, “1.2.1”, “1.3.0”, “1.4.0”, etc.

6.4.1.2.18 NVMeControllerProperties.NVMeControllerAttributes.ReportsNamespaceGranularity

The mapping for `NVMeControllerProperties.NVMeControllerAttributes.ReportsNamespaceGranularity` is summarized in Table 40.

Table 40: NVMeControllerProperties.NVMeControllerAttributes.ReportsNamespaceGranularity mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeControllerProperties.NVMeControllerAttributes.ReportsNamespaceGranularity	**NVM Spec Property / Field:** Controller Attributes (CTRATT): Bit 7 (Namespace Granularity) **NVM Spec: Section:Figure** NVMe 1.4a: Section 5.15.2.2 (IdentifyController), Figure 249
Type	Boolean	**NVM Spec Property Type:** Single bit (bool) **Additional NVM Spec Identifying Information:** Bit 7 of Byte 99:96
Description	Indicates whether or not the controller supports reporting of Namespace Granularity.	
LongDescription	This property shall indicate whether or not the controller supports reporting of Namespace Granularity.	
M andatory	Recommended for NVM Drives and more complex devices with NVMe front ends, such as opaque arrays.	
Notes		

6.4.1.2.19 NVMeControllerProperties.NVMeControllerAttributes.SupportsSQAssociations

The mapping for `NVMeControllerProperties.NVMeControllerAttributes.SupportsSQAssociations` is summarized in Table 41.

Table 41: NVMeControllerProperties.NVMeControllerAttributes. SupportsSQAssociations mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeControllerProperties.NVMeControllerAttributes.SupportsSQAssociations	**NVM Spec Property / Field:** Controller Attributes (CTRATT): Bit 8 (SQ Associations) **NVM Spec: Section:Figure** NVMe 1.4a: Section 5.15.2.2 (IdentifyController), Figure 249
Type	Boolean	**NVM Spec Property Type:** Single bit (bool) **Additional NVM Spec Identifying Information:** Bit 8 of Byte 99:96
Description	Indicates whether or not the controller supports SQ Associations.	
LongDescription	This property shall indicate whether or not the controller supports SQ Associations.	
M andatory		
Notes		

6.4.1.2.20 NVMeControllerProperties.NVMeControllerAttributes.SupportsTrafficBasedKeepAlive

The mapping for `NVMeControllerProperties.NVMeControllerAttributes.TrafficBasedKeepAlive` is summarized in Table 42.

Table 42: NVMeControllerProperties.NVMeControllerAttributes. SupportsTrafficBasedKeepAlive mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeControllerProperties. NVMeControllerAttributes.SupportsTrafficBasedKeepAlive	**NVM Spec Property / Field:** Controller Attributes (CTRATT): Bit 6 (Traffic Based Keep Alive Support – TBKAS) **NVM Spec: Section:Figure** NVMe 1.4a: Section 5.15.2.2 (IdentifyController), Figure 249
Type	Boolean	**NVM Spec Property Type:** Single bit (bool) **Additional NVM Spec Identifying Information:** Bit 6 of Byte 99:96
Des cription	Indicates whether or not the controller supports restarting KeepAlive Timer if traffic is processed from an admin command or IO during KeepAlive Timeout interval.	
LongDes cription	This property shall indicate whether or not the controller supports restarting KeepAlive Timer if traffic is processed from an admin command or IO during KeepAlive Timeout interval.	
M andatory	Required for Ethernet-Attach Drives; required for more complex devices with NVMe front ends, such as opaque arrays.	

	Redfish/Swordfish	NVMe / NVMe-oF
Notes		

6.4.1.2.21 NVMeControllerProperties.NVMeControllerAttributes.SupportsExceedingPowerOfNonOpera

The mapping for `NVMeControllerProperties.NVMeControllerAttributes.SupportsExceedingPowerOfNonOp` is summarized in Table 43.

Table 43: NVMeControllerProperties.NVMeControllerAttributes. SupportsExceedingPowerOfNonOperationalState mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeControllerProperties.NVMeControllerAttributes.SupportsExceedingPowerOfNonOperationalState	**NVM Spec Property / Field:** Controller Attributes (CTRATT): Bit 1 (Non-Operational Power State Permissive Mode) **NVM Spec: Section:Figure** NVMe 1.4a: Section 5.15.2.2 (IdentifyController), Figure 249
Type	Boolean	**NVM Spec Property Type:** Single bit (bool) **Additional NVM Spec Identifying Information:** Bit 1 of Byte 99:96
Description	Indicates whether or not the controller supports exceeding Power of NonOperational State in order to execute controller initiated background operations in a non-operational power state.	
LongDescription	This property shall indicate whether or not the controller supports exceeding Power of NonOperational State in order to execute controller initiated background operations in a non-operational power state.	
M andatory		
Notes		

6.4.1.2.22 NVMeControllerProperties.NVMeControllerAttributes.Supports128BitHostId

The mapping for `NVMeControllerProperties.NVMeControllerAttributes.Supports128BitHostId` is summarized in Table 44.

Table 44: NVMeControllerProperties.NVMeControllerAttributes. Supports128BitHostId mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeControllerProperties.NVMeControllerAttributes.Supports128BitHostId	**NVM Spec Property / Field:** Controller Attributes (CTRATT): Bit 0 **NVM Spec: Section:Figure** NVMe 1.4a: Section 5.15.2.2 (IdentifyController), Figure 249
Type	Boolean	**NVM Spec Property Type:** Single bit (bool) **Additional NVM Spec Identifying Information:** Bit 0 of Byte 99:96
Des cription	Indicates whether or not the controller supports a 128-bit Host Identifier.	
LongDes cription	This property shall indicate whether or not the controller supports a 128-bit Host Identifier.	
M andatory		
Notes		

6.4.1.2.23 NVMeControllerProperties.MaxQueueSize The mapping for `NVMeControllerProperties.MaxQueueSize` is summarized in Table 45.

Table 45: NVMeControllerProperties.ANACharacteristics mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeContro llerProperties.MaxQueueSize	**NVM Spec Property / Field:** Maximum Queues Entries Supported (MQES) **NVM Spec: Section:Figure** NVMe 1.4a; Section 3.1.1 Controller Capabilities; Figure 69
Type	Int64	**NVM Spec Property Type: Additional NVM Spec Identifying Information:** ByteOffset: Bits 15:00
Des cription	Indicates the maximum individual queue size that an NVMe IO Controller supports.	
LongDes cription	This property shall contain the maximum individual queue entry size supported per queue. This is a zero-based value, where the minimum value is one, indicating two entries. For PCIe, this applies to both submission and completion queues. For NVMe-oF, this applies to only submission queues.	
M andatory	Implement for more complex devices with NVMe front ends, such as opaque arrays.	
Notes		

6.4.1.2.24 NVMeControllerProperties.MaxQueueSize The mapping for `NVMeControllerProperties.ANACharacteristics` is summarized in Table 46.

Table 46: NNVMeControllerProperties.ANACharacteristics mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeControllerPr operties.ANACharacteristics	
Type	Co llection(StorageController. v1_0_0.ANACharacteristics)	
Des cription	This property contains the combination of ANA type and volume information.	
LongDes cription	This property shall contain the combination of ANA type and volume information.	
M andatory		
Notes		

6.4.1.2.25 NVMeControllerProperties.NVMeSMARTCriticalWarnings.OverallSystemDegraded

The mapping for `NVMeControllerProperties.NVMeSMARTCriticalWarnings.OverallSystemDegraded` is summarized in Table 47.

Table 47: NVMeControllerProperties.NVMeSMARTCriticalWarnings. OverallSystemDegraded mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeControllerProperties.NVMeSMARTCriticalWarnings.OverallSystemDegraded	**NVM Spec Property / Field:** Critical Warning **NVM Spec: Section:Figure** NVMe 1.4a: Section 5.14.1.2, SMART / Health Information, Figure 196
Type	Boolean	**NVM Spec Property Type:** Single bit (bool) **Additional NVM Spec Identifying Information:** Bit 2 of Byte 00
Description	Indicates that the NVM subsystem reliability has been compromised.	
LongDescription	This property shall indicate that the NVM subsystem reliability has been compromised.	
M andatory	Required	
Notes		

6.4.1.2.26 NVMeControllerProperties.NVMeSMARTCriticalWarnings.PowerBackupFailed

The mapping for `NVMeControllerProperties.NVMeSMARTCriticalWarnings.PowerBackupFailed` is summarized in Table 48.

Table 48: NVMeControllerProperties.NVMeSMARTCriticalWarnings.PowerBackupFailed mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeController Properties.NVMeSMARTCritica lWarnings.PowerBackupFailed	**NVM Spec Property / Field:** Critical Warning **NVM Spec: Section:Figure** NVMe 1.4a: Section 5.14.1.2, SMART / Health Information, Figure 196
Type	Boolean	**NVM Spec Property Type:** Single bit (bool) **Additional NVM Spec Identifying Information:** Bit 4 of Byte 00
Des cription	Indicates that the volatile memory backup device has failed.	
LongDes cription	This property shall indicate that the volatile memory backup device has failed.	
M andatory	Recommended for NVM Drives. Required for more complex devices with NVMe front ends, such as opaque arrays.	
Notes		

6.4.1.2.27 SKU The mapping for SKU is summarized in Table 49.

Table 49: SKU mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	SKU	**NVM Spec Property / Field:** N/A **NVM Spec: Section:Figure** N/A
Type	Edm.String	N/A
Des cription	The SKU for this storage controller.	N/A
LongDes cription	This property shall contain the stock-keeping unit number for this storage storage controller.	N/A
M andatory	Do Not Implement	
Notes		

6.4.1.2.28 SpeedGbps The mapping for `SpeedGbps` is summarized in Table 50.

Table 50: SpeedGbps mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	SpeedGbps	**NVM Spec Property / Field:** N/A **NVM Spec: Section:Figure** N/A
Type	Decimal	N/A
Des cription	The maximum speed of the storage controller's device interface.	N/A
LongDes cription	This property shall represent the maximum supported speed of the storage bus interface, in Gbit/s. The specified interface connects the controller to the storage devices, not the controller to a host. For example, SAS bus not PCIe host bus.	N/A
M andatory	Do Not Implement	
Notes		

6.4.1.2.29 Status.Health The mapping for `Status.Health` is summarized in Table 51.

Table 51: Status.Health mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Status.Health	**NVM Spec Property / Field:** CSTS – Controller Status **NVM Spec: Section:Figure** NVMe 1.4a: Section 3.1.6, Figure 79 **NVM Spec Property / Field:** Critical Warning **NVM Spec: Section:Figure** NVMe 1.4a: Section 5.14.1.2, SMART / Health Information, Figure 196
Type	Resource.Health	**NVM Spec Property Type:**
Des cription	The health state of this resource in the absence of its dependent resources.	
LongDes cription	This property shall represent the health state of the resource without considering its dependent resources. The values shall conform to those defined in the Redfish Specification.	
M andatory		
Notes	Possible Values: OK / Warning / Critical	This comes from CSTS Controller Failure Status, and from the SMART / health information log critical warning field.

6.4.1.2.30 Status.State The mapping for `Status.State` is summarized in Table 52.

Table 52: Status.State mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Status.State	**NVM Spec Property / Field:** CSTS – Controller Status **NVM Spec: Section:Figure** NVMe 1.4a: Section 3.1.6, Figure 79
Type	Resource.State (enum)	**NVM Spec Property Type:**
Des cription	The known state of the resource, such as, enabled.	
LongDes cription	This property shall indicate whether and why this component is available. Enabled indicates the resource is available. Disabled indicates the resource has been intentionally made unavailable but it can be enabled. Offline indicates the resource is unavailable intentionally and requires action to make it available. InTest indicates that the component is undergoing testing. Starting indicates that the resource is becoming available. Absent indicates the resource is physically unavailable.	
M andatory		Mandatory

	Redfish/Swordfish	NVMe / NVMe-oF
Notes	Possible values: Enabled / Disabled / StandbyOffline / StandbySpare / InTest / Starting / ABsent / UnavaialableOffline / Deferring / Quiesced / Updating / Qualified	Ready (CSTS.RDY) maps to Enabled, Shutdown (CSTS.SHST) value will tell you if shutdown is in progress or complete (StandbyOffline), ProcessingPaused (CSTS.PP) maps to Deferring. If both Ready and Shutdown are indicated, then the system should indicate StandbyOffline. If both Ready and ProcessingPaused are indicated, then the system should indicate Deferring.

6.4.1.2.31 SupportedControllerProtocols The mapping for `SupportedControllerProtocols` is summarized in Table 53.

Table 53: SupportedControllerProtocols mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	S upportedControllerProtocols	N/A
Type	Co llection(Protocol.Protocol)	**NVM Spec Property Type:** N/A
Des cription	The supported set of protocols for communicating to this storage controller.	
LongDes cription	This property shall contain the supported set of protocols for communicating to this storage controller.	
M andatory	Required.	
Notes	This is an array of protocols supported by the StorageController. This can be set to values including, but not limited to, PCIe, RDMA, NVMe-oF, RoCE, RoCEv2, and InfiniBand.	

6.4.1.2.32 SupportedDeviceProtocols The mapping for `SupportedDeviceProtocols` is summarized in Table 54.

Table 54: SupportedDeviceProtocols mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	SupportedDeviceProtocols	N/A
Type	Co llection(Protocol.Protocol)	**NVM Spec Property Type:** N/A
Des cription	The protocols that the storage controller can use to communicate with attached devices.	
LongDes cription	This property shall contain the set of protocols this storage controller can use to communicate with attached devices.	
M andatory		
Notes	Do not implement.	

6.4.2 Discovery Controller

6.4.2.1 Mockup The following mockup shows a sample representation of the StorageController schema used to represent a Discovery Controller.

```
{
  "@Redfish.Copyright": "Copyright 2014-2020 SNIA. All rights reserved.",
  "@odata.id":
  ↪  "/redfish/v1/Storage/NVMe-oF-Subsystem/Controllers/NVMeIOController",
  "@odata.type": "#StorageController.v1_0_0.StorageController",
  "Id": "9",
  "Name": "NVMe Logical Discovery Controller",
  "Description": "Single NVMe Discovery Controller presented to host.",
  "Status": {
    "State": "Enabled",
    "Health": "OK"
  },
  "Id": "NVMeDiscoveryController",

  "SupportedControllerProtocols": [
    "TCP", "RDMA", "FC"
  ],
  "SupportedDeviceProtocols": [
    "NVMeOverFabrics"
  ],

  "NVMeControllerProperties": {
    "ControllerType": "Discovery",
    "NVMeVersion": "1.3",
    "NVMeControllerAttributes": {
      "SupportsTrafficBasedKeepAlive": false,
      "SupportsExceedingPowerOfNonOperationalState": false,
      "Supports128BitHostId": false
    }
  }
}
```

6.4.2.2 Property Mapping

6.4.2.2.1 Assembly The mapping for `Assembly` is summarized in Table 55.

Table 55: Assembly mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Assembly	**NVM Spec Property / Field:** N/A **NVM Spec: Section:Figure** N/A
Type	ComplexType	N/A
Des cription	The Assembly schema defines an assembly. Assembly information contains details about a device, such as part number, serial number, manufacturer, and production date. It also provides access to the original data for the assembly.	N/A
LongDes cription	This Resource shall represent an assembly for a Redfish implementation. Assembly information contains details about a device, such as part number, serial number, manufacturer, and production date. It also provides access to the original data for the assembly.	N/A
M andatory	Do Not Implement for NVMe drives, or devices with NVMe front ends, such as opaque arrays.	
Notes		

6.4.2.2.2 AssetTag The mapping for `AssetTag` is summarized in Table 56.

Table 56: Assembly mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	AssetTag	**NVM Spec Property / Field:** N/A **NVM Spec: Section:Figure** N/A
Type	Edm.String	N/A
Des cription	The user-assigned asset tag for this storage controller.	N/A
LongDes cription	This property shall track the storage controller for inventory purposes.	N/A
M andatory	Do Not Implement for NVMe Drives, or devices with NVMe front ends, such as opaque arrays.	
Notes		

6.4.2.2.3 CacheSummary The mapping for `CacheSummary` is summarized in Table 57.

Table 57: CacheSummary mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	CacheSummary	N/A
Type	ComplexType	N/A
Des cription	The cache memory of the storage controller in general detail.	N/A
LongDes cription	This property shall contain properties that describe the cache memory for this resource.	N/A
M andatory	Do Not Implement	
Notes		This property exists for hw cache reporting in other RF/SF use cases. Not used in NVMe controllers.

6.4.2.2.4 ControllerRates The mapping for `ControllerRates` is summarized in Table 58.

Table 58: ControllerRates mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	ControllerRates	N/A
Type	ComplexType	N/A
Des cription	This property describes the various controller rates used for processes such as volume rebuild or consistency checks.	N/A
LongDes cription	This object shall contain all the rate settings available on the controller.	N/A
M andatory	Do Not Implement	
Notes		

6.4.2.2.5 Description The mapping for `Description` is summarized in Table 59.

Table 59: Description mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Description	N/A
Type	String	
Des cription	The description of this resource.	
LongDes cription	This object represents the description of this resource. The resource values shall comply with the Redfish Specification-described requirements.	
M andatory	Optional	N/A
Notes	In Redfish, Description is a read-only field.	Return the common description: "An NVM Discovery Controller exposes capabilities that allow a host to retrieve information required to connect to one or more NVM Subsystems. Discovery controllers only support commands providing discovery capabilities; they do not provide IO or management access."

6.4.2.2.6 FirmwareVersion The mapping for `FirmwareVersion` is summarized in Table 60.

Table 60: FirmwareVersion mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	FirmwareVersion	**NVM Spec Property / Field:** IdentifyController: Firmare Revision (FR) **NVM Spec: Section:Figure** 249: 71:64
Type	String	
Des cription	The firmware version of this storage controller.	
LongDes cription	This property shall contain the firmware version as defined by the manufacturer for the associated storage controller.	
M andatory	Required.	Required
Notes		Return the currently active firmware revision information.

6.4.2.2.7 Identifiers The mapping for `Identifiers` is summarized in Table 61.

Table 61: Identifiers mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Identifiers	N/A
Type	Coll ection(Resource.Identifier)	N/A
Des cription	The Durable names for the storage controller.	N/A
LongDes cription	This property shall contain a list of all known durable names for the associated storage controller.	N/A

	Redfish/Swordfish	NVMe / NVMe-oF
M andatory	Recommend not implementing. There isn't a good mapping for these in the NVMe spec to a property that has an appropriate / mapping to a durable name format.	No
Notes		N/A

6.4.2.2.8 Identifiers.DurableName The mapping for `Identifiers.DurableName` is summarized in Table 62.

Table 62: Identifiers.DurableName mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Identifiers.DurableName	N/A
Type	Variable - see notes	N/A
Des cription	The Durable names for the storage controller.	N/A
LongDes cription	T	N/A
M andatory	Recommend not implementing.	No
Notes	Recommend not implementing. There isn't a good mapping for these in the NVMe spec to a property that has an appropriate / mapping to a durable name format.	N/A

6.4.2.2.9 Identifiers.DurableNameFormat The mapping for `Identifiers.DurableNameFormat` is summarized in Table 63.

Table 63: Identifiers.DurableNameFormat mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Id entifiers.DurableNameFormat	N/A
Type	enum (DurableNameFormat)	N/A
Des cription	The Durable names for the storage controller.	N/A
LongDes cription	This property shall contain a list of the types for all known durable names for the associated storage controller. The type determines the length of the corresponding Namespace ID	N/A
M andatory	Recommend not implementing.	No
Notes	Recommend not implementing. There isn't a good mapping for these in the NVMe spec to a property that has an appropriate / mapping to a durable name format.	N/A

6.4.2.2.10 Links.AttachedVolumes The mapping for `Links.AttachedVolumes` is summarized in Table 64.

Table 64: Links.AttachedVolumes mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Links.AttachedVolumes	N/A
Type	Collection(Volume.Volume)	N/A
Des cription	An array of links to volumes that are attached to this controller instance.	

	Redfish/Swordfish	NVMe / NVMe-oF
LongDescription	This property shall contain a link to the Resources of type Volume that are attached to this instance of storage controller.	
M andatory	Do not implement.	N/A
Notes		

6.4.2.2.11 Links.Endpoints The mapping for `Links.Endpoints` is summarized in Table 65.

Table 65: Links.Endpoints mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Links.Endpoints	N/A
Type	Co llection(Endpoint.Endpoint)	
Des cription	An array of links to the endpoints that connect to this controller.	
LongDes cription	This property shall contain an array of links to the Resources of type Endpoint associated with this controller.	
M andatory	Yes	
Notes		

6.4.2.2.12 Links.Connections The mapping for `Links.Connections` is summarized in Table 66.

Table 66: Links.Connections mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Links.Connections	N/A
Type	Collec tion(Connection.Connection)	
Des cription	An array of links to volumes that are attached to this controller instance.	
LongDes cription	This property shall contain a link to the Resources of type Volume that are attached to this instance of storage controller.	
M andatory		
Notes	This contains the information used to represented the allowed hosts.	This property contains pointers to the Connections objects. The information about allowed hosts is mapped to the Connections objects for NVMe-oF configurations.

6.4.2.2.13 Links.NetworkDeviceFunctions The mapping for `Links.NetworkDeviceFunctions` is summarized in Table 67.

Table 67: Links.NetworkDeviceFunctions mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	L inks.NetworkDeviceFunctions	N/A
Type	C ollection(NetworkDeviceFunc tion.NetworkDeviceFunction)	

	Redfish/Swordfish	NVMe / NVMe-oF
Des cription	The network device functions that provide connectivity to this controller.	
LongDes cription	This property shall contain an array of links to resources of type NetworkDeviceFunction that represent the devices that provide connectivity to this controller.	
M andatory	Recommended to implement for more complex devices with NVMe front ends, such as opaque arrays.	
Notes	For NVMe-oF configurations.	

6.4.2.2.14 Location The mapping for `Location` is summarized in Table 68.

Table 68: Location mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Location	**NVM Spec Property / Field:** N/A **NVM Spec: Section:Figure** N/A
Type	Co llection(Resource.Location)	
Des cription	The location of the storage controller.	
LongDes cription	This property shall contain location information of the associated storage controller.	N/A
M andatory	Do Not Implement for NVM Drives, or more complex devices with NVMe front ends, such as opaque arrays.	

	Redfish/Swordfish	NVMe / NVMe-oF
Notes		

6.4.2.2.15 Manufacturer The mapping for `Manufacturer` is summarized in Table 69.

Table 69: Manufacturer mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Manufacturer	NVM Spec Property / Field:** IdentifyController / PCI Vendor ID (VID) **NVM Spec: Section: Figure** 249 byte 00:01
Type	String	
Des cription	The manufacturer of this storage controller.	The company vendor identifier
LongDes cription	This property shall contain the name of the organization responsible for producing the storage controller. This organization might be the entity from whom the storage controller is purchased, but this is not necessarily true.	The company vendor identifier that is assigned by the PCI SIG. This is the same value as reported in the ID register
M andatory	Optional	
Notes		End clients expect to see the name of the company (e.g,; Contoso, BestVendor). While the value may be filled from the IdentifyController PCI Vendor ID or SubsystemID field, it would be preferable to have this filled with the actual string value of the company name.

6.4.2.2.16 Model The mapping for `Model` is summarized in Table 70.

Table 70: Model mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Model	
Type	String	N/A
Des cription	The model number for the storage controller.	
LongDes cription	This property shall contain the name by which the manufacturer generally refers to the storage controller.	N/A
M andatory	Optional	
Notes		

6.4.2.2.17 Name The mapping for `Name` is summarized in Table 71.

Table 71: Name mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Name	**NVM Spec Property / Field:** Controller ID (CNTLID) **NVM Spec: Section:Figure** NVMe 1.4a: Section 5.15.2.2 (IdentifyController), Figure 249
Type	String	**NVM Spec Property Type:** 16-bit hex value **Additional NVM Spec Identifying Information:** ByteOffset: 79:78, IdentifyController data structure
Des cription	The name of the resource or array member.	
LongDes cription	This object represents the name of this resource or array member. The resource values shall comply with the Redfish Specification-described requirements. This string value shall be of the 'Name' reserved word format.	
M andatory	Required	Mandatory
Notes	In Redfish, Name is a read-only field.	Map the CNTLID field to a string with the format: "0xABCD"

6.4.2.2.18 NVMeControllerProperties.ControllerType The mapping for `NVMeControllerProperties.ControllerType` is summarized in Table 72.

Table 72: NVMeControllerProperties.ControllerType mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeControll erProperties.ControllerType	**NVM Spec Property / Field:** Controller Type (CNTRL_TYPE) **NVM Spec: Section:Figure** Section 5.15.2.2 (IdentifyController), Figure 249 Byte 111
Type	StorageController .v1_0_0.NVMeControllerType	Hex value
Des cription	This property specifies the type of NVMe Controller.	Controller type.
LongDes cription	This property shall specify the type of NVMe Controller.	This field specifies the controller type. A value of 0h indicates that the controller type is not reported.
M andatory	Required property when Discovery controller is implemented.	Required
Notes	This property must be used to specify the type of NVMe Controller. For a discovery controller, set to Discovery.	For Discovery Controller - value in Identify Controller is ‘02h’. Return “Discovery”

6.4.2.2.19 NVMeControllerProperties.NVMeVersion The mapping for `NVMeControllerProperties.NVMeVersion` is summarized in Table 73.

Table 73: NVMeControllerProperties.NVMeVersion mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeContr ollerProperties.NVMeVersion	**NVM Spec Property / Field:** Version (VER) **NVM Spec: Section:Figure** NVMe 1.4a: Section 5.15.2.2 (IdentifyController), Figure 249
Type	String	**NVM Spec Property Type:** Maps from register 3.1.2. **Additional NVM Spec Identifying Information:** ByteOffset: 83:90
Des cription	The version of the NVMe Base Specification supported.	
LongDes cription	This property shall specify the type of NVMe Controller.	
M andatory		Mandatory
Notes		Implementations compliant to revision 1.2 or later of this specification shall report a non-zero value in this field. Map from corresponding value in register 3.1.2 to string as “1.0”, “1.1”, “1.2”, “1.2.1”, “1.3.0”, “1.4.0”, etc.

6.4.2.2.20 NVMeControllerProperties.NVMeControllerAttributes.SupportsTrafficBasedKeepAlive

The mapping for `NVMeControllerProperties.NVMeControllerAttributes.TrafficBasedKeepAlive` is summarized in Table 74.

Table 74: NVMeControllerProperties.NVMeControllerAttributes. SupportsTrafficBasedKeepAlive mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeControllerProperties. NVMeControllerAttributes.SupportsTrafficBasedKeepAlive	**NVM Spec Property / Field:** Controller Attributes (CTRATT): Bit 6 (Traffic Based Keep Alive Support – TBKAS) **NVM Spec: Section:Figure** NVMe 1.4a: Section 5.15.2.2 (IdentifyController), Figure 249
Type	Boolean	**NVM Spec Property Type:** Single bit (bool) **Additional NVM Spec Identifying Information:** Bit 6 of Byte 99:96
Des cription	Indicates whether or not the controller supports restarting KeepAlive Timer if traffic is processed from an admin command or IO during KeepAlive Timeout interval.	
LongDes cription	This property shall indicate whether or not the controller supports restarting KeepAlive Timer if traffic is processed from an admin command or IO during KeepAlive Timeout interval.	
M andatory	Required for Ethernet-Attach Drives; required for more complex devices with NVMe front ends, such as opaque arrays.	

	Redfish/Swordfish	NVMe / NVMe-oF
Notes		

6.4.2.2.21 NVMeControllerProperties.NVMeControllerAttributes.SupportsExceedingPowerOfNonOpera

The mapping for `NVMeControllerProperties.NVMeControllerAttributes.SupportsExceedingPowerOfNonOp` is summarized in Table 75.

Table 75: NVMeControllerProperties.NVMeControllerAttributes. SupportsExceeding-PowerOfNonOperationalState mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeControllerProperties.NVMeControllerAttributes.SupportsExceedingPowerOfNonOperationalState	**NVM Spec Property / Field:** Controller Attributes (CTRATT): Bit 1 (Non-Operational Power State Permissive Mode) **NVM Spec: Section:Figure** NVMe 1.4a: Section 5.15.2.2 (IdentifyController), Figure 249
Type	Boolean	**NVM Spec Property Type:** Single bit (bool) **Additional NVM Spec Identifying Information:** Bit 1 of Byte 99:96
Description	Indicates whether or not the controller supports exceeding Power of NonOperational State in order to execute controller initiated background operations in a non-operational power state.	
LongDescription	This property shall indicate whether or not the controller supports exceeding Power of NonOperational State in order to execute controller initiated background operations in a non-operational power state.	
M andatory		
Notes		

6.4.2.2.22 NVMeControllerProperties.NVMeControllerAttributes.Supports128BitHostId

The mapping for `NVMeControllerProperties.NVMeControllerAttributes.Supports128BitHostId` is summarized in Table 76.

Table 76:
NVMeControllerProperties.NVMeControllerAttributes. Supports128BitHostId mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeControllerProperties.NVMeControllerAttributes.Supports128BitHostId	**NVM Spec Property / Field:** Controller Attributes (CTRATT): Bit 0 **NVM Spec: Section:Figure** NVMe 1.4a: Section 5.15.2.2 (IdentifyController), Figure 249
Type	Boolean	**NVM Spec Property Type:** Single bit (bool) **Additional NVM Spec Identifying Information:** Bit 0 of Byte 99:96
Description	Indicates whether or not the controller supports a 128-bit Host Identifier.	
LongDescription	This property shall indicate whether or not the controller supports a 128-bit Host Identifier.	
M andatory		
Notes		

6.4.2.2.23 NVMeControllerProperties.MaxQueueSize The mapping for `NVMeControllerProperties.MaxQueueSize` is summarized in Table 77.

Table 77: NVMeControllerProperties.ANACharacteristics mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeContro llerProperties.MaxQueueSize	**NVM Spec Property / Field:** Maximum Queues Entries Supported (MQES) **NVM Spec: Section:Figure** NVMe 1.4a; Section 3.1.1 Controller Capabilities; Figure 69
Type	Int64	**NVM Spec Property Type: Additional NVM Spec Identifying Information:** ByteOffset: Bits 15:00
Des cription	Indicates the maximum individual queue size that an NVMe IO Controller supports.	
LongDes cription	This property shall contain the maximum individual queue entry size supported per queue. This is a zero-based value, where the minimum value is one, indicating two entries. For PCIe, this applies to both submission and completion queues. For NVMe-oF, this applies to only submission queues.	
M andatory	Implement for more complex devices with NVMe front ends, such as opaque arrays.	
Notes		

6.4.2.2.24 NVMeControllerProperties.NVMeSMARTCriticalWarnings.OverallSubsystemDegraded

The mapping for `NVMeControllerProperties.NVMeSMARTCriticalWarnings.OverallSubsystemDegraded` is summarized in Table 78.

Table 78: NVMeControllerProperties.NVMeSMARTCriticalWarnings. OverallSubsystemDegraded mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeControllerProperties.NVMeSMARTCriticalWarnings.OverallSubsystemDegraded	**NVM Spec Property / Field:** Critical Warning **NVM Spec: Section:Figure** NVMe 1.4a: Section 5.14.1.2, SMART / Health Information, Figure 196
Type	Boolean	**NVM Spec Property Type:** Single bit (bool) **Additional NVM Spec Identifying Information:** Bit 2 of Byte 00
Description	Indicates that the NVM subsystem reliability has been compromised.	
LongDescription	This property shall indicate that the NVM subsystem reliability has been compromised.	
M andatory	Required	
Notes		

6.4.2.2.25 NVMeControllerProperties.NVMeSMARTCriticalWarnings.SpareCapacityWornOut

The mapping for `NVMeControllerProperties.NVMeSMARTCriticalWarnings.SpareCapacityWornOut` is summarized in Table 79.

Table 79:
NVMeControllerProperties.NVMeSMARTCriticalWarnings. SpareCapacityWornOut mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeControllerProperties.NVMeSMARTCriticalWarnings.SpareCapacityWornOut	**NVM Spec Property / Field:** Critical Warning **NVM Spec: Section:Figure** NVMe 1.4a: Section 5.14.1.2, SMART / Health Information, Figure 196
Type	Boolean	**NVM Spec Property Type:** Single bit (bool) **Additional NVM Spec Identifying Information:** Bit 0 of Byte 00
Description	Indicates that the available spare capacity has fallen below the threshold.	
LongDescription	This property shall indicate that the available spare capacity has fallen below the threshold.	
M andatory	Required	
Notes		

6.4.2.2.26 NVMeControllerProperties.NVMeSMARTCriticalWarnings.PowerBackupFailed

The mapping for `NVMeControllerProperties.NVMeSMARTCriticalWarnings.PowerBackupFailed` is summarized in Table 80.

Table 80: NVMeControllerProperties.NVMeSMARTCriticalWarnings.PowerBackupFailed mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeController Properties.NVMeSMARTCritica lWarnings.PowerBackupFailed	**NVM Spec Property / Field:** Critical Warning **NVM Spec: Section:Figure** NVMe 1.4a: Section 5.14.1.2, SMART / Health Information, Figure 196
Type	Boolean	**NVM Spec Property Type:** Single bit (bool) **Additional NVM Spec Identifying Information:** Bit 4 of Byte 00
Des cription	Indicates that the volatile memory backup device has failed.	
LongDes cription	This property shall indicate that the volatile memory backup device has failed.	
M andatory	Recommended for NVM Drives. Required for more complex devices with NVMe front ends, such as opaque arrays.	
Notes		

6.4.2.2.27 Status.Health The mapping for `Status.Health` is summarized in Table 81.

Table 81: Status.Health mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Status.Health	**NVM Spec Property / Field:** CSTS – Controller Status **NVM Spec: Section:Figure** NVMe 1.4a: Section 3.1.6, Figure 79 **NVM Spec Property / Field:** Critical Warning **NVM Spec: Section:Figure** NVMe 1.4a: Section 5.14.1.2, SMART / Health Information, Figure 196
Type	Resource.Health	**NVM Spec Property Type:**
Des cription	The health state of this resource in the absence of its dependent resources.	
LongDes cription	This property shall represent the health state of the resource without considering its dependent resources. The values shall conform to those defined in the Redfish Specification.	
M andatory		
Notes	Possible Values: OK / Warning / Critical	This comes from CSTS Controller Failure Status, and from the SMART / health information log critical warning field.

6.4.2.2.28 Status.State The mapping for `Status.State` is summarized in Table 82.

Table 82: Status.State mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Status.State	**NVM Spec Property / Field:** CSTS – Controller Status **NVM Spec: Section:Figure** NVMe 1.4a: Section 3.1.6, Figure 79
Type	Resource.State (enum)	**NVM Spec Property Type:**
Des cription	The known state of the resource, such as, enabled.	
LongDes cription	This property shall indicate whether and why this component is available. Enabled indicates the resource is available. Disabled indicates the resource has been intentionally made unavailable but it can be enabled. Offline indicates the resource is unavailable intentionally and requires action to make it available. InTest indicates that the component is undergoing testing. Starting indicates that the resource is becoming available. Absent indicates the resource is physically unavailable.	
M andatory		Mandatory

	Redfish/Swordfish	NVMe / NVMe-oF
Notes	Possible values: Enabled / Disabled / StandbyOffline / StandbySpare / InTest / Starting / ABsent / UnavaialableOffline / Deferring / Quiesced / Updating / Qualified	Ready (CSTS.RDY) maps to Enabled, Shutdown (CSTS.SHST) value will tell you if shutdown is in progress or complete (StandbyOffline), ProcessingPaused (CSTS.PP) maps to Deferring. If both Ready and Shutdown are indicated, then the system should indicate StandbyOffline. If both Ready and ProcessingPaused are indicated, then the system should indicate Deferring.

6.4.2.2.29 SupportedControllerProtocols The mapping for `SupportedControllerProtocols` is summarized in Table 83.

Table 83: SupportedControllerProtocols mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	S upportedControllerProtocols	N/A
Type	Co llection(Protocol.Protocol)	**NVM Spec Property Type:** N/A
Des cription	The supported set of protocols for communicating to this storage controller.	
LongDes cription	This property shall contain the supported set of protocols for communicating to this storage controller.	
M andatory	Required.	
Notes	This is an array of protocols supported by the StorageController. This can be set to values including, but not limited to, PCIe, RDMA, NVMe-oF, RoCE, RoCEv2, and InfiniBand.	

6.4.2.2.30 SupportedDeviceProtocols The mapping for `SupportedDeviceProtocols` is summarized in Table 84.

Table 84: SupportedDeviceProtocols mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	SupportedDeviceProtocols	N/A
Type	Co llection(Protocol.Protocol)	**NVM Spec Property Type:** N/A
Des cription	The protocols that the storage controller can use to communicate with attached devices.	
LongDes cription	This property shall contain the set of protocols this storage controller can use to communicate with attached devices.	
M andatory		
Notes	Do not implement.	

6.4.3 IO Controller

6.4.3.1 Mockup The following mockup shows a sample representation of the StorageController schema used to represent an IO Controller.

```
{
  "@Redfish.Copyright": "Copyright 2014-2020 SNIA. All rights reserved.",
  "@odata.id":
  ↪  "/redfish/v1/Storage/NVMe-oF-Subsystem/Controllers/NVMeIOController",
  "@odata.type": "#StorageController.v1_0_0.StorageController",
  "Id": "9",
  "Name": "NVMe Logical I/O Controller",
  "Description": "Single NVMe I/O Controller presented to host.",
  "Status": {
    "State": "Enabled",
    "Health": "OK"
  },
  "Id": "NVMeIOController",

  "SupportedControllerProtocols": [
    "TCP", "RDMA", "FC"
  ],
  "SupportedDeviceProtocols": [
    "NVMeOverFabrics"
  ],

  "NVMeControllerProperties": {
    "NVMeVersion": "1.3",
    "NVMeControllerAttributes": {
      "ReportsUUIDList": false,
      "SupportsSQAssociations": false,
      "ReportsNamespaceGranularity": false,
      "SupportsReservations": true,
      "SupportsTrafficBasedKeepAlive": false,
      "SupportsPredictableLatencyMode": false,
      "SupportsEnduranceGroups": false,
      "SupportsReadRecoveryLevels": false,
      "SupportsNVMSets": false,
      "SupportsExceedingPowerOfNonOperationalState": false,
      "Supports128BitHostId": false,
    },

    "MaxQueueSize": 1,
    "ANACharacteristics": [{
      "AccessState": "Optimized",
      "Volume": {
        "@odata.id":
        ↪  "/redfish/v1/Systems/Sys-1/Storage/NVMeSSD-EG/Volumes/Namespace1"
      }
    }]
  },
  "Links": {
    "AttachedVolumes": [{
      "@odata.id":
      ↪  "/redfish/v1/Systems/Sys-1/Storage/NVMeSSD-EG/Volumes/Namespace1"
    }],
    "Endpoints": [{
        "@odata.id": "/redfish/v1/Fabrics/NVMe-oF/Endpoints/NVMeEndpoint"
      },
      {
        "@odata.id": "/redfish/v1/Fabrics/NVMe-oF/Endpoints/Host"
      }
    ],
    "Connections": [{
      "@odata.id": "/redfish/v1/Fabrics/NVMe-oF/Connections/Host1"
    }]
  }
}
```

6.4.3.2 Property Mapping

6.4.3.2.1 Assembly The mapping for `Assembly` is summarized in Table 85.

Table 85: Assembly mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Assembly	**NVM Spec Property / Field:** N/A **NVM Spec: Section:Figure** N/A
Type	ComplexType	N/A
Des cription	The Assembly schema defines an assembly. Assembly information contains details about a device, such as part number, serial number, manufacturer, and production date. It also provides access to the original data for the assembly.	N/A
LongDes cription	This Resource shall represent an assembly for a Redfish implementation. Assembly information contains details about a device, such as part number, serial number, manufacturer, and production date. It also provides access to the original data for the assembly.	N/A
M andatory	Do Not Implement for NVMe drives, or devices with NVMe front ends, such as opaque arrays.	N/A
Notes		

6.4.3.2.2 AssetTag

The mapping for `AssetTag` is summarized in Table 86.

Table 86: Assembly mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	AssetTag	**NVM Spec Property / Field:** N/A **NVM Spec: Section:Figure** N/A
Type	Edm.String	N/A
Des cription	The user-assigned asset tag for this storage controller.	N/A
LongDes cription	This property shall track the storage controller for inventory purposes.	N/A
M andatory	Do Not Implement for NVMe Drives, or devices with NVMe front ends, such as opaque arrays.	N/A
Notes		

6.4.3.2.3 CacheSummary The mapping for `CacheSummary` is summarized in Table 87.

Table 87: CacheSummary mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	CacheSummary	N/A
Type	ComplexType	N/A
Des cription	The cache memory of the storage controller in general detail.	N/A
LongDes cription	This property shall contain properties that describe the cache memory for this resource.	N/A
M andatory	Do Not Implement	N/A
Notes		This property exists for hw cache reporting in other RF/SF use cases. Not used in NVMe controllers.

6.4.3.2.4 ControllerRates The mapping for `ControllerRates` is summarized in Table 88.

Table 88: ControllerRates mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	ControllerRates	N/A
Type	ComplexType	N/A
Des cription	This property describes the various controller rates used for processes such as volume rebuild or consistency checks.	N/A
LongDes cription	This object shall contain all the rate settings available on the controller.	N/A
M andatory	Do Not Implement	N/A
Notes		

6.4.3.2.5 Description

The mapping for `Description` is summarized in Table 89.

Table 89: Description mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Description	N/A
Type	String	N/A
Des cription	The description of this resource.	N/A
LongDes cription	This object represents the description of this resource. The resource values shall comply with the Redfish Specification-described requirements.	N/A
M andatory	Required	N/A
Notes	In Redfish, Description is a read-only field.	Return the common description: "An NVM IO controller is a general-purpose controller that provides access to logical block data and metadata stored on an NVM subsystem's non-volatile storage medium. IO Controllers may also support management capabilities."

6.4.3.2.6 FirmwareVersion

The mapping for `FirmwareVersion` is summarized in Table 90.

Table 90: FirmwareVersion mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	FirmwareVersion	**NVM Spec Property / Field:** IdentifyController: Firmare Revision (FR) **NVM Spec: Section:Figure** 249: 71:64
Type	String	
Des cription	The firmware version of this storage controller.	
LongDes cription	This property shall contain the firmware version as defined by the manufacturer for the associated storage controller.	
M andatory	Required.	Required
Notes		Return the currently active firmware revision information.

6.4.3.2.7 Identifiers The mapping for `Identifiers` is summarized in Table 91.

Table 91: Identifiers mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Identifiers	N/A
		N/A
Des cription	The Durable names for the storage controller.	N/A
LongDes cription	This property shall contain a list of all known durable names for the associated storage controller.	N/A

	Redfish/Swordfish	NVMe / NVMe-oF
M andatory	Recommend not implementing. There isn't a good mapping for these in the NVMe spec to a property that has an appropriate / mapping to a durable name format.	N/A
Notes		N/A

6.4.3.2.8 Identifiers.DurableName The mapping for `Identifiers.DurableName` is summarized in Table 92.

Table 92: Identifiers.DurableName mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Identifiers.DurableName	N/A
Type	Variable - see notes	N/A
Des cription	The Durable names for the storage controller.	N/A
LongDes cription	T	N/A
M andatory	Yes	N/A
Notes	Recommend not implementing. There isn't a good mapping for these in the NVMe spec to a property that has an appropriate / mapping to a durable name format.	N/A

6.4.3.2.9 Identifiers.DurableNameFormat The mapping for `Identifiers.DurableNameFormat` is summarized in Table 93.

Table 93: Identifiers.DurableNameFormat mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Id entifiers.DurableNameFormat	N/A
Type	enum (DurableNameFormat)	N/A
Des cription	The Durable names for the storage controller.	N/A
LongDes cription	This property shall contain a list of the types for all known durable names for the associated storage controller. The type determines the length of the corresponding Namespace ID	N/A
M andatory	Recommend not implementing.	N/A
Notes	Recommend not implementing. There isn’t a good mapping for these in the NVMe spec to a property that has an appropriate / mapping to a durable name format.	N/A

6.4.3.2.10 Links.AttachedVolumes The mapping for `Links.AttachedVolumes` is summarized in Table 94.

Table 94: Links.AttachedVolumes mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Links.AttachedVolumes	N/A
Type	Collection(Volume.Volume)	N/A
Des cription	An array of links to volumes that are attached to this controller instance.	
LongDes cription	This property shall contain a link to the Resources of type Volume that are attached to this instance of storage controller.	
M andatory	Required.	Required
Notes	This contains a pointer to the set of namespaces attached to this IO Controller.	The Identify command (refer to NVMe Base spec section 5.15.1) may be used to return a data buffer that describes information about the NVM subsystem, the controller or the namespace(s) and thus may be used to determine the active NSIDs for a controller and the allocated NSIDs in the NVM subsystem

6.4.3.2.11 Links.Endpoints The mapping for `Links.Endpoints` is summarized in Table 95.

Table 95: Links.Endpoints mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Links.Endpoints	N/A
Type	Co llection(Endpoint.Endpoint)	N/A
Des cription	An array of links to the endpoints that connect to this controller.	N/A
LongDes cription	This property shall contain an array of links to the Resources of type Endpoint associated with this controller.	N/A
M andatory		N/A
Notes	For NVMe-oF configurations.	

6.4.3.2.12 Links.Connections The mapping for `Links.Connections` is summarized in Table 96.

Table 96: Links.Connections mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Links.Connections	N/A
Type	Collec tion(Connection.Connection)	
Des cription	An array of links to volumes that are attached to this controller instance.	
LongDes cription	This property shall contain a link to the Resources of type Volume that are attached to this instance of storage controller.	
M andatory	N/A for NVMe Drives.	
Notes	This contains the information used to represented the allowed hosts.	This property contains pointers to the Connections objects. The information about allowed hosts is mapped to the Connections objects for NVMe-oF configurations.

6.4.3.2.13 Links.NetworkDeviceFunctions The mapping for `Links.NetworkDeviceFunctions` is summarized in Table 97.

Table 97: Links.NetworkDeviceFunctions mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	L inks.NetworkDeviceFunctions	N/A
Type	C ollection(NetworkDeviceFunc tion.NetworkDeviceFunction)	N/A

	Redfish/Swordfish	NVMe / NVMe-oF
Des cription	The network device functions that provide connectivity to this controller.	N/A
LongDes cription	This property shall contain an array of links to resources of type NetworkDeviceFunction that represent the devices that provide connectivity to this controller.	N/A
M andatory	Recommended to implement for more complex devices with NVMe front ends, such as opaque arrays.	N/A
Notes	For NVMe-oF configurations.	

6.4.3.2.14 Location The mapping for `Location` is summarized in Table 98.

Table 98: Location mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Location	**NVM Spec Property / Field:** N/A **NVM Spec: Section:Figure** N/A
Type	Co llection(Resource.Location)	N/A
Des cription	The location of the storage controller.	N/A
LongDes cription	This property shall contain location information of the associated storage controller.	N/A
M andatory	Do Not Implement for NVM Drives, or more complex devices with NVMe front ends, such as opaque arrays.	N/A

	Redfish/Swordfish	NVMe / NVMe-oF
Notes		

6.4.3.2.15 Manufacturer The mapping for `Manufacturer` is summarized in Table 99.

Table 99: Manufacturer mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Manufacturer	NVM Spec Property / Field:** IdentifyController / PCI Vendor ID (VID) **NVM Spec: Section: Figure** 249 byte 00:01
Type	String	
Des cription	The manufacturer of this storage controller.	The company vendor identifier
LongDes cription	This property shall contain the name of the organization responsible for producing the storage controller. This organization might be the entity from whom the storage controller is purchased, but this is not necessarily true.	The company vendor identifier that is assigned by the PCI SIG. This is the same value as reported in the ID register
M andatory	Required	Required

	Redfish/Swordfish	NVMe / NVMe-oF
Notes		End clients expect to see the name of the company (e.g,; Contoso, BestVendor). While the value may be filled from the IdentifyController PCI Vendor ID (figure 249) or Subsystem Vendor ID field (figure 24), it would be preferable to have this filled with the actual string value of the company name.

6.4.3.2.16 Model The mapping for `Model` is summarized in Table 100.

Table 100: Model mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Model	**NVM Spec Property / Field:** IdentifyController / Model Number (MN) **NVM Spec: Section: Figure** 249 byte 24:63
Type	String	
Des cription	The model number for the storage controller.	
LongDes cription	This property shall contain the name by which the manufacturer generally refers to the storage controller.	Model Number (MN): Contains the model number for the NVM subsystem that is assigned by the vendor as an ASCII string. Refer to section 7.10 for unique identifier requirements. Refer to section 1.5 for ASCII string requirements
M andatory	Required	Required
Notes		

6.4.3.2.17 Name The mapping for `Name` is summarized in Table 101.

Table 101: Name mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Name	**NVM Spec Property / Field:** Controller ID (CNTLID) **NVM Spec: Section:Figure** NVMe 1.4a: Section 5.15.2.2 (IdentifyController), Figure 249
Type	String	**NVM Spec Property Type:** 16-bit hex value **Additional NVM Spec Identifying Information:** ByteOffset: 79:78, IdentifyController data structure
Des cription	The name of the resource or array member.	
LongDes cription	This object represents the name of this resource or array member. The resource values shall comply with the Redfish Specification-described requirements. This string value shall be of the 'Name' reserved word format.	
M andatory	Required	Required
Notes	In Redfish, Name is a read-only field.	Map the CNTLID field to a string with the format: "0xABCD"

6.4.3.2.18 NVMeControllerProperties.ControllerType The mapping for `NVMeControllerProperties.ControllerType` is summarized in Table 102.

Table 102: NVMeControllerProperties.ControllerType mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeControll erProperties.ControllerType	**NVM Spec Property / Field:** Controller Type (CNTRL_TYPE) **NVM Spec: Section:Figure** Section 5.15.2.2 (IdentifyController), Figure 249 Byte 111
Type	StorageController .v1_0_0.NVMeControllerType	Hex Value
Des cription	This property specifies the type of NVMe Controller.	Controller Type
LongDes cription	This property shall specify the type of NVMe Controller.	This field specifies the controller type. A value of 0h indicates that the controller type is not reported.
M andatory	Required	Required
Notes	This property must be used to specify the type of NVMe Controller. For an IO controller, set to IO.	For IO Controller - value in Identify Controller is ‘01h’. Return “IO”

6.4.3.2.19 NVMeControllerProperties.NVMeVersion The mapping for `NVMeControllerProperties.NVMeVersion` is summarized in Table 103.

Table 103: NVMeControllerProperties.NVMeVersion mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeContr ollerProperties.NVMeVersion	**NVM Spec Property / Field:** Version (VER) **NVM Spec: Section:Figure** NVMe 1.4a: Section 5.15.2.2 (IdentifyController), Figure 249
Type	String	**NVM Spec Property Type:** Maps from register 3.1.2. **Additional NVM Spec Identifying Information:** ByteOffset: 83:90
Des cription	The version of the NVMe Base Specification supported.	
LongDes cription	This property shall specify the type of NVMe Controller.	
M andatory	Required when NVMe version is 1.2 or higher.	Required when NVMe version is 1.2 or higher.
Notes		Implementations compliant to revision 1.2 or later of this specification shall report a non-zero value in this field. Map from corresponding value in register 3.1.2 to string as “1.0”, “1.1”, “1.2”, “1.2.1”, “1.3.0”, “1.4.0”, etc.

6.4.3.2.20 NVMeControllerProperties.NVMeControllerAttributes.ReportsUUIDList

The mapping for `NVMeControllerProperties.NVMeControllerAttributes.ReportsUUIDList` is summarized in Table 104.

Table 104: NVMeControllerProperties.NVMeControllerAttributes.ReportsUUIDList mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeControllerProperties.NVMeControllerAttributes.ReportsUUIDList	**NVM Spec Property / Field:** Controller Attributes (CTRATT): UUID List (Bit 9) **NVM Spec: Section:Figure** NVMe 1.4a: Section 5.15.2.2 (IdentifyController), Figure 249
Type	Boolean	**NVM Spec Property Type:** Single bit (bool) **Additional NVM Spec Identifying Information:** Bit 9 of Byte 99:96
Description	Indicates whether or not the controller supports reporting of a UUID list.	
LongDescription	This property shall indicate whether or not the controller supports reporting of a UUID list.	
M andatory	Optional	Optional
Notes		

6.4.3.2.21 NVMeControllerProperties.NVMeControllerAttributes.SupportsSQAssociations

The mapping for `NVMeControllerProperties.NVMeControllerAttributes.SupportsSQAssociations` is summarized in Table 105.

Table 105: NVMeControllerProperties.NVMeControllerAttributes. SupportsSQAssociations mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeControllerProperties.NVMeControllerAttributes.SupportsSQAssociations	**NVM Spec Property / Field:** Controller Attributes (CTRATT): Bit 8 (SQ Associations) **NVM Spec: Section:Figure** NVMe 1.4a: Section 5.15.2.2 (IdentifyController), Figure 249
Type	Boolean	**NVM Spec Property Type:** Single bit (bool) **Additional NVM Spec Identifying Information:** Bit 8 of Byte 99:96
Description	Indicates whether or not the controller supports SQ Associations.	
LongDescription	This property shall indicate whether or not the controller supports SQ Associations.	
M andatory	Optional	Optional
Notes		

6.4.3.2.22 NVMeControllerProperties.NVMeControllerAttributes.ReportsNamespaceGranularity

The mapping for `NVMeControllerProperties.NVMeControllerAttributes.ReportsNamespaceGranularity` is summarized in Table 106.

Table 106: NVMeControllerProperties.NVMeControllerAttributes. ReportsNamespaceGranularity mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeControllerProperties.NVMeControllerAttributes. ReportsNamespaceGranularity	**NVM Spec Property / Field:** Controller Attributes (CTRATT): Bit 7 (Namespace Granularity) **NVM Spec: Section:Figure** NVMe 1.4a: Section 5.15.2.2 (IdentifyController), Figure 249
Type	Boolean	**NVM Spec Property Type:** Single bit (bool) **Additional NVM Spec Identifying Information:** Bit 7 of Byte 99:96
Des cription	Indicates whether or not the controller supports reporting of Namespace Granularity.	
LongDes cription	This property shall indicate whether or not the controller supports reporting of Namespace Granularity.	
M andatory	Recommended for NVM Drives and more complex devices with NVMe front ends, such as opaque arrays.	Optional
Notes		

6.4.3.2.23 NVMeControllerProperties.NVMeControllerAttributes.SupportsReservations

The mapping for `NVMeControllerProperties.NVMeControllerAttributes.SupportsReservations` is summarized in Table 107.

Table 107: NVMeControllerProperties.NVMeControllerAttributes. ReportsNamespaceGranularity mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Pro perty	NVMeControll erProperties.NVMeControllerAt tributes.SupportsReservations	**NVM Spec Property / Field:** Optional NVM Command Support (ONCS): Bit 5 **NVM Spec: Section:Figure** NVMe Base specification 1.4a: Section 5.15.2.2 (IdentifyController), Figure 249
Type	Boolean	**NVM Spec Property Type:** Single bit (bool) **Additional NVM Spec Identifying Information:** Bit 5 of Byte 521:520
D escri ption	Indicates whether or not the controller supports namespace reservations.	Indicates whether or not the controller supports namespace reservations.
LongD escri ption	This property shall indicate whether or not the controller supports namespace reservations to enable two or more hosts to coordinate access to a shared namespace.	
Mand atory	Recommended for NVM Drives and more complex devices with NVMe front ends, such as opaque arrays.	Optional

	Redfish/Swordfish	NVMe / NVMe-oF
Notes		If the controller supports reservations then the following commands associated with reservations shall be supported: Reservation Report, Reservation Register, Reservation Acquire, and Reservation Release. Refer to section NVMe Base specification, Section 8.8 for additional requirements.

6.4.3.2.24 NVMeControllerProperties.NVMeControllerAttributes.SupportsTrafficBasedKeepAlive

The mapping for `NVMeControllerProperties.NVMeControllerAttributes.TrafficBasedKeepAlive` is summarized in Table 108.

Table 108: NVMeControllerProperties.NVMeControllerAttributes. TrafficBasedKeepAlive mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeControllerProperties. NVMeControllerAttributes.SupportsTrafficBasedKeepAlive	**NVM Spec Property / Field:** Controller Attributes (CTRATT): Bit 6 (Traffic Based Keep Alive Support – TBKAS) **NVM Spec: Section:Figure** NVMe 1.4a: Section 5.15.2.2 (IdentifyController), Figure 249
Type	Boolean	**NVM Spec Property Type:** Single bit (bool) **Additional NVM Spec Identifying Information:** Bit 6 of Byte 99:96
Des cription	Indicates whether or not the controller supports restarting KeepAlive Timer if traffic is processed from an admin command or IO during KeepAlive Timeout interval.	
LongDes cription	This property shall indicate whether or not the controller supports restarting KeepAlive Timer if traffic is processed from an admin command or IO during KeepAlive Timeout interval.	
M andatory	Required for Ethernet-Attach Drives; required for more complex devices with NVMe front ends, such as opaque arrays.	Optional

	Redfish/Swordfish	NVMe / NVMe-oF
Notes	For NVMe SSD Drives: If “Ethernet-Attach for NVMe Drives” feature is advertised, this is required. (This means EnduranceGroups and NVM Sets are supported.)	

Table: NVMe ControllerProperties.NVMeControllerAttributes.SupportsTrafficBasedKeepAlive mapping

6.4.3.2.25 NVMeControllerProperties.NVMeControllerAttributes.SupportsPredictableLatencyMode

The mapping for `NVMeControllerProperties.NVMeControllerAttributes.PredictableLatencyMode` is summarized in Table 109.

Table 109: NVMeControllerProperties.NVMeControllerAttributes. SupportsPredictableLatencyMode mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeControllerProperties.NVMeControllerAttributes.SupportsPredictableLatencyMode	**NVM Spec Property / Field:** Controller Attributes (CTRATT): Bit 5 (Predictable Latency Mode) **NVM Spec: Section:Figure** NVMe 1.4a: Section 5.15.2.2 (IdentifyController), Figure 249
Type	Boolean	**NVM Spec Property Type:** Single bit (bool) **Additional NVM Spec Identifying Information:** Bit 5 of Byte 99:96
Des cription	Indicates whether or not the controller supports Predictable Latency Mode.	
LongDes cription	This property shall indicate whether or not the controller supports Predictable Latency Mode.	
M andatory	Optional	Optional
Notes		

6.4.3.2.26 NVMeControllerProperties.NVMeControllerAttributes.SupportsEnduranceGroups

The mapping for `NVMeControllerProperties.NVMeControllerAttributes.EnduranceGroups` is summarized in Table 110.

Table 110: NVMeControllerProperties.NVMeControllerAttributes. SupportsEnduranceGroups mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeControllerProperties.NVMeControllerAttributes.SupportsEnduranceGroups	**NVM Spec Property / Field:** Controller Attributes (CTRATT): Bit 4 (Endurance Groups) **NVM Spec: Section:Figure** NVMe 1.4a: Section 5.15.2.2 (IdentifyController), Figure 249
Type	Boolean	**NVM Spec Property Type:** Single bit (bool) **Additional NVM Spec Identifying Information:** Bit 4 of Byte 99:96
Description	Indicates whether or not the controller supports Endurance Groups.	
LongDescription	This property shall indicate whether or not the controller supports Endurance Groups.	
M andatory	Required when EnduranceGroups/Sets are supported.	Optional
Notes	For NVMe SSD Drives: If “Advanced Features for NVMe Drives” feature is advertised, this is required. (This means EnduranceGroups and NVM Sets are supported.)	

6.4.3.2.27 NVMeControllerProperties.NVMeControllerAttributes.SupportsReadRecoveryLevels

The mapping for `NVMeControllerProperties.NVMeControllerAttributes.SupportsReadRecoveryLevels` is summarized in Table 111.

Table 111: NVMeControllerProperties.NVMeControllerAttributes. SupportsReadRecoveryLevels mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeControllerProperties.NVMeControllerAttributes.SupportsReadRecoveryLevels	**NVM Spec Property / Field:** Controller Attributes (CTRATT): Bit 3 (Read Recovery Levels) **NVM Spec: Section:Figure** NVMe 1.4a: Section 5.15.2.2 (IdentifyController), Figure 249
Type	Boolean	**NVM Spec Property Type:** Single bit (bool) **Additional NVM Spec Identifying Information:** Bit 3 of Byte 99:96
Description	Indicates whether or not the controller supports Read Recovery Levels.	
LongDescription	This property shall indicate whether or not the controller supports Read Recovery Levels.	
M andatory	Optional	Optional
Notes		

6.4.3.2.28 NVMeControllerProperties.NVMeControllerAttributes.SupportsNVMSets

The mapping for `NVMeControllerProperties.NVMeControllerAttributes.SupportsNVMSets` is summarized in Table 112.

Table 112: NVMeControllerProperties.NVMeControllerAttributes. SupportsNVMSets mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeControllerProperties.NVMeControllerAttributes.SupportsNVMSets	**NVM Spec Property / Field:** Controller Attributes (CTRATT): Bit 2 (NVM Sets) **NVM Spec: Section:Figure** NVMe 1.4a: Section 5.15.2.2 (IdentifyController), Figure 249
Type	Boolean	**NVM Spec Property Type:** Single bit (bool) **Additional NVM Spec Identifying Information:** Bit 2 of Byte 99:96
Description	Indicates whether or not the controller supports NVM Sets.	
LongDescription	This property shall indicate whether or not the controller supports NVM Sets.	
M andatory	Required when EnduranceGroups/Sets are supported.	Optional
Notes	For NVMe SSD Drives: If “Advanced Features for NVMe Drives” feature is advertised, this is required. (This means EnduranceGroups and NVM Sets are supported.)	

6.4.3.2.29 NVMeControllerProperties.NVMeControllerAttributes.SupportsExceedingPowerOfNonOpera

The mapping for `NVMeControllerProperties.NVMeControllerAttributes.SupportsExceedingPowerOfNonOp` is summarized in Table 113.

Table 113: NVMeControllerProperties.NVMeControllerAttributes. SupportsExceeding-PowerOfNonOperationalState mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeControll erProperties.NVMeController Attributes.SupportsExceedin gPowerOfNonOperationalState	**NVM Spec Property / Field:** Controller Attributes (CTRATT): Bit 1 (Non-Operational Power State Permissive Mode) **NVM Spec: Section:Figure** NVMe 1.4a: Section 5.15.2.2 (IdentifyController), Figure 249
Type	Boolean	**NVM Spec Property Type:** Single bit (bool) **Additional NVM Spec Identifying Information:** Bit 1 of Byte 99:96
Des cription	Indicates whether or not the controller supports exceeding Power of NonOperational State in order to execute controller initiated background operations in a non-operational power state.	
LongDes cription	This property shall indicate whether or not the controller supports exceeding Power of NonOperational State in order to execute controller initiated background operations in a non-operational power state.	
M andatory	Optional	Optional
Notes		

6.4.3.2.30 NVMeControllerProperties.NVMeControllerAttributes.Supports128BitHostId

The mapping for `NVMeControllerProperties.NVMeControllerAttributes.Supports128BitHostId` is summarized in Table 114.

Table 114:
NVMeControllerProperties.NVMeControllerAttributes. Supports128BitHostId mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeControllerProperties.NVMeControllerAttributes.Supports128BitHostId	**NVM Spec Property / Field:** Controller Attributes (CTRATT): Bit 0 **NVM Spec: Section:Figure** NVMe 1.4a: Section 5.15.2.2 (IdentifyController), Figure 249
Type	Boolean	**NVM Spec Property Type:** Single bit (bool) **Additional NVM Spec Identifying Information:** Bit 0 of Byte 99:96
Description	Indicates whether or not the controller supports a 128-bit Host Identifier.	
LongDescription	This property shall indicate whether or not the controller supports a 128-bit Host Identifier.	
M andatory	Required	Optional
Notes		

6.4.3.2.31 NVMeControllerProperties.MaxQueueSize

The mapping for `NVMeControllerProperties.MaxQueueSize` is summarized in Table 115.

Table 115: NVMeControllerProperties.ANACharacteristics mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeContro llerProperties.MaxQueueSize	**NVM Spec Property / Field:** Maximum Queues Entries Supported (MQES) **NVM Spec: Section:Figure** NVMe 1.4a; Section 3.1.1 Controller Capabilities; Figure 69
Type	Int64	**NVM Spec Property Type: Additional NVM Spec Identifying Information:** ByteOffset: Bits 15:00
Des cription	Indicates the maximum individual queue size that an NVMe IO Controller supports.	
LongDes cription	This property shall contain the maximum individual queue entry size supported per queue. This is a zero-based value, where the minimum value is one, indicating two entries. For PCIe, this applies to both submission and completion queues. For NVMe-oF, this applies to only submission queues.	
M andatory	Implement for more complex devices with NVMe front ends, such as opaque arrays.	Required
Notes		

6.4.3.2.32 NVMeControllerProperties.ANACharacteristics The mapping for `NVMeControllerProperties.ANACharacteristics` is summarized in Table 116.

Table 116: NNVMeControllerProperties.ANACharacteristics mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeControllerPr operties.ANACharacteristics	
Type	Co llection(StorageController. v1_0_0.ANACharacteristics)	
Des cription	This property contains the combination of ANA type and volume information.	
LongDes cription	This property shall contain the combination of ANA type and volume information.	
M andatory		
Notes		

6.4.3.2.33 NVMeControllerProperties.ANACharacteristics.AccessState The mapping for `NVMeControllerProperties.ANACharacteristics.AccessState` is summarized in Table 117.

Table 117: NNVMeControllerProperties.ANACharacteristics.AccessState mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	N VMeControllerProperties.ANA Characteristics.AccessState	**NVM Spec Property / Field:** Asymmetric Namespace Access State **NVM Spec: Section:Figure** NVMe 1.4a; Section 5.14.1.12; Figure 211
Type	StorageContro ller.v1_0_0.ANAAccessState	**NVM Spec Property Type: Additional NVM Spec Identifying Information:** ByteOffset: Bits 03:00 of Byte 16
Des cription	Reported ANA Access state.	
LongDes cription	This property shall contain the reported ANA Access State.	
M andatory		
Notes	Available values: Optimized / NonOptimized / Inaccessible / PersistentLoss	

6.4.3.2.34 NVMeControllerProperties.ANACharacteristics.Volume The mapping for `NVMeControllerProperties.ANACharacteristics.Volume` is summarized in Table 118.

Table 118: NNVMeControllerProperties.ANACharacteristics.Volume mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeControllerPropertie s.ANACharacteristics.Volume	**NVM Spec Property / Field:** Namespace Identifier X: **NVM Spec: Section:Figure** NVMe 1.4a; Section 5.14.1.12; Figure 211
Type	Volume.Volume	**NVM Spec Property Type: Additional NVM Spec Identifying Information:** Bits 35:32, 39:36, …, ((n*4) + 35):
((n*4) + 32) - up to “n” n amespace iden tifiers.		
Des cription	The specified volume.	
LongDes cription	This property shall contain a link to the specified volume.	
M andatory		
Notes	This field contains the pointer to the namespace for which the access state applies.	The namespace id should be redirected / linked to the corresponding namespace (volume) object with that namespace id. If this set of fields contains multiple namespaces (e.g., a group of namespaces), a unique entry in the ANACharacteristics array should be created for each namespace.

6.4.3.2.35 NVMeControllerProperties.NVMeSMARTCriticalWarnings.PRMUnreliable

The mapping for `NVMeControllerProperties.NVMeSMARTCriticalWarnings.PMRUnreliable` is summarized in Table 119.

Table 119: NNVMeControllerProperties.NVMeSMARTCriticalWarnings.PRMunreliable mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeControllerProperties.NVMeSMARTCriticalWarnings.PMRUnreliable	**NVM Spec Property / Field:** Critical Warning **NVM Spec: Section:Figure** NVMe 1.4a: Section 5.14.1.2, SMART / Health Information, Figure 196
Type	Boolean	**NVM Spec Property Type:** Single bit (bool) **Additional NVM Spec Identifying Information:** Bit 5 of Byte 00
Des cription	The Persistent Memory Region has become unreliable.	
LongDes cription	This property shall indicate that the Persistent Memory Region has become unreliable. PCI Express memory reads may return invalid data or generate poisoned PCI Express TLP(s). Persistent Memory Region memory writes may not update memory or may update memory with undefined data. The Persistent Memory Region may also have become non-persistent.	
M andatory	Recommended for NVM Drives; required for more complex devices with NVMe front ends, such as opaque arrays.	

6.4.3.2.36 NVMeControllerProperties.NVMeSMARTCriticalWarnings.PowerBackupFailed

The mapping for `NVMeControllerProperties.NVMeSMARTCriticalWarnings.PowerBackupFailed` is summarized in Table 120.

Table 120:
NVMeControllerProperties.NVMeSMARTCriticalWarnings. PowerBackupFailed mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeController Properties.NVMeSMARTCritica lWarnings.PowerBackupFailed	**NVM Spec Property / Field:** Critical Warning **NVM Spec: Section:Figure** NVMe 1.4a: Section 5.14.1.2, SMART / Health Information, Figure 196
Type	Boolean	**NVM Spec Property Type:** Single bit (bool) **Additional NVM Spec Identifying Information:** Bit 4 of Byte 00
Des cription	Indicates that the volatile memory backup device has failed.	
LongDes cription	This property shall indicate that the volatile memory backup device has failed.	
M andatory	Recommended for NVM Drives. Required for more complex devices with NVMe front ends, such as opaque arrays.	
Notes		

6.4.3.2.37 NVMeControllerProperties.NVMeSMARTCriticalWarnings.MediaInReadOnly

The mapping for `NVMeControllerProperties.NVMeSMARTCriticalWarnings.MediaInReadOnly` is summarized in Table 121.

Table 121: NVMeControllerProperties.NVMeSMARTCriticalWarnings.MediaInReadOnly mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeControllerProperties.NVMeSMARTCriticalWarnings.MediaInReadOnly	**NVM Spec Property / Field:** Critical Warning **NVM Spec: Section:Figure** NVMe 1.4a: Section 5.14.1.2, SMART / Health Information, Figure 196
Type	Boolean	**NVM Spec Property Type:** Single bit (bool) **Additional NVM Spec Identifying Information:** Bit 3 of Byte 00
Description	Indicates the media has been placed in read only mode.	
LongDescription	This property shall indicate the media has been placed in read only mode. This is not set when the read-only condition on the media is a result of a change in the write protection state of a namespace.	
M andatory	Required	
Notes		

6.4.3.2.38 NVMeControllerProperties.NVMeSMARTCriticalWarnings.OverallSystemDegraded

The mapping for `NVMeControllerProperties.NVMeSMARTCriticalWarnings.OverallSubsystemDegraded` is summarized in Table 122.

Table 122: NVMeControllerProperties.NVMeSMARTCriticalWarnings. OverallSubsystemDegraded mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeControllerProperties.NVMeSMARTCriticalWarnings.OverallSubsystemDegraded	**NVM Spec Property / Field:** Critical Warning **NVM Spec: Section:Figure** NVMe 1.4a: Section 5.14.1.2, SMART / Health Information, Figure 196
Type	Boolean	**NVM Spec Property Type:** Single bit (bool) **Additional NVM Spec Identifying Information:** Bit 2 of Byte 00
Des cription	Indicates that the NVM subsystem reliability has been compromised.	
LongDes cription	This property shall indicate that the NVM subsystem reliability has been compromised.	
M andatory	Required	
Notes		

6.4.3.2.39 NVMeControllerProperties.NVMeSMARTCriticalWarnings.SpareCapacityWornOut

The mapping for `NVMeControllerProperties.NVMeSMARTCriticalWarnings.SpareCapacityWornOut` is summarized in Table 123.

Table 123:
NVMeControllerProperties.NVMeSMARTCriticalWarnings. SpareCapacityWornOut mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeControllerPro perties.NVMeSMARTCriticalWa rnings.SpareCapacityWornOut	**NVM Spec Property / Field:** Critical Warning **NVM Spec: Section:Figure** NVMe 1.4a: Section 5.14.1.2, SMART / Health Information, Figure 196
Type	Boolean	**NVM Spec Property Type:** Single bit (bool) **Additional NVM Spec Identifying Information:** Bit 0 of Byte 00
Des cription	Indicates that the available spare capacity has fallen below the threshold.	
LongDes cription	This property shall indicate that the available spare capacity has fallen below the threshold.	
M andatory	Required for NVMe drives, as well as for more complex devices with NVMe front ends, such as opaque arrays.	
Notes		

6.4.3.2.40 PCIeInterface.PCIeType The mapping for `PCIeInterface.PCIeType` is summarized in Table 124.

Table 124: PCIeInterface.PCIeType mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	PCIeInterface.PCIeType	N/A
Type	enum (PCIeDevice.PCIeType)	N/A
Des cription	The version of the PCIe specification in use by this device.	N/A
LongDes cription	This property shall contain the negotiated PCIe interface version in use by this device.	N/A
M andatory	Required for PCIe attach NVMe Drives; do not implement for ethernet-attach drives. Optional for more complex devices with NVMe front ends, such as opaque arrays.	N/A
Notes	Possible values: Gen1/Gen2/Gen3/Gen4/Gen5	The functionality comes from the underlying implementation and does not originate in the NVMe specs

6.4.3.2.41 PCIeInterface.MaxPCIeType

The mapping for `PCIeInterface.MaxPCIeType` is summarized in Table 125.

Table 125: PCIeInterface.MaxPCIeType mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	PCIeInterface.MaxPCIeType	N/A
Type	enum (PCIeDevice.PCIeType)	N/A
Des cription	The highest version of the PCIe specification supported by this device.	N/A
LongDes cription	This property shall contain the maximum PCIe specification that this device supports.	N/A
M andatory	Required for PCIe attach NVMe Drives; do not implement for ethernet-attach drives.	N/A
Notes	Possible values: Gen1/Gen2/Gen3/Gen4/Gen5	The functionality comes from the underlying implementation and does not originate in the NVMe specs

6.4.3.2.42 PCIeInterface.LanesInUse The mapping for `PCIeInterface.LanesInUse` is summarized in Table 126.

Table 126: PCIeInterface.LanesInUse mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	PCIeInterface.LanesInUse	N/A
Type	int64	N/A
Des cription	The number of PCIe lanes in use by this device.	N/A
LongDes cription	This property shall contain the number of PCIe lanes in use by this device, which shall be equal to or less than the MaxLanes property value.	N/A
M andatory	Required for PCIe attach NVMe Drives; do not implement for ethernet-attach drives.	N/A
Notes		The functionality comes from the underlying implementation and does not originate in the NVMe specs

6.4.3.2.43 PCIeInterface.MaxLanes The mapping for `PCIeInterface.LanesInUse` is summarized in Table 127.

Table 127: PCIeInterface.LanesInUse mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	PCIeInterface.LanesInUse	N/A
Type	int64	N/A
Des cription	The number of PCIe lanes supported by this device.	N/A
LongDes cription	This property shall contain the maximum number of PCIe lanes supported by this device.	N/A
M andatory	Required for PCIe attach NVMe Drives; do not implement for ethernet-attach drives.	N/A
Notes		The functionality comes from the underlying implementation and does not originate in the NVMe specs

6.4.3.2.44 Ports The mapping for `Ports` is summarized in Table 128.

Table 128: Ports mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Ports	**NVM Spec Property / Field:** N/A **NVM Spec: Section:Figure** N/A
Type	Po rtCollection.PortCollection	N/A
Des cription	The link to the collection of ports that exist on the storage controller.	N/A
LongDes cription	This property shall contain a link to a resource collection of type PortCollection.	N/A
M andatory	Do Not Implement for NVMe drives; optional for more complex devices with NVMe front ends, such as opaque arrays.	
Notes		

6.4.3.2.45 SKU The mapping for SKU is summarized in Table 129.

Table 129: SKU mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	SKU	**NVM Spec Property / Field:** N/A **NVM Spec: Section:Figure** N/A
Type	Edm.String	N/A
Des cription	The SKU for this storage controller.	N/A
LongDes cription	This property shall contain the stock-keeping unit number for this storage storage controller.	N/A
M andatory	Do Not Implement	
Notes		

6.4.3.2.46 SpeedGbps The mapping for `SpeedGbps` is summarized in Table 130.

Table 130: SpeedGbps mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	SpeedGbps	**NVM Spec Property / Field:** N/A **NVM Spec: Section:Figure** N/A
Type	Decimal	N/A
Des cription	The maximum speed of the storage controller's device interface.	N/A
LongDes cription	This property shall represent the maximum supported speed of the storage bus interface, in Gbit/s. The specified interface connects the controller to the storage devices, not the controller to a host. For example, SAS bus not PCIe host bus.	N/A
M andatory	Do Not Implement	
Notes		

6.4.3.2.47 Status.State The mapping for `Status.State` is summarized in Table 131.

Table 131: Status.State mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Status.State	**NVM Spec Property / Field:** CSTS – Controller Status **NVM Spec: Section:Figure** NVMe 1.4a: Section 3.1.6, Figure 79
Type	Resource.State (enum)	**NVM Spec Property Type:**
Des cription	The known state of the resource, such as, enabled.	
LongDes cription	This property shall indicate whether and why this component is available. Enabled indicates the resource is available. Disabled indicates the resource has been intentionally made unavailable but it can be enabled. Offline indicates the resource is unavailable intentionally and requires action to make it available. InTest indicates that the component is undergoing testing. Starting indicates that the resource is becoming available. Absent indicates the resource is physically unavailable.	
M andatory		Mandatory

	Redfish/Swordfish	NVMe / NVMe-oF
Notes	Possible values: Enabled / Disabled / StandbyOffline / StandbySpare / InTest / Starting / ABsent / UnavaialableOffline / Deferring / Quiesced / Updating / Qualified	Ready (CSTS.RDY) maps to Enabled, Shutdown (CSTS.SHST) value will tell you if shutdown is in progress or complete (StandbyOffline), ProcessingPaused (CSTS.PP) maps to Deferring. If both Ready and Shutdown are indicated, then the system should indicate StandbyOffline. If both Ready and ProcessingPaused are indicated, then the system should indicate Deferring.

6.4.3.2.48 Status.Health The mapping for `Status.Health` is summarized in Table 132.

Table 132: Status.Health mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Status.Health	**NVM Spec Property / Field:** CSTS – Controller Status **NVM Spec: Section:Figure** NVMe 1.4a: Section 3.1.6, Figure 79 **NVM Spec Property / Field:** Critical Warning **NVM Spec: Section:Figure** NVMe 1.4a: Section 5.14.1.2, SMART / Health Information, Figure 196
Type	Resource.Health	**NVM Spec Property Type:**
Des cription	The health state of this resource in the absence of its dependent resources.	
LongDes cription	This property shall represent the health state of the resource without considering its dependent resources. The values shall conform to those defined in the Redfish Specification.	
M andatory		
Notes	Possible Values: OK / Warning / Critical	This comes from CSTS Controller Failure Status, and from the SMART / health information log critical warning field.

6.4.3.2.49 SupportedControllerProtocols The mapping for `SupportedControllerProtocols` is summarized in Table 133.

Table 133: SupportedControllerProtocols mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	S upportedControllerProtocols	N/A
Type	Co llection(Protocol.Protocol)	**NVM Spec Property Type:** N/A
Des cription	The supported set of protocols for communicating to this storage controller.	
LongDes cription	This property shall contain the supported set of protocols for communicating to this storage controller.	
M andatory		
Notes	This is an array of protocols supported by the StorageController. This can be set to values including, but not limited to, PCIe, RDMA, NVMe-oF, RoCE, RoCEv2, and InfiniBand.	

6.4.3.2.50 SupportedDeviceProtocols The mapping for `SupportedDeviceProtocols` is summarized in Table 134.

Table 134: SupportedDeviceProtocols mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	SupportedDeviceProtocols	N/A
Type	Co llection(Protocol.Protocol)	**NVM Spec Property Type:** N/A
Des cription	The protocols that the storage controller can use to communicate with attached devices.	
LongDes cription	This property shall contain the set of protocols this storage controller can use to communicate with attached devices.	
M andatory		
Notes	Do not implement.	

6.5 Namespace

The Redfish/Swordfish Volume schema is used to represent an NVM Namespace.

6.5.1 Mockup

The following mockup shows a sample representation of the Volume schema used to represent an NVM Namespace.

```
{
  "@Redfish.Copyright": "Copyright 2014-2020 SNIA. All rights reserved.",
  "@odata.id":
  ↪  "/redfish/v1/Storage/NVMe-oF-Subsystem/Volumes/LogicalNamespace1",
  "@odata.type": "#Volume.v1_5_0.Volume",
  "Id": "1",
  "Name": "Namespace 1",
  "LogicalUnitNumber": 1,
  "Status": {
    "State": "Enabled",
    "Health": "OK"
  },
  "Identifiers": [{
    "DurableNameFormat": "NGUID",
    "DurableName": "FEDCBA9876543210h"
  }],
  "Capacity": {
    "Data": {
      "ConsumedBytes": 0,
      "AllocatedBytes": 10737418240
    },
    "Metadata": {
      "AllocatedBytes": 536870912
    }
  },
  "CapacitySources": [{
    "@odata.id": "/redfish/v1/Storage/NVMe-oF-
    ↪  Subsystem/Volumes/LogicalNamespace1/CapacitySources/Source1"

  }],
  "NVMeNamespaceProperties": {
    "NamespaceId": "0x014",
    "NamespaceFeatures": {
      "SupportsThinProvisioning": false,
      "SupportsAtomicTransactionSize": false,
      "SupportsDeallocatedOrUnwrittenLBError": false,
      "SupportsNGUIDReuse": false,
      "SupportsIOPerformanceHints": false
    },
    "NumberLBAFormats": 0,
    "FormattedLBASize": "LBAFormat0Support",
    "MetadataTransferredAtEndOfDataLBA": false,

    "NVMeVersion": "1.4"
  }
}
```

6.5.2 Property Mapping

6.5.2.1 BlockSizeBytes

The mapping for `BlockSizeBytes` is summarized in Table 135.

Table 135: BlockSizeBytes mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	BlockSizeBytes	**NVM Spec Property / Field:** Formatted LBA Size (FLBAS) **NVM Spec: Section:Figure** 247: byte 26
Type	Int64	
Des cription	The size, in bytes, of the smallest addressable unit, or block.	
LongDes cription	This property shall contain size of the smallest addressable unit of the associated drive or device.	
M andatory	Required	
Notes		Follow index in 247: Byte 26 (bits 3:0) to the LBA format structure to get the metadata size and LBA data size combination. These combined values are BlockSizeBytes.

6.5.2.2 Capacity.Data.ConsumedBytes

The mapping for `Capacity.Data.ConsumedBytes` is summarized in Table 136.

Table 136: Capacity.Data.ConsumedBytes mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Capacity.Data.ConsumedBytes	**NVM Spec Property / Field:** Namespace Utilization (NUSE) **NVM Spec: Section:Figure** NVMe 1.4a: Section 5.15.2.1 (Identify Namespace), Figure 247
Type	Int64	**NVM Spec Property Type:** int 64 **Additional NVM Spec Identifying Information:** ByteOffset: 23:16, Identify Namespace data structure
Des cription	The number of bytes consumed in this data store for this data type.	The current number of logical blocks allocated in the namespace.
LongDes cription	The value shall be the number of logical bytes currently consumed in this data store for this data type.	This field indicates the current number of logical blocks allocated in the namespace. This field is smaller than or equal to the Namespace Capacity. The number of logical blocks is based on the formatted LBA size.
M andatory	Yes	Yes
Notes	Reporting capacity in bytes is the Redfish and Swordfish standard mechanism. Clients expect the capacity information to be reported consistently for these devices, so the calculation here is to convert the NVMe properties (in blocks) to bytes.	Returned in bytes 23:16 of the Identify Namespace Data Structure (NVM Command Set Specific). Reference NVMe Base Specification section n 5.15.2.1 and figure 247).

6.5.2.3 Capacity.Data.ProvisionedBytes

The mapping for `Capacity.Data.ProvisionedBytes` is summarized in Table 137.

Table 137: Capacity.Data.ProvisionedBytes mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Cap acity.Data.ProvisionedBytes	**NVM Spec Property / Field:** NVM Capacity (NCAP) **NVM Spec: Section:Figure** NVMe 1.4a: Section 5.15.2.1 (Identify Namespace), Figure 247
Type	Int64	**NVM Spec Property Type:** int 64 **Additional NVM Spec Identifying Information:** ByteOffset: 15:08, Identify Namespace data structure
Des cription	The maximum number of bytes that can be allocated in this data store for this data type.	The total size of the namespace in logical blocks (LBA 0 through n-1).
LongDes cription	The value shall be the maximum number of bytes that can be allocated in this data store for this data type.	This field indicates the total size of the namespace in logical bytes. The value is in bytes. A namespace of size n consists of LBA 0 through (n - 1). The number of logical blocks is based on the formatted LBA size. This field is undefined prior to the namespace being formatted.
M andatory	No	No
Notes	This property is required when issuing a create namespace command. It is also required for “change namespace” when modifying the size of the namespace.	Returned in bytes 07:00 of the Identify Namespace Data Structure (NVM Command Set Specific). Reference NVMe Base Specification section in 5.15.2.1 and figure 247.

6.5.2.4 Capacity.Data.AllocatedBytes The mapping for `Capacity.Data.AllocatedBytes` is summarized in Table 138.

Table 138: Capacity.Data.AllocatedBytes mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	C apacity.Data.AllocatedBytes	**NVM Spec Property / Field:** Namespace Size (NSZE) **NVM Spec: Section:Figure** NVMe 1.4a: Section 5.15.2.1 (Identify Namespace), Figure 247
Type	Int64	**NVM Spec Property Type:** int 64 **Additional NVM Spec Identifying Information:** ByteOffset: 07:00, Identify Namespace data structure
Des cription	The number of bytes currently allocated by the storage system in this data store for this data type.	The total size of the NVM allocated to this namespace.
LongDes cription	The value shall be the number of bytes currently allocated by the storage system in this data store for this data type.	The total size of the NVM allocated to this namespace. The value is in bytes. This field shall be supported if the Namespace Management capability (refer to NVMe Base Specification section 8.12) is supported. This field may not correspond to the logical block size multiplied by the Namespace Size field. Due to thin provisioning or other settings (e.g., endurance), this field may be larger or smaller than the Namespace Size reported.
M andatory	No	No

	Redfish/Swordfish	NVMe / NVMe-oF
Notes	Reporting capacity in bytes is the Redfish and Swordfish standard mechanism.	Returned in bytes 63:48 of the Identify Namespace Data Structure (NVM Command Set Specific). Reference NVMe Base Specification section in 5.15.2.1 and figure 247.

6.5.2.5 Capacity.Metadata The mapping for `Capacity.Metadata` is summarized in Table 139.

Table 139: Capacity.Metadata.AllocatedBytes mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Capacity.Metadata.*	N/A
Type	Int64	N/A
Des cription	The number of bytes currently allocated by the storage system in this data store for this data type.	N/A
LongDes cription	The value shall be the number of bytes currently allocated by the storage system in this data store for this data type.	N/A
M andatory	No; not required for NVMe Drives; optional to implement for more complex devices.	No
Notes		Do not return metadata information for NVMe devices. This is included in the overall reported capacity information.

6.5.2.6 CapacitySources The mapping for `CapacitySources` is summarized in Table 140.

Table 140: CapacitySources mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	CapacitySources	**NVM Spec Property / Field:** NVM Set Identifier (NVMSETID) **NVM Spec: Section:Figure** NVMe 1.4a: Section 5.15.2.1 (Identify Namespace), Figure 247
Type	Collecti on(Capacity.CapacitySource)	**NVM Spec Property Type:** int 64 **Additional NVM Spec Identifying Information:** ByteOffset: 101:100, Identify Namespace data structure
Des cription	An array of space allocations to this volume.	The NVM Set in which the namespace is allocated
LongDes cription	Fully or partially consumed storage from a source resource. Each entry provides capacity allocation information from a named source resource.	
M andatory	No	No

	Redfish/Swordfish	NVMe / NVMe-oF
Notes	Contains the information about the entity providing capacity (e.g, EnduranceGroup) for this namespace. This property is a collection pointer; each CapacitySource instance contains a CapacitySource object. The CapacitySource contains the overall capacity (in bytes), the types and pointers to the underlying capacity sources.	For NVMe devices that do *not* implement Endurance Groups and NVM Sets, it is recommended that the CapacitySource not be implemented. For Implementations that do instantiate Endurance Groups and NVM Sets, the capacity source should be implemented as a pointer to the corresponding Endurance Group. (See mockups for examples.)

6.5.2.7 Description

The mapping for `Description` is summarized in Table 141.

Table 141: Description mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Description	N/A
Type	String	N/A
Des cription	The description of this resource.	N/A
LongDes cription	This object represents the description of this resource. The resource values shall comply with the Redfish Specification-described requirements.	N/A
M andatory	Yes	N/A

	Redfish/Swordfish	NVMe / NVMe-oF
Notes	In Redfish, Description is a read-only field.	Return the common description: “A Namespace is a quantity of non-volatile memory that may be formatted into logical blocks. When formatted, a namespace of size n is a collection of logical blocks with logical block addresses from 0 to (n-1). NVMe systems can support multiple namespaces.”

6.5.2.8 DisplayName

The mapping for `DisplayName` is summarized in Table 142.

Table 142: DisplayName mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	DisplayName	N/A
Type	String	N/A
Des cription	A user-configurable string to name the volume.	N/A
LongDes cription	his property shall contain a user-configurable string to name the volume.	N/A
M andatory	Recommended for NVMe Drives, as well as for more complex devices with NVMe front ends, such as opaque arrays.	N/A

	Redfish/Swordfish	NVMe / NVMe-oF
Notes	This contains an (end) user settable “friendly” name for the namespace. This may map to a property that shows up in the proprietary software, as long as it is configurable by the end-user, and is intended to contain a displayable string.	

6.5.2.9 Identifiers

The mapping for `Identifiers` is summarized in Table 143.

Table 143: Identifiers mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Identifiers	Namespace Identification Descriptor list
Type	Coll ection(Resource.Identifier)	A variable length Namespace Identification Descriptor structures
Des cription	The Durable names for the storage controller.	A list of Namespace Identification Descriptor structures containing Namespace Type, Namespace Identifier Length (NIDL), and Namespace ID (NID).

	Redfish/Swordfish	NVMe / NVMe-oF
LongDescription	This property shall contain a list of all known durable names for the associated storage controller.	A list of any number of variable length Namespace Identification Descriptor structures that fit into the 4,096 byte Identify payload. All remaining bytes after the namespace identification descriptor structures should be cleared to 0h, and the host shall interpret a Namespace Identifier Descriptor Length (NIDL) value of 0h as the end of the list. The host should ignore any Namespace Identification Descriptor with a Namespace Identifier Type not supported by the host.
M andatory	No	No
Notes	This is an array of unique identifiers for the NVM Subsystem including Namespace Type and Namespace ID.	Refer to NVMe Base Specification Figure 246 CNS 03h and Figure 251 (Identify – Namespace Identification Descriptor).

6.5.2.10 Identifiers.DurableName

The mapping for `Identifiers.DurableName` is summarized in Table 144.

Table 144: Identifiers.DurableName mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Identifiers.DurableName	Namespace Identifier (NID)
Type	Variable - see notes	Variable - see notes
Description	The Durable names for the storage controller.	Durable Namespace Identifier

	Redfish/Swordfish	NVMe / NVMe-oF
LongDes cription	This property shall contain a list of all known durable names for the Namespace.	A list of globally unique values assigned to the Namespace when the Namespace is created. Values remain fixed throughout the life of the Namespace and are preserved across Namespace and Controller operations (e.g., Controller Level Reset, Namespace format, etc.).
M andatory	No	No
Notes	This is an array of unique identifiers for the Namespace. Type and length of descriptor are in the corresponding Id entifiers.DurableNameFormat property.	This is an array of unique identifiers for the NVM Namespace. Type and length of the descriptor are in the corresponding Namespace Identifier Type (NIDT). Refer to NVMe Base Specification Figure 246 CNS 03h and Figure 251 - Figure 251 Byte NID of Identify – Namespace Identification Descriptor.

6.5.2.11 Identifiers.DurableNameFormat The mapping for `Identifiers.DurableNameFormat` is summarized in Table 145.

Table 145: Identifiers.DurableNameFormat mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Id entifiers.DurableNameFormat	Namespace Identifier Type (NIDT)
Type	Int64	Int64
Des cription	The Durable names for the storage controller.	The Namespace Identifier data type and length.

	Redfish/Swordfish	NVMe / NVMe-oF
LongDes cription	This property shall contain a list of the types for all known durable names for the associated storage controller. The type determines the length of the corresponding Namespace ID	the data type contained in the Namespace Identifier field and the length of that type as defined in the NVMe Base Specification Figure 251 - Identify (Namespace Identification Descriptor) byte 00. Allowed values are: 1 = an 8-byte IEEE Extended Unique Identifier . 2 = a 10-byte Namespace Globally Unique Identifier. 3 = an 8-byte Namespace UUID
M andatory	No	No
Notes	This is an array of types for the unique identifiers for the NVM Subsystem. Values may be ‘EUI64’, ‘NGUID’, or ‘UUID’.	Refer to NVMe Base Specification Figure 246 CNS 03h and Figure 251 - Figure 251 (NIDT) of Identify – Namespace Identification Descriptor.

6.5.2.12 InitializeMethod The mapping for `InitializeMethod` is summarized in Table 146.

Table 146: InitializeMethod mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	InitializeMethod	N/A
Type	Volume.InitializeMethod (enum)	
Des cription	Indicates the Initialization Method used for this volume. If InitializeMethod is not specified, the InitializeMethod should be Foreground.	

	Redfish/Swordfish	NVMe / NVMe-oF
LongDescription	This property shall indicate the initialization method used for this volume. If InitializeMethod is not specified, the InitializeMethod should be Foreground. This value reflects the most recently used Initialization Method, and may be changed using the Initialize Action.	
M andatory	Recommended for NVMe Drives, as well as for more complex devices with NVMe front ends, such as opaque arrays.	
Notes	Available values: Fast / Slow	Not in NVMe Specification today.

6.5.2.13 Links.Drives The mapping for `Links.Drives` is summarized in Table 147.

Table 147: Links.Drives mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Links.Drives	
Type	Collection(Drive.Drive)	
Des cription	An array of the drives to be used by the volume	
LongDescription	This parameter shall contain an array of the drives to be used by the volume.	

	Redfish/Swordfish	NVMe / NVMe-oF
M andatory	Required for NVMe Drives. Optional, not recommended for more complex devices with an NVMe front end, such as opaque arrays; for these configurations the mapping is to the underlying storage pool, rather than to drives.	Redfish construct. Not in the NVMe spec.
Notes	This array shall contain links to the drive object for this namespace.	

6.5.2.14 LogicalUnitNumber The mapping for `LogicalUnitNumber` is summarized in Table 148.

Table 148: LogicalUnitNumber mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	LogicalUnitNumber	N/A
Type	Int64	N/A
Des cription	Indicates the host-visible LogicalUnitNumber assigned to this Volume.	N/A
LongDes cription	This property shall contain host-visible LogicalUnitNumber assigned to this Volume. This property shall only be used when in a single connect configuration and no StorageGroup configuration is used.	N/A
M andatory	No	No

	Redfish/Swordfish	NVMe / NVMe-oF
Notes	Do not use with NVMe devices. This is represented more correctly with (NVMeNamespaceProperties).NamespaceId.	Do not implement.

6.5.2.15 MaxBlockSizeBytes The mapping for `MaxBlockSizeBytes` is summarized in Table 149.

Table 149: MaxBlockSizeBytes mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	MaxBlockSizeBytes	**NVM Spec Property / Field:** Formatted LBA Size (FLBAS) **NVM Spec: Section:Figure** 247: byte 26
Type	Int64	
Des cription	The size, in bytes, of the smallest addressable unit, or block.	
LongDes cription	This property shall contain size of the smallest addressable unit of the associated drive or device.	
M andatory	Recommended to not implement; however, if implemented this should have the same value as BlockSizeBytes.	
Notes		Follow index in 247: Byte 26 (bits 3:0) to the LBA format structure to get the metadata size and LBA data size combination. These combined values are BlockSizeBytes.

6.5.2.16 Name

The mapping for `Name` is summarized in Table 150.

Table 150: Name mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Name	**NVM Spec Property / Field:** Namespace ID (NSID) **NVM Spec: Section:Figure** NVMe 1.4a
Type	String	
Des cription	The name of the resource or array member.	N/A
LongDes cription	This object represents the name of this resource or array member. The resource values shall comply with the Redfish Specification-described requirements. This string value shall be of the 'Name' reserved word format.	To determine the active NSIDs for a particular controller, the host may follow either of the following methods: 1. Issue an Identify command with the CNS field cleared to 0h for each valid NSID (based on the Number of Namespaces value (i.e., MNAM field or NN field) in the Identify Controller data structure). If a non-zero data structure is returned for a particular NSID, then that is an active NSID; or 2. Issue an Identify command with a CNS field set to 2h to retrieve a list of up to 1,024 active NSIDs. If there are more than 1,024 active NSIDs, continue to issue Identify commands with a CNS field set to 2h until all active NSIDs are retrieved.
M andatory	Yes	N/A
Notes	In Redfish, Name is a read-only field.	Map the NSID field to a string with the format: "0xABCD"

6.5.2.17 NVMeNamespaceProperties.NamespaceId The mapping for `NVMeNamespaceProperties.NamespaceId` is summarized in Table 151.

Table 151: NVMeNamespaceProperties.NamespaceId mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeName spaceProperties.NamespaceId	Namespace Identifier (NSID)
Type	String	8-byte value
Des cription	The NVMe Namespace Identifier for this namespace.	An identifier used by a controller to provide access to a namespace.
LongDes cription	This property shall contain the NVMe Namespace Identifier for this namespace. This property shall be a hex value. Namespace identifiers are not durable and do not have meaning outside the scope of the NVMe subsystem. NSID 0x0, 0xFFFFFFFF, 0xFFFFFFFE are special purpose values.	An identifier used by a controller to provide access to a namespace or the name of the field in the SQE that contains the namespace identifier.
M andatory	Yes	Yes

	Redfish/Swordfish	NVMe / NVMe-oF
Notes		To determine the active NSIDs for a particular controller, the host may follow either of the following methods: 1. Issue an Identify command with the CNS field cleared to 0h for each valid NSID (based on the Number of Namespaces value (i.e., MNAM field or NN field) in the Identify Controller data structure). If a non-zero data structure is returned for a particular NSID, then that is an active NSID; or 2. Issue an Identify command with a CNS field set to 2h to retrieve a list of up to 1,024 active NSIDs. If there are more than 1,024 active NSIDs, continue to issue Identify commands with a CNS field set to 2h until all active NSIDs are retrieved.

6.5.2.18 NVMeNamespaceProperties.IsShareable

The mapping for `NVMeNamespaceProperties.IsShareable` is summarized in Table 152.

Table 152: NVMeNamespaceProperties.IsShareable mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeName spaceProperties.IsShareable	**NVM Spec Property / Field:** Namespace Multi-path I/O and Namespace Sharing Capabilities (NMIC) **NVM Spec: Section:Figure** NVMe 1.4a: Section 5.15.2.1 (Identify Namespace), Figure 247
Type	Boolean	**NVM Spec Property Type:** Single bit (bool) **Additional NVM Spec Identifying Information:** Bit 0 of Byte 30
Des cription	Indicates the namespace is shareable.	Specifies multi-path I/O and namespace sharing capabilities of the namespace.
LongDes cription	This property shall indicate whether the namespace is shareable.	If set to '1', then the namespace may be attached to two or more controllers in the NVM subsystem concurrently (i.e., may be a shared namespace). Bits 7:1 are reserved. Bit 0, if cleared to '0', indicates the namespace is a private namespace and is able to be attached to only one controller at a time.
M andatory	No	No
Notes		Returned in byte 30 of the Namespace Features (NSFEAT) of the of the Identify Namespace Data Structure (Reference NVMe Base Specification section 5.15.2.1 and figure 247).

6.5.2.19 NVMeNamespaceProperties.NamespaceFeatures.SupportsThinProvisioning

The mapping for `NVMeNamespaceProperties.NamespaceFeatures.SupportsThinProvisioning` is summarized in Table 153.

Table 153:
NVMeNamespaceProperties.NamespaceFeatures. SupportsThinProvisioning mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeNamespaceProperties.NamespaceFeatures.SupportsThinProvisioning	**NVM Spec Property / Field:** THINP **NVM Spec: Section:Figure** NVMe 1.4a: Section 5.15.2.1 (Identify Namespace), Figure 247
Type	Boolean	**NVM Spec Property Type:** Single bit (bool) **Additional NVM Spec Identifying Information:** Bit 0 of Byte 24
Des cription	This property indicates whether or not the NVMe Namespace supports thin provisioning.	Indicates that the namespace supports thin provisioning
LongDes cription	This property shall indicate whether or not the NVMe Namespace supports thin provisioning. Specifically, the namespace capacity reported may be less than the namespace size.	if set to '1' indicates that the namespace supports thin provisioning. If cleared to '0' indicates that thin provisioning is not supported. Refer to NVMe Base Specification section 6.1.7 for details on the usage of this field.
M andatory	Yes	Yes
Notes		Returned in byte 24, bit 0 of the Namespace Features (NSFEAT) of the of the Identify Namespace Data Structure (Reference NVMe Base Specification section 5.15.2.1 and figure 247).

6.5.2.20 NVMeNamespaceProperties.NamespaceFeatures.SupportsDeallocatedOrUnwrittenLBError

The mapping for `NVMeNamespaceProperties.NamespaceFeatures.SupportsDeallocatedOrUnwrittenLBError` is summarized in Table 154.

Table 154: NVMeNamespaceProperties.NamespaceFeatures. SupportsDeallocatedOrUnwrittenLBError mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeNamespaceProperties.NamespaceFeatures.SupportsDeallocatedOrUnwrittenLBError	**NVM Spec Property / Field:** DAE **NVM Spec: Section:Figure** NVMe 1.4a: Section 5.15.2.1 (Identify Namespace), Figure 247
Type	Boolean	**NVM Spec Property Type:** Single bit (bool) **Additional NVM Spec Identifying Information:** Bit 2 of Byte 24
Des cription	This property indicates that the controller supports deallocated or unwritten logical block error for this namespace.	Indicates that the controller supports the Deallocated or Unwritten Logical Block error for this namespace.
LongDes cription	This property shall indicate that the controller supports deallocated or unwritten logical block error for this namespace.	If set to ‘1’ indicates that the controller supports the Deallocated or Unwritten Logical Block error for this namespace. If cleared to ‘0’, then the controller does not support the Deallocated or Unwritten Logical Block error for this namespace. Refer to NVMe Base Specification section 6.7.1.1
M andatory	Yes	Yes

	Redfish/Swordfish	NVMe / NVMe-oF
Notes		Returned in byte 24, bit 2 of the Namespace Features (NSFEAT) of the of the Identify Namespace Data Structure (Reference NVMe Base Specification section 5.15.2.1 and figure 247.

6.5.2.21 NVMeNamespaceProperties.NamespaceFeatures.SupportsNGUIDReuse

The mapping for `NVMeNamespaceProperties.NamespaceFeatures.SupportsNGUIDReuse` is summarized in Table 155.

Table 155: NVMeNamespaceProperties.NamespaceFeatures.SupportsNGUIDReuse mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeNa mespaceProperties.Namespace Features.SupportsNGUIDReuse	**NVM Spec Property / Field:** UIDREUSE **NVM Spec: Section:Figure** NVMe 1.4a: Section 5.15.2.1 (Identify Namespace), Figure 247
Type	Boolean	**NVM Spec Property Type:** Single bit (bool) **Additional NVM Spec Identifying Information:** Bit 3 of Byte 24
Des cription	This property indicates that the namespace supports the use of an NGUID (namespace globally unique identifier) value.	Indicates if the value in the NGUID field and the value in the EUI64 field for this namespace may be reused by the controller for a new namespace created after this namespace is deleted.

	Redfish/Swordfish	NVMe / NVMe-oF
LongDescription	This property shall indicate that the namespace supports the use of an NGUID (namespace globally unique identifier) value.	If set to '1' indicates that the value in the NGUID field for this namespace, if non-zero, is never reused by the controller and that the value in the EUI64 field for this namespace, if non-zero, is never reused by the controller. If cleared to '0', then the NGUID value may be reused and the EUI64 value may be reused by the controller for a n3333ew namespace created after this namespace is deleted. This bit shall be cleared to '0' if both NGUID and EUI64 fields are cleared to 0h. Refer to NVMe Base Specification, section 7.11.
M andatory	Yes	Yes
Notes		Returned in byte 24, bit 3 of the Namespace Features (NSFEAT) of the of the Identify Namespace Data Structure (Reference NVMe Base Specification section 5.15.2.1 and figure 247).

6.5.2.22 NVMeNamespaceProperties.NamespaceFeatures.SupportsAtomicTransactionSize

The mapping for `NVMeNamespaceProperties.NamespaceFeatures.SupportsAtomicTransactionSize` is summarized in Table 156.

Table 156:
NVMeNamespaceProperties.NamespaceFeatures. SupportsAtomicTransactionSize mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeNamespaceProperties.NamespaceFeatures.SupportsAtomicTransactionSize	**NVM Spec Property / Field:** OPTPERF **NVM Spec: Section:Figure** NVMe 1.4a: Section 5.15.2.1 (Identify Namespace), Figure 247
Type	Boolean	**NVM Spec Property Type:** Single bit (bool) **Additional NVM Spec Identifying Information:** Bit 4 of Byte 24
Description	Indicates whether or not the NVM fields for Namespace preferred write granularity (NPWG), write alignment (NPWA), deallocate granularity (NPDG), deallocate alignment (NPDA) and optimimal write size (NOWS) are defined for this namespace and should be used by the host for I/O optimization.	Indicates support for the fields NPWG, NPWA, NPDG, NPDA, and NOWS for this namespace; and optimal Write Size field in NVM Sets Attributes Entry

	Redfish/Swordfish	NVMe / NVMe-oF
LongDescription	This property shall indicate whether or not the NVM fields for Namespace preferred write granularity (NPWG), write alignment (NPWA), deallocate granularity (NPDG), deallocate alignment (NPDA) and optimimal write size (NOWS) are defined for this namespace and should be used by the host for I/O optimization.	If set to ‘1’ indicates that the fields NAWUN, NAWUPF, and NACWU are defined for this namespace and should be used by the host for this namespace instead of the AWUN, AWUPF, and ACWU fields in the Identify Controller data structure. If cleared to ‘0’, then the controller does not support the fields NAWUN, NAWUPF, and NACWU for this namespace. In this case, the host should use the AWUN, AWUPF, and ACWU fields defined in the Identify Controller data structure in Figure 247. Refer to NVMe Base Specification section 6.4.
M andatory	Yes	Yes
Notes		Returned in byte 24, bit 4 of the Namespace Features (NSFEAT) of the of the Identify Namespace Data Structure (Reference NVMe Base Specification section 5.15.2.1 and figure 247).

6.5.2.23 NVMeNamespaceProperties.NamespaceFeatures.SupportsIOPerformanceHints

The mapping for `NVMeNamespaceProperties.NamespaceFeatures.SupportsIOPerformanceHints` is summarized in Table 157.

Table 157:
NVMeNamespaceProperties.NamespaceFeatures. SupportsIOPerformanceHints mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeNamespaceP roperties.NamespaceFeatures .SupportsIOPerformanceHints	**NVM Spec Property / Field:** NSABP **NVM Spec: Section:Figure** NVMe 1.4a: Section 5.15.2.1 (Identify Namespace), Figure 247
Type	Boolean	**NVM Spec Property Type:** Single bit (bool) **Additional NVM Spec Identifying Information:** Bit 1 of Byte 24
Des cription	Indicates that the Namepsace Atomic Write Unit Normal (NAWUN), Namespace Atomic Write Unit Power Fail (NAWUPF), and Namespace Atomic Compare and Write Unit (NACWU) fields are defined for this namespace and should be used by the host for this namespace instead of the controller-level properties AWUN, AWUPF, and ACWU.	indicates whether or not the fields NAWUN, NAWUPF, and NACWU are defined for this namespace and should be used by the host for this namespace instead of the AWUN, AWUPF, and ACWU fields in the Identify Controller data structure.

	Redfish/Swordfish	NVMe / NVMe-oF
LongDescription	This property shall indicate that the Namepsace Atomic Write Unit Normal (NAWUN), Namespace Atomic Write Unit Power Fail (NAWUPF), and Namespace Atomic Compare and Write Unit (NACWU) fields are defined for this namespace and should be used by the host for this namespace instead of the controller-level properties AWUN, AWUPF, and ACWU.	If set to '1' indicates that the fields NAWUN, NAWUPF, and NACWU are defined for this namespace and should be used by the host for this namespace instead of the AWUN, AWUPF, and ACWU fields in the Identify Controller data structure. If cleared to '0', then the controller does not support the fields NAWUN, NAWUPF, and NACWU for this namespace. In this case, the host should use the AWUN, AWUPF, and ACWU fields defined in the Identify Controller data structure in NVMe Base Specification Figure 247. Refer to NVMe Base Specification section 6.4.
M andatory	Yes	Yes
Notes		Returned in byte 24, bit 1 of the Namespace Features (NSFEAT) of the of the Identify Namespace Data Structure (Reference NVMe Base Specification section 5.15.2.1 and figure 247).

6.5.2.24 NVMeNamespaceProperties.NumberLBAFormats The mapping for `NVMeNamespaceProperties.NumberLBAFormats` is summarized in Table 158.

Table 158: 158: NVMeNamespaceProperties.NumberLBAFormats mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeNamespace Properties.NumberLBAFormats	**NVM Spec Property / Field:** Number of LBA Formats (NLBAF) **NVM Spec: Section:Figure** NVMe 1.4a: Section 5.15.2.1 (Identify Namespace), Figure 247
Type	Int64	Type:** Int64 **Additional NVM Spec Identifying Information:** ByteOffset: 25
Des cription	The number of LBA data size and metadata size combinations supported by this namespace. The value of this property is between 0 and 16.	The number of supported LBA data size and metadata size combinations supported by the namespace.
LongDes cription	This property shall contain the number of LBA data size and metadata size combinations supported by this namespace. The value of this property is between 0 and 16. LBA formats with an index set beyond this value will not be supported.	This property shall contain the number of LBA formats allocated in order starting with 0 and packed sequentially. This is a 0's based value. The maximum number of LBA formats that may be indicated as supported is 16. The supported LBA formats are indicated in bytes 128 to 191 in this data structure. The LBA Format fields with an index beyond the value set in this field are invalid and not supported. LBA Formats that are valid, but not currently available may be indicated by setting the LBA Data Size for that LBA Format to 0h.
M andatory	Yes	Yes

	Redfish/Swordfish	NVMe / NVMe-oF
Notes		Returned in byte 25 (Number of LBA Formats) of the Identify Namespace Data Structure (Reference NVMe Base Specification section 5.15.2.1 & figure 247.

6.5.2.25 NVMeNamespaceProperties.FormattedLBASize The mapping for `NVMeNamespaceProperties.FormattedLBASize` is summarized in Table 159.

Table 159: NVMeNamespaceProperties.FormattedLBASize mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeNamespace Properties.FormattedLBASize	**NVM Spec Property / Field:** Formatted LBA Size (FLBAS) **NVM Spec: Section:Figure** NVMe 1.4a: Section 5.15.2.1 (Identify Namespace), Figure 247
Type	Int64	Type:** Int64 **Additional NVM Spec Identifying Information:** ByteOffset: 26
Des cription	The LBA data size and metadata size combination that the namespace has been formatted with.	The LBA data size and metadata size combination that the namespace has been formatted with.
LongDes cription	This property shall contain the LBA data size and metadata size combination that the namespace has been formatted with. This is a 4-bit data structure.	This field indicates the LBA data size & metadata size combination that the namespace has been formatted with (refer to section 5.23). Bits 3:0 indicates one of the 16 supported LBA Formats indicated in this data structure.

	Redfish/Swordfish	NVMe / NVMe-oF
M andatory	Yes	Yes
Notes		Returned in byte 26 (Formatted LBA Size), bits 3:0 of the Identify Namespace Data Structure (Reference NVMe Base Specification section 5.15.2.1 and figure 247.

6.5.2.26 NVMeNamespaceProperties.MetadataTransferredAtEndOfDataLBA

The mapping for `NVMeNamespaceProperties.MetadataTransferredAtEndOfDataLBA` is summarized in Table 160.

Table 160: NVMeNamespaceProperties.MetadataTransferredAtEndOfDataLBA mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVM eNamespaceProperties.Metada taTransferredAtEndOfDataLBA	**NVM Spec Property / Field:** Metadata transferred at end of LBA **NVM Spec: Section:Figure** NVMe 1.4a: Section 5.15.2.1 (Identify Namespace), Figure 247
Type	Boolean	Type:** Bit (bool) **Additional NVM Spec Identifying Information:** Bit 4 of Byte 26
Des cription	This property indicates whether or not the metadata is transferred at the end of the LBA creating an extended data LBA.	This property indicates whether or not the metadata is transferred at the end of the data LBA.

	Redfish/Swordfish	NVMe / NVMe-oF
LongDes cription	This property shall indicate whether or not the metadata is transferred at the end of the LBA creating an extended data LBA.	If set to ‘1’ indicates that metadata is transferred at the end of the data LBA, creating an extended data LBA. Bit 4 if cleared to ‘0’ indicates that all of the metadata for a command is transferred as a separate contiguous buffer of data.
M andatory	Yes	Yes
Notes		Returned in byte 26 (Formatted LBA Size), bit 4 of the Identify Namespace Data Structure (Reference NVMe Base Specification section 5.15.2.1 and figure 247. Bit 4 is not applicable when there is no metadata.

6.5.2.27 NVMeNamespaceProperties.NVMeVersion The mapping for `NVMeNamespaceProperties.NVMeVersion` is summarized in Table 161.

Table 161: NVMeNamespaceProperties.NVMeVersion mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeName spaceProperties.NVMeVersion	Version (VER)
Type	String	Int64
Des cription	The version of the NVMe Base Specification supported.	This property shall contain the version of the NVMe Base Specification supported.

	Redfish/Swordfish	NVMe / NVMe-oF
LongDes cription		Indicates the major, minor, and tertiary version of the NVM Express base specification that the controller implementation supports. Valid versions of the specification are: 1.0, 1.1, 1.2, 1.2.1, 1.3, 1.4, and 2.0.
M andatory	Yes	Yes
Notes		Returned in bytes 83:80 of the Identify Controller data structure - CNS 01h (reference NVMe Base Specification section 5.15.2.2 and figure 249.

6.5.2.28 OptimumIOSizeBytes

The mapping for `OptimumIOSizeBytes` is summarized in Table 162.

Table 162: OptimumIOSizeBytes mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	OptimumIOSizeBytes	**NVM Spec Property / Field:** Namespace Optimal Write Size (NOWS) **NVM Spec: Section:Figure** Figure 247: Bytes 73:72
Type	Int64	
Des cription	The size in bytes of this Volume's optimum IO size.	N/A

	Redfish/Swordfish	NVMe / NVMe-oF
LongDes cription	This property shall contain the optimum IO size to use when performing IO on this volume. For logical disks, this is the stripe size. For physical disks, this describes the physical sector size.	This field indicates the size in logical blocks for optimal write performance for this namespace. This is a 0's based value.
M andatory	Recommended for implementations with Endurance Groups and NVM Sets.	
Notes		Convert from blocks to bytes.

6.5.2.29 ProvisioningPolicy The mapping for `ProvisioningPolicy` is summarized in Table 163.

Table 163: OptimumIOSizeBytes mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	ProvisioningPolicy	**NVM Spec Property / Field:** Identify Namespace / THINP **NVM Spec: Section:Figure** 247: Byte 24, Bit 0
Type	enum (DataStorageLoSCapabilities.ProvisioningPolicy)	
Des cription	This property specifies the volume's storage allocation, or provisioning policy.	N/A
LongDes cription	This property shall specify the volume's supported storage allocation policy.	

	Redfish/Swordfish	NVMe / NVMe-oF
M andatory	Recommended for implementations that support thin provisioning.	Figure 247: Byte 24, Bit 0 indicates thin provisioning support for the namespace
Notes	possible values: Fixed / Thin	

6.5.2.30 Status.State

The mapping for `Status.state` is summarized in Table 164.

Table 164: Status.State mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Status.State	Enable (EN)
Type	Resource.State (enum)	Boolean
Des cription	The known state of the resource, such as, enabled.	Indicates if the controller is in 'enabled' state.
LongDes cription	This property shall indicate whether and why this component is available. Enabled indicates the resource is available. Disabled indicates the resource has been intentionally made unavailable but can be enabled. Offline indicates the resource is unavailable intentionally and requires action to make it available. InTest indicates that the component is undergoing testing. Starting indicates that the resource is becoming available. Absent indicates the resource is physically unavailable.	When set to '1', then the controller shall process commands based on Submission Queue Tail doorbell writes. When cleared to '0', then the controller shall not process commands nor post completion queue entries to Completion Queues. When this bit transitions from '1' to '0', the controller is reset (i.e., a Controller Reset). That reset deletes all I/O Submission Queues and I/O Completion Queues, resets the Admin Submission Queue and Completion Queue, and brings the hardware to an idle state.
M andatory	Yes	Yes

	Redfish/Swordfish	NVMe / NVMe-oF
Notes	Possible values: Enabled / Disabled / StandbyOffline / StandbySare / InTest / Starting / Absent / UnavailableOffline / Deferring / Quiesced / Updating / Qualified	Reference Controller Configuration (CC), offset 14h, bit 00 of the NVMe Base Specification (figure 78)

6.5.2.31 Status.Health The mapping for `Status.Health` is summarized in Table 165.

Table 165: Status.Health mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Status.Health	Controller Fatal Status (CSTS.CFS)
Type	Resource.Health	Bit
Des cription	The health state of this resource in the absence of its dependent resources.	Indicates if the controller is able to communicate with host software via completion queue entries in the Admin Completion Queue or I/O Completion Queues.

	Redfish/Swordfish	NVMe / NVMe-oF
LongDescription	This property shall represent the health state of the resource without considering its dependent resources. The values shall conform to those defined in the Redfish Specification.	If the controller has a serious error condition and is unable to communicate with host software via completion queue entries in the Admin Completion Queue or I/O Completion Queues, then the controller may set the Controller Fatal Status (CSTS.CFS) bit to '1' (refer to section 3.1.6). This indicates to host software that a serious error condition has occurred. When this condition occurs, host software should attempt to reset and then re-initialize the controller. The Controller Fatal Status condition is not indicated with an interrupt. If host software experiences timeout conditions and/or repeated errors, then host software should consult the Controller Fatal Status (CSTS.CFS) bit to determine if a more serious error has occurred.
M andatory	Yes	Yes
Notes	Possible Values: OK / Warning / Critical	Reference Figure 222 (NVM Subsystem Hardware Error Event Codes) bit 09h of the NVMe Base Specification.

6.5.2.32 Status.HealthRollup The mapping for `Status.HealthRollup` is summarized in Table 166.

Table 166: Status.HealthRollup mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Status.HealthRollup	N/A
Type	Resource.Health	N/A
Des cription	The overall health state from the view of this resource.	N/A
LongDes cription	This property shall represent the health state of the resource and its dependent resources. The values shall conform to those defined in the Redfish Specification.	N/A
M andatory	No	N/A
Notes	Do not use for Namespace. There are no dependent resources.	Do not implement.

6.5.2.33 StorageGroups The mapping for `StorageGroups` is summarized in Table 167.

Table 167: StorageGroups mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	StorageGroups	**NVM Spec Property / Field:** N/A **NVM Spec: Section:Figure** N/A
Type	StorageGroupCollection.StorageGroupCollection	N/A
Description	An array of references to Storage Groups that includes this volume.	N/A
LongDescription	The value of this property shall contain references to all storage groups that include this volume.	N/A
M andatory	Do Not Implement	
Notes		

6.5.2.34 WriteCachePolicy The mapping for `WriteCachePolicy` is summarized in Table 168.

Table 168: WriteCachePolicy mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	WriteCachePolicy	**NVM Spec Property / Field:** N/A **NVM Spec: Section:Figure** N/A
Type	StorageGroupCollection.StorageGroupCollection	N/A
Description	Indicates the write cache policy setting for the Volume	N/A
LongDescription	This property shall contain a boolean indicator of the write cache policy for the Volume.	N/A
M andatory	Recommended.	
Notes	Possible Values: WriteThrough, ProtectedWriteBack, UnprotectedWriteBack, Off. Set to “Off” when write cache disabled; set to other values when enabled.	

6.6 Endurance Group

The Redfish/Swordfish Storage Pool schema is used to represent an NVM Endurance Group.

6.6.1 Mockup

The following mockup shows a sample representation of the Storage Pool schema used to represent an NVM Endurance Group.

```
{
  "@Redfish.Copyright": "Copyright 2015-2020 SNIA. All rights reserved.",
  "@odata.id":
  ↪  "/redfish/v1/Storage/FabricAttachArray/StoragePools/EnduranceGroup1",
  "@odata.type": "#StoragePool.v1_4_0.StoragePool",
  "Id": "1",
  "Name": "Endurance Group 1",
  "Description": "Single Endurance Group",
  "Status": {
    "State": "Enabled",
    "Health": "OK"
  },

  "NVMeEnduranceGroupProperties": {
    "EndGrpLifetime": {
      "PercentUsed": 0,
      "EnduranceEstimate": 0,
      "DataUnitsRead": 0,
      "DataUnitsWritten": 0,
      "MediaUnitsWritten": 0,
      "HostReadCommandCount": 0,
      "HostWriteCommandCount": 0,
      "MediaAndDataIntegrityErrorCount": 0,
      "ErrorInformationLogEntryCount": 0
    }
  },

    "Capacity": {
      "Data": {
        "AllocatedBytes": 10995116277760,
        "ConsumedBytes": 10995116277760
      }
    },
  "CapacitySources": [{
    "@odata.id": "/redfish/v1/Storage/FabricAttachArray/StoragePools/EnduranceGroup1/CapacitySources/Source1"
}]
}
```

6.6.2 Property Mapping

6.6.2.1 AllocatedPools The mapping for `AllocatedPools` is summarized in Table 169.

Table 169: AllocatedPools mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	AllocatedPools	NVM Set List
Type	StoragePoolCollection.StoragePoolCollection	
Description	A reference to the collection of storage pools allocated from this storage pool.	A list of NVM sets
LongDescription	The value of this property shall contain a reference to the collection of storage pools allocated from this storage pool.	An ordered list by NVM Set Identifier, starting with the first NVM Set Identifier supported by the NVM subsystem that is equal to or greater than the NVM Set Identifier indicated in CDW11.NVMSETID. The NVM Set List describes the attributes for each NVM Set in the list based on the NVM Set Attributes Entry
M andatory	Required.	Optional (if NVM sets are supported)
Notes	Contains a pointer to the NVM Set allocated from this Endurance Group.	Reference Figure 250

6.6.2.2 Capacity.Data.AllocatedBytes

The mapping for `Capacity.Data.AllocatedBytes` is summarized in Table 170.

Table 170: Capacity.Data.AllocatedBytes mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	C apacity.Data.AllocatedBytes	The Total Endurance Group Capacity. Just the "Total Endurance Group Capacity" in the Endurance Group Log. See 5.14.1.9 + TP 4009
Type	Int64	
Des cription	The number of bytes currently allocated by the storage system in this data store for this data type.	
LongDes cription	The value shall be the number of bytes currently allocated by the storage system in this data store for this data type.	
M andatory	Required for NVMe Drives.	
Notes		Note: This is not in 1.4a because TP 4009 was not integrated.

6.6.2.3 Capacity.Data.ConsumedBytes

The mapping for `Capacity.Data.ConsumedBytes` is summarized in Table 171.

Table 171: Capacity.Data.ConsumedBytes mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Capacity.Data.ConsumedBytes	The Consumed (allocated to NVM Sets) Endurance Group Capacity. Just the “Total Endurance Group Capacity” (TEGCAP) minus the Uallocated Endurance Group Capacity (UEGCAP) in the Endurance Group Log. See 5.14.1.9 + TP 4009
Type	Int64	
Des cription	The number of bytes consumed in this data store for this data type.	
LongDes cription	The value shall be the number of logical bytes currently consumed in this data store for this data type.	
M andatory	Required for NVMe Drives.	
Notes		Note: This is not in 1.4a because TP 4009 was not integrated.

6.6.2.4 CapacitySources The mapping for `CapacitySources` is summarized in Table 172.

Table 172: CapacitySources mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	CapacitySources	NVM Set List
Type	Collecti on(Capacity.CapacitySource)	
Des cription	An array of space allocations to this volume.	A list of NVM Sets
LongDes cription	Fully or partially consumed storage from a source resource. Each entry provides capacity allocation information from a named source resource.	An ordered list by NVM Set Identifier, starting with the first NVM Set Identifier supported by the NVM subsystem that is equal to or greater than the NVM Set Identifier indicated in CDW11.NVMSETID. The NVM Set List describes the attributes for each NVM Set in the list based on the NVM Set Attributes Entry
M andatory	Required for NVMe Drives.	Optional (if NVM sets are supported)
Notes	Contains the information about the providing capacity (e.g, EnduranceGroup) for this namespace.	Reference Figure 250

6.6.2.5 CapacitySources@odata.count The mapping for `CapacitySources@odata.count` is summarized in Table 173.

Table 173: CapacitySources@odata.count mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	C apacitySources@odata.count	
Type	(odata property)int64	
Des cription	Count of the number of items in the CapacitySources array.	
LongDes cription		
M andatory	Required	
Notes		This should be the same as the number of NVM Sets in the endurance group.

6.6.2.6 Description The mapping for `Description` is summarized in Table 174.

Table 174: Description mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Description	N/A
Type	String	N/A
Des cription	The description of this resource.	
LongDes cription	This object represents the description of this resource. The resource values shall comply with the Redfish Specification-described requirements.	
M andatory		
Notes	In Redfish, Description is a read-only field.	Return the common description: “An Endurance Group consists of zero or more NVM Sets. Endurance Groups divide the media into distinct wear-leveling domains.”

6.6.2.7 Links.OwningStorageResource The mapping for `Links.OwningStorageResource` is summarized in Table 175.

Table 175: Links.OwningStorageResource mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Links.OwningStorageResource	N/A
Type	Storage.Storage	N/A
Des cription	A pointer to the Storage resource that owns or contains this StoragePool.	N/A
LongDes cription	This shall be a pointer to the Storage resource that owns or contains this StoragePool.	N/A
M andatory	Required.	N/A
Notes	Contains a pointer to the NVM Subsystem that contains this Endurance Group.	A pointer to the owning subsystem - this is a Swordfish construct which points back to the subsystem

6.6.2.8 Name

The mapping for `Name` is summarized in Table 176.

Table 176: Name mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Name	Endurance Group ID NVMe 1.4a: Section 5.14,1,9, Get Log Page - Endurance Group Log
Type	String	16-bit value
Des cription	The name of the resource or array member.	
LongDes cription	This object represents the name of this resource or array member. The resource values shall comply with the Redfish Specification-described requirements. This string value shall be of the 'Name' reserved word format.	
M andatory		
Notes	In Redfish, Name is a read-only field.	The Endurance Group Identifier is specified in the Log Specific Identifier field in Command Dword 11 of the Get Log Page command. Map the Endurance Group ID field to a string with the format: "0xABCD"

6.6.2.9 NVMeProperties.NVMePoolType The mapping for `NVMeProperties.NVMePoolType` is summarized in Table 177.

Table 177: NVMeProperties.NVMePoolType

	Redfish/Swordfish	NVMe / NVMe-oF
Property	StoragePool.NVMePoolType	
Type	Enum	
Des cription	Indicates whether the StoragePool is used as an EnduranceGroup or an NVMSet.	
LongDes cription	This property shall indicate whether the StoragePool is used as an EnduranceGroup or an NVMSet.	
M andatory	Required	
Notes	Set as “EnduranceGroup”	

6.6.2.10 NVMeEnduranceGroupProperties.PredictedMediaLifeLeftPercent

The mapping for `NVMeEnduranceGroupProperties.PredictedMediaLifeLeftPercent` is summarized in Table 178.

Table 178: NVMeEnduranceGroupProperties.PredictedMediaLifeLeftPercent mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMe EnduranceGroupProperties.PredictedMediaLifeLeftPercent	“Percentage Used” in the Endurance Group Log. See 5.14.1.9
Type	Decimal %	
Des cription	The percentage of reads and writes that are predicted to be available for the media.	
LongDes cription	This property shall contain an indicator of the percentage of life remaining in the drive’s media.	
M andatory	Required	
Notes		Inverse of “Percentage Used” in the Endurance Group Log. See 5.14.1.9 . Subtract the percentage used from 100% to report this value.

6.6.2.11 NVMeEnduranceGroupProperties.EndGrpLifetime.PercentUsed The mapping for `NVMeEnduranceGroupProperties.EndGrpLifetime.PercentUsed` is summarized in Table 179.

Table 179: NVMeEnduranceGroupProperties.EndGrpLifetime.PercentUsed mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	N VMeEnduranceGroupProperties .EndGrpLifetime.PercentUsed	“Percentage Used” in the Endurance Group Log. See 5.14.1.9
Type	Int64	
Des cription	A vendor-specific estimate of the percent life used for the endurance group based on the actual usage and the manufacturer prediction of NVM life.	
LongDes cription	This property shall contain A vendor-specific estimate of the percent life used for the endurance group based on the actual usage and the manufacturer prediction of NVM life. A value of 100 indicates that the estimated endurance of the NVM in the Endurance Group has been consumed, but may not indicate an NVM failure. According to the NVMe and JEDEC specs, the value is allowed to exceed 100. Percentages greater than 254 shall be represented as 255.	
M andatory	Required	
Notes		

6.6.2.12 NVMeEnduranceGroupProperties.EndGrpLifetime.EnduranceEstimate

The mapping for `NVMeEnduranceGroupProperties.EndGrpLifetime.EnduranceEstimate` is summarized in Table 180.

Table 180: NVMeEnduranceGroupProperties.EndGrpLifetime.EnduranceEstimate mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeEnd uranceGroupProperties.EndGr pLifetime.EnduranceEstimate	“Endurance Estimate” in the Endurance Group Log. See 5.14.1.9
Type	Int64	
Des cription	This property contains an estimate of the total number of data bytes that may be written to the Endurance Group over the lifetime of the Endurance Group assuming a write amplication of 1.	
LongDes cription	This property shall contain an estimate of the total number of data bytes that may be written to the Endurance Group over the lifetime of the Endurance Group assuming a write amplication of 1. The value is reported in billions, where a value of 1 corresponds to 1 billion bytes written, and is rounded up. A value of zero indicates endurance estimates are unsupported.	
M andatory	Required	
Notes		

6.6.2.13 NVMeEnduranceGroupProperties.EndGrpLifetime.DataUnitsRead

The mapping for `NVMeEnduranceGroupProperties.EndGrpLifetime.DataUnitsRead` is summarized in Table 181.

Table 181: NVMeEnduranceGroupProperties.EndGrpLifetime.DataUnitsRead mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVM eEnduranceGroupProperties.E ndGrpLifetime.DataUnitsRead	
Type	Int64	
Des cription	The property contains the total number of data units read from this endurance group.	
LongDes cription	The property shall contain the total number of data units read from this endurance group. This value does not include controller reads due to internal operations such as garbage collection. The value is reported in billions, where a value of 1 corresponds to 1 billion bytes written, and is rounded up. A value of zero indicates the property is unsupported.	
M andatory	Required	
Notes		

6.6.2.14 NVMeEnduranceGroupProperties.EndGrpLifetime.DataUnitsWritten

The mapping for `NVMeEnduranceGroupProperties.EndGrpLifetime.DataUnitsWritten` is summarized in Table 182.

Table 182: NVMeEnduranceGroupProperties.EndGrpLifetime.DataUnitsWritten mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeEn duranceGroupProperties.EndG rpLifetime.DataUnitsWritten	“Data Units Written” in the Endurance Group Log. See 5.14.1.9
Type	Int64	
Des cription	The property contains the total number of data units written from this endurance group.	
LongDes cription	The property shall contain the total number of data units written from this endurance group. This value does not include controller writes due to internal operations such as garbage collection. The value is reported in billions, where a value of 1 corresponds to 1 billion bytes written, and is rounded up. A value of zero indicates the property is unsupported.	
M andatory	Required	
Notes		

6.6.2.15 NVMeEnduranceGroupProperties.EndGrpLifetime.MediaUnitsWritten

The mapping for `NVMeEnduranceGroupProperties.EndGrpLifetime.MediaUnitsWritten` is summarized in Table 183.

Table 183: NVMeEnduranceGroupProperties.EndGrpLifetime.MediaUnitsWritten mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeEnd uranceGroupProperties.EndGr pLifetime.MediaUnitsWritten	“Media Units Written” in the Endurance Group Log. See 5.14.1.9
Type	Int64	
Des cription	The property contains the total number of data units written from this endurance group.	
LongDes cription	The property shall contain the total number of data units written from this endurance group. This value includes host and controller writes due to internal operations such as garbage collection. The value is reported in billions, where a value of 1 corresponds to 1 billion bytes written, and is rounded up. A value of zero indicates the property is unsupported.	
M andatory	Required	
Notes		

6.6.2.16 NVMeEnduranceGroupProperties.EndGrpLifetime.HostReadCommandCount

The mapping for `NVMeEnduranceGroupProperties.EndGrpLifetime.HostReadCommandCount` is summarized in Table 184.

Table 184: NVMeEnduranceGroupProperties.EndGrpLifetime.HostReadCommandCount mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeEnduranceGroupProperties.EndGrpLifetime.HostReadCommandCount	“Host Read Commands” in the Endurance Group Log. See 5.14.1.9
Type	Int64	
Des cription	This property contains the number of read commands completed by all controllers in the NVM subsystem for the Endurance Group.	
LongDes cription	This property shall contain the number of read commands completed by all controllers in the NVM subsystem for the Endurance Group. For the NVM command set, the is the number of compare commands and read commands.	
M andatory	Required	
Notes		

6.6.2.17 NVMeEnduranceGroupProperties.EndGrpLifetime.HostWriteCommandCount

The mapping for `NVMeEnduranceGroupProperties.EndGrpLifetime.HostWriteCommandCount` is summarized in Table 185.

Table 185: NVMeEnduranceGroupProperties.EndGrpLifetime. HostWriteCommandCount mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeEnduranceGroupProperties.EndGrpLifetime.HostWriteCommandCount	“Host Write Commands” in the Endurance Group Log. See 5.14.1.9
Type	Int64	
Des cription	This property contains the number of write commands completed by all controllers in the NVM subsystem for the Endurance Group.	
LongDes cription	This property shall contain the number of write commands completed by all controllers in the NVM subsystem for the Endurance Group. For the NVM command set, the is the number of compare commands and write commands.	
M andatory	Required	
Notes		

6.6.2.18 NVMeEnduranceGroupProperties.EndGrpLifetime.MediaAndDataIntegrityErrorCount

The mapping for `NVMeEnduranceGroupProperties.EndGrpLifetime.MediaAndDataIntegrityErrorCount` is summarized in Table 186.

Table 186: NVMeEnduranceGroupProperties.EndGrpLifetime. MediaAndDataIntegrityErrorCount mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeEnduranceGroupProperties.EndGrpLifetime.MediaAndDataIntegrityErrorCount	"Media and Data Integrity Errors" in the Endurance Group Log. See 5.14.1.9
Type	Int64	
Description	This property contains the number of occurences where the controller detected an unrecovered data integrity error for the Endurance Group.	
LongDescription	This property shall contain the number of occurences where the controller detected an unrecovered data integrity error for the Endurance Group. Errors such as uncorrectable ECC, CRC checksum failure, or LBA tag mismatch are included in this field.	
M andatory	Required	
Notes		

6.6.2.19 NVMeEnduranceGroupProperties.EndGrpLifetime.ErrorInformationLogEntryCount

The mapping for `NVMeEnduranceGroupProperties.EndGrpLifetime.ErrorInformationLogEntryCount` is summarized in Table 187.

Table 187: NVMeEnduranceGroupProperties.EndGrpLifetime.ErrorInformationLogEntryCount mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeEnduranceGroupProperties.EndGrpLifetime.ErrorInformationLogEntryCount	“Number of Error Information Log Entries” in the Endurance Group Log. See 5.14.1.9
Type	Int64	
Des cription	This property contains the number of error information log entries over the life of the controller for the endurance group.	
LongDes cription	This property shall contain the number of error information log entries over the life of the controller for the endurance group.	
M andatory	Required	
Notes		

6.6.2.20 NVMeSetProperties.SetIdentifier The mapping for `NVMeSetProperties.SetIdentifier` is summarized in Table 188.

Table 188: NVMeSetProperties.SetIdentifier

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMe SetProperties.SetIdentifier	**NVM Spec Property / Field:** NVMSETID **NVM Spec: Section:Figure** NVMe 1.4a: Section 5.15.2.5, Figure 253: NVM Set Attributes Entry
Type	String	**NVM Spec Property Type:** 16-bit value **Additional NVM Spec Identifying Information:** ByteOffset: 01:00
Des cription	A 16-bit hex value that contains the NVMe Set identifier.	
LongDes cription	This property shall contain a 16-bit hex value that contains the NVMe Set identifier. The NVM Set identifier is unique within a subsystem. Reserved values include 0.	
M andatory	Do Not Implement.	
Notes		Do not implement NVMeSetProperties as part of an EnduranceGroup.

6.6.2.21 NVMeSetProperties.OptimalWriteSizeBytes The mapping for `NVMeSetProperties.OptimalWriteSizeBytes` is summarized in Table 189.

Table 189: NVMeSetProperties.OptimalWriteSizeBytes mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeSetPrope rties.OptimalWriteSizeBytes	**NVM Spec Property / Field:** OptimalWriteSize **NVM Spec: Section:Figure** NVMe 1.4a: Section 5.15.2.5, Figure 253: NVM Set Attributes Entry
Type	Int64	**NVM Spec Property Type:** Bytes **Additional NVM Spec Identifying Information:** ByteOffset: 15:12
Des cription	This property contains the Optimal Write Size in Bytes for this NVMe Set.	
LongDes cription	This property shall contain the Optimal Write Size in Bytes for this NVMe Set.	
M andatory	Do Not Implement.	
Notes		Do not implement NVMeSetProperties as part of an EnduranceGroup.

6.6.2.22 NVMeSetProperties.EnduranceGroupIdentifier The mapping for `NVMeSetProperties.EnduranceGroupIdentifier` is summarized in Table 190.

Table 190: NVMeSetProperties.EnduranceGroupIdentifier mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeSetProperti es.EnduranceGroupIdentifier	**NVM Spec Property / Field:** EnduranceGroupIdentifier **NVM Spec: Section:Figure** NVMe 1.4a: Section 5.15.2.5, Figure 253: NVM Set Attributes Entry
Type	String	**NVM Spec Property Type:** 2 bytes **Additional NVM Spec Identifying Information:** ByteOffset: 03:02
Des cription	A 16-bit hex value that contains the endurance group identifier.	
LongDes cription	This property shall contain a 16-bit hex value that contains the endurance group identifier. The endurance group identifier is unique within a subsystem. Reserved values include 0.	
M andatory	Do Not Implement.	
Notes		Do not implement NVMeSetProperties as part of an EnduranceGroup.

6.6.2.23 NVMeSetProperties.Random4kReadTypicalNanoSeconds The mapping for `NVMeSetProperties.Random4kReadTypicalNanoSeconds` is summarized in Table 191.

Table 191: NVMeSetProperties.Random4kReadTypicalNanoSeconds mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeSetProperties.Ran dom4kReadTypicalNanoSeconds	**NVM Spec Property / Field:** Random 4 KiB Read Typical **NVM Spec: Section:Figure** NVMe 1.4a: Section 5.15.2.5, Figure 253: NVM Set Attributes Entry
Type	Int64	**NVM Spec Property Type:** 4 bytes **Additional NVM Spec Identifying Information:** ByteOffset: 11:08
Des cription	Indicates the typical time to complete a 4k read in 100 nano-second units when the NVM Set is in a Predictable Latency Mode Deterministic Window and there is 1 outstanding command per NVM Set.	
LongDes cription	This property shall contain the typical time to complete a 4k read in 100 nano-second units when the NVM Set is in a Predictable Latency Mode Deterministic Window and there is 1 outstanding command per NVM Set.	
M andatory	Do Not Implement.	
Notes		Do not implement NVMeSetProperties as part of an EnduranceGroup.

6.6.2.24 Status.Health

The mapping for `Status.Health` is summarized in Table 192.

Table 192: Status.Health mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Status.Health	N/A
Type	Resource.Health	N/A
Des cription	The health state of this resource in the absence of its dependent resources.	
LongDes cription	This property shall represent the health state of the resource without considering its dependent resources. The values shall conform to those defined in the Redfish Specification.	
M andatory		Do not implement
Notes	Possible Values: OK / Warning / Critical	There is not a clear mapping for health of an Endurance Group. Do not implement this property.

6.6.2.25 Status.State The mapping for `Status.State` is summarized in Table 193.

Table 193: Status.State mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Status.State	N/A
Type	Resource.State (enum)	N/A
Des cription	The known state of the resource, such as, enabled.	
LongDes cription	This property shall indicate whether and why this component is available. Enabled indicates the resource is available. Disabled indicates the resource has been intentionally made unavailable but it can be enabled. Offline indicates the resource is unavailable intentionally and requires action to make it available. InTest indicates that the component is undergoing testing. Starting indicates that the resource is becoming available. Absent indicates the resource is physically unavailable.	
M andatory		Do not implement
Notes	Possible values: Enabled / Disabled / StandbyOffline / StandbySare / InTest / Starting / ABsent / UnavaialableOffline / Deferring / Quiesced / Updating / Qualified	There is not a clear mapping for State of an Endurance Group. Do not implement this property.

6.7 NVM Set

The Redfish/Swordfish Storage Pool schema is used to represent an NVM Set.

6.7.1 Mockup

The following mockup shows a sample representation of the Storage Pool schema used to represent an NVM Set.

```
{
  "@Redfish.Copyright": "Copyright 2015-2020 SNIA. All rights reserved.",
  "@odata.id": "/redfish/v1/Storage/FabricAttachArray/StoragePools/NVMeSet1",
  "@odata.type": "#StoragePool.v1_4_0.StoragePool",
  "Id": "1",
  "Name": "Set 1",
  "Description": "First Set",
  "Status": {
    "State": "Enabled",
    "Health": "OK"
  },
  "NVMeSetProperties": {
    "SetIdentifier": "0x1F",
    "EnduranceGroupIdentifier": "0x1",
    "Random4kReadTypicalNanoSeconds": 34534345348,
    "UnallocatedNVMNamespaceCapacityBytes": 5497558138880,
    "OptimalWriteSizeBytes": 512
  },

  "Capacity": {
    "Data": {
      "AllocatedBytes": 10995116277760,
      "ConsumedBytes": 5497558138880
    }
  },

  "AllocatedVolumes": {
    "@odata.id": "/red-
    ↪   fish/v1/Storage/FabricAttachArray/StoragePools/NVMeSet1/AllocatedVolumes"
  }
}
```

6.7.2 Property Mapping

6.7.2.1 AllocatedVolumes The mapping for `AllocatedVolumes` is summarized in Table 194.

Table 194: AllocatedVolumes mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	AllocatedVolumes	**NVM Spec Property / Field: NVM Spec: Section:Figure**
Type	Volume Collection.VolumeCollection	
Des cription	A reference to the collection of volumes allocated from this storage pool.	
LongDes cription	The value of this property shall contain a reference to the collection of volumes allocated from this storage pool.	
M andatory	Required.	
Notes	A pointer to the set of namespaces allocated from this NVM Set.	The allocated volumes contains pointers to the allocated volumes objects. These are the set of namespaces created from this NVM Set.

6.7.2.2 Capacity.Data.AllocatedBytes The mapping for `Name` is summarized in Table 195

Table 195: Capacity.Data.AllocatedBytes mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	C apacity.Data.AllocatedBytes	**NVM Spec Property / Field:** Total NVM Set Capacity **NVM Spec: Section:Figure** NVMe 1.4a: Section 5.15.2.5, Figure 253: NVM Set Attributes Entry
Type	Int64	**NVM Spec Property Type:** bytes **Additional NVM Spec Identifying Information:** ByteOffset: 31:16 for Total NVM Set Capacity
Des cription	The number of bytes currently allocated by the storage system in this data store for this data type.	
LongDes cription	The value shall be the number of bytes currently allocated by the storage system in this data store for this data type.	
M andatory	Required	
Notes		

6.7.2.3 Capacity.Data.ConsumedBytes The mapping for `Capacity.Data.ConsumedBytes` is summarized in Table 196.

Table 196: Capacity.Data.ConsumedBytes mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Capacity.Data.ConsumedBytes	**NVM Spec Property / Field:** Total NVM Set Capacity, Unallocated NVM Set Capacity **NVM Spec: Section:Figure** NVMe 1.4a: Section 5.15.2.5, Figure 253: NVM Set Attributes Entry
Type	Int64	**NVM Spec Property Type:** bytes **Additional NVM Spec Identifying Information:** ByteOffset: 31:16 for Total NVM Set Capacity
Byte 47:32 for Una llocated NVM Set C apacity.		
Des cription	The maximum number of bytes that can be allocated in this data store for this data type.	
LongDes cription	The value shall be the maximum number of bytes that can be allocated in this data store for this data type.	
M andatory	Required	
Notes		This is calculated as “Total NVM Set Capacity” - “Unallocated NVM Set Capacity”.

6.7.2.4 CapacitySources The mapping for `CapacitySources` is summarized in Table 197.

Table 197: CapacitySources mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	CapacitySources	
Type	Collection(Capacity.CapacitySource)	
Description	An array of space allocations to this volume.	
LongDescription	Fully or partially consumed storage from a source resource. Each entry provides capacity allocation information from a named source resource.	
M andatory	Recommended to not implement for NVMe Drives.	
Notes	Contains the information about the providing capacity (e.g, memory) for this namespace.	

6.7.2.5 CapacitySources@odata.count The mapping for `CapacitySources@odata.count` is summarized in Table 198.

Table 198: CapacitySources@odata.count mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	CapacitySources@odata.count	
Type	(odata property)int64	
Description	Count of the number of items in the CapacitySources array.	
LongDescription		
Mandatory	Do not implement for NVMe Drives.	
Notes		

6.7.2.6 Description The mapping for `Description` is summarized in Table 199.

Table 199: Description mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Description	**NVM Spec Property / Field:** N/A
Type	String	N/A
Des cription	The description of this resource.	See note below.
LongDes cription	This object represents the description of this resource. The resource values shall comply with the Redfish Specification-described requirements.	
M andatory		
Notes	In Redfish, Description is a read-only field.	Return the common description: "An NVM Set is a collection of NVM that is separate (logically and potentially physically) from NVM in other NVM Sets. One or more namespaces may be created within an NVM Set and those namespaces inherit the attributes of the NVM Set. A namespace is wholly contained within a single NVM Set and shall not span more than one NVM Set."

6.7.2.7 Links.OwningStorageResource The mapping for `Links.OwningStorageResource` is summarized in Table 200.

Table 200: Links.OwningStorageResource mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Links.OwningStorageResource	
Type	Storage.Storage	
Des cription	A pointer to the Storage resource that owns or contains this StoragePool.	
LongDes cription	This shall be a pointer to the Storage resource that owns or contains this StoragePool.	
M andatory	Required.	
Notes	Contains a pointer to the NVM Subsystem that contains this NVM Set.	

6.7.2.8 Name

The mapping for `Name` is summarized in Table 201

Table 201: Name mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Name	**NVM Spec Property / Field:** NVMSETID **NVM Spec: Section:Figure** NVMe 1.4a: Section 5.15.2.5, Figure 253: NVM Set Attributes Entry
Type	String	**NVM Spec Property Type:** 16-bit value **Additional NVM Spec Identifying Information:** ByteOffset: 01:00
Des cription	The name of the resource or array member.	
LongDes cription	This object represents the name of this resource or array member. The resource values shall comply with the Redfish Specification-described requirements. This string value shall be of the 'Name' reserved word format.	
M andatory	Required	
Notes	In Redfish, Name is a read-only field.	Map the NVMSETID field to a string with the format: "0xABCD"

6.7.2.9 NVMeProperties.NVMePoolType The mapping for `NVMeProperties.NVMePoolType` is summarized in Table 202.

Table 202: NVMeProperties.NVMePoolType

	Redfish/Swordfish	NVMe / NVMe-oF
Property	StoragePool.NVMePoolType	
Type	Enum	
Des cription	Indicates whether the StoragePool is used as an EnduranceGroup or an NVMSet.	
LongDes cription	This property shall indicate whether the StoragePool is used as an EnduranceGroup or an NVMSet.	
M andatory	Required	
Notes	Set as “NVMSet”	

6.7.2.10 NVMeEnduranceGroupProperties.PredictedMediaLifeLeftPercent

The mapping for `NVMeEnduranceGroupProperties.PredictedMediaLifeLeftPercent` is summarized in Table 203.

Table 203: NVMeEnduranceGroupProperties.PredictedMediaLifeLeftPercent mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMe EnduranceGroupProperties.PredictedMediaLifeLeftPercent	"Percentage Used" in the Endurance Group Log. See 5.14.1.9
Type	Decimal %	
Des cription	The percentage of reads and writes that are predicted to be available for the media.	
LongDes cription	This property shall contain an indicator of the percentage of life remaining in the drive's media.	
M andatory	Do Not Implement.	
Notes		Do not implement N VMeEnduranceGroupProperties as part of an NVM Set.

6.7.2.11 NVMeEnduranceGroupProperties.EndGrpLifetime.PercentUsed The mapping for `NVMeEnduranceGroupProperties.EndGrpLifetime.PercentUsed` is summarized in Table 204.

Table 204: NVMeEnduranceGroupProperties.EndGrpLifetime.PercentUsed mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	N VMeEnduranceGroupProperties .EndGrpLifetime.PercentUsed	“Percentage Used” in the Endurance Group Log. See 5.14.1.9
Type	Int64	
Des cription	A vendor-specific estimate of the percent life used for the endurance group based on the actual usage and the manufacturer prediction of NVM life.	
LongDes cription	This property shall contain A vendor-specific estimate of the percent life used for the endurance group based on the actual usage and the manufacturer prediction of NVM life. A value of 100 indicates that the estimated endurance of the NVM in the Endurance Group has been consumed, but may not indicate an NVM failure. According to the NVMe and JEDEC specs, the value is allowed to exceed 100. Percentages greater than 254 shall be represented as 255.	
M andatory	Do Not Implement.	

	Redfish/Swordfish	NVMe / NVMe-oF
Notes		Do not implement NVMeEnduranceGroupProperties as part of an NVM Set.

6.7.2.12 NVMeEnduranceGroupProperties.EndGrpLifetime.EnduranceEstimate

The mapping for `NVMeEnduranceGroupProperties.EndGrpLifetime.EnduranceEstimate` is summarized in Table 205.

Table 205: NVMeEnduranceGroupProperties.EndGrpLifetime.EnduranceEstimate mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeEnd uranceGroupProperties.EndGr pLifetime.EnduranceEstimate	“Endurance Estimate” in the Endurance Group Log. See 5.14.1.9
Type	Int64	
Des cription	This property contains an estimate of the total number of data bytes that may be written to the Endurance Group over the lifetime of the Endurance Group assuming a write amplication of 1.	
LongDes cription	This property shall contain an estimate of the total number of data bytes that may be written to the Endurance Group over the lifetime of the Endurance Group assuming a write amplication of 1. The value is reported in billions, where a value of 1 corresponds to 1 billion bytes written, and is rounded up. A value of zero indicates endurance estimates are unsupported.	
M andatory	Do Not Implement.	
Notes		Do not implement N VMeEnduranceGroupProperties as part of an NVM Set.

6.7.2.13 NVMeEnduranceGroupProperties.EndGrpLifetime.DataUnitsRead

The mapping for `NVMeEnduranceGroupProperties.EndGrpLifetime.DataUnitsRead` is summarized in Table 206.

Table 206: NVMeEnduranceGroupProperties.EndGrpLifetime.DataUnitsRead mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVM eEnduranceGroupProperties.E ndGrpLifetime.DataUnitsRead	
Type	Int64	
Des cription	The property contains the total number of data units read from this endurance group.	
LongDes cription	The property shall contain the total number of data units read from this endurance group. This value does not include controller reads due to internal operations such as garbage collection. The value is reported in billions, where a value of 1 corresponds to 1 billion bytes written, and is rounded up. A value of zero indicates the property is unsupported.	
M andatory	Do Not Implement.	
Notes		Do not implement N VMeEnduranceGroupProperties as part of an NVM Set.

6.7.2.14 NVMeEnduranceGroupProperties.EndGrpLifetime.DataUnitsWritten

The mapping for `NVMeEnduranceGroupProperties.EndGrpLifetime.DataUnitsWritten` is summarized in Table 207.

Table 207: NVMeEnduranceGroupProperties.EndGrpLifetime.DataUnitsWritten mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeEn duranceGroupProperties.EndG rpLifetime.DataUnitsWritten	"Data Units Written" in the Endurance Group Log. See 5.14.1.9
Type	Int64	
Des cription	The property contains the total number of data units written from this endurance group.	
LongDes cription	The property shall contain the total number of data units written from this endurance group. This value does not include controller writes due to internal operations such as garbage collection. The value is reported in billions, where a value of 1 corresponds to 1 billion bytes written, and is rounded up. A value of zero indicates the property is unsupported.	
M andatory	Do Not Implement.	
Notes		Do not implement N VMeEnduranceGroupProperties as part of an NVM Set.

6.7.2.15 NVMeEnduranceGroupProperties.EndGrpLifetime.MediaUnitsWritten

The mapping for `NVMeEnduranceGroupProperties.EndGrpLifetime.MediaUnitsWritten` is summarized in Table 208.

Table 208: NVMeEnduranceGroupProperties.EndGrpLifetime.MediaUnitsWritten mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeEnd uranceGroupProperties.EndGr pLifetime.MediaUnitsWritten	“Media Units Written” in the Endurance Group Log. See 5.14.1.9
Type	Int64	
Des cription	The property contains the total number of data units written from this endurance group.	
LongDes cription	The property shall contain the total number of data units written from this endurance group. This value includes host and controller writes due to internal operations such as garbage collection. The value is reported in billions, where a value of 1 corresponds to 1 billion bytes written, and is rounded up. A value of zero indicates the property is unsupported.	
M andatory	Do Not Implement.	
Notes		Do not implement N VMeEnduranceGroupProperties as part of an NVM Set.

6.7.2.16 NVMeEnduranceGroupProperties.EndGrpLifetime.HostReadCommandCount

The mapping for `NVMeEnduranceGroupProperties.EndGrpLifetime.HostReadCommandCount` is summarized in Table 209.

Table 209: NVMeEnduranceGroupProperties.EndGrpLifetime. HostReadCommandCount mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeEndura nceGroupProperties.EndGrpLi fetime.HostReadCommandCount	“Host Read Commands” in the Endurance Group Log. See 5.14.1.9
Type	Int64	
Des cription	This property contains the number of read commands completed by all controllers in the NVM subsystem for the Endurance Group.	
LongDes cription	This property shall contain the number of read commands completed by all controllers in the NVM subsystem for the Endurance Group. For the NVM command set, the is the number of compare commands and read commands.	
M andatory	Do Not Implement.	
Notes		Do not implement N VMeEnduranceGroupProperties as part of an NVM Set.

6.7.2.17 NVMeEnduranceGroupProperties.EndGrpLifetime.HostWriteCommandCount

The mapping for `NVMeEnduranceGroupProperties.EndGrpLifetime.HostWriteCommandCount` is summarized in Table 210.

Table 210: NVMeEnduranceGroupProperties.EndGrpLifetime. HostWriteCommandCount mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeEnduranceGroupProperties.EndGrpLifetime.HostWriteCommandCount	“Host Write Commands” in the Endurance Group Log. See 5.14.1.9
Type	Int64	
Des cription	This property contains the number of write commands completed by all controllers in the NVM subsystem for the Endurance Group.	
LongDes cription	This property shall contain the number of write commands completed by all controllers in the NVM subsystem for the Endurance Group. For the NVM command set, the is the number of compare commands and write commands.	
M andatory	Do Not Implement.	
Notes		Do not implement NVMeEnduranceGroupProperties as part of an NVM Set.

6.7.2.18 NVMeEnduranceGroupProperties.EndGrpLifetime.MediaAndDataIntegrityErrorCount

The mapping for `NVMeEnduranceGroupProperties.EndGrpLifetime.MediaAndDataIntegrityErrorCount` is summarized in Table 211.

Table 211: NVMeEnduranceGroupProperties.EndGrpLifetime. MediaAndDataIntegrityErrorCount mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeEnduranceGroupProperties.EndGrpLifetime.MediaAndDataIntegrityErrorCount	“Media and Data Integrity Errors” in the Endurance Group Log. See 5.14.1.9
Type	Int64	
Description	This property contains the number of occurences where the controller detected an unrecovered data integrity error for the Endurance Group.	
LongDescription	This property shall contain the number of occurences where the controller detected an unrecovered data integrity error for the Endurance Group. Errors such as uncorrectable ECC, CRC checksum failure, or LBA tag mismatch are included in this field.	
M andatory	Do Not Implement.	
Notes		Do not implement NVMeEnduranceGroupProperties as part of an NVM Set.

6.7.2.19 NVMeEnduranceGroupProperties.EndGrpLifetime.ErrorInformationLogEntryCount

The mapping for `NVMeEnduranceGroupProperties.EndGrpLifetime.ErrorInformationLogEntryCount` is summarized in Table 212.

Table 212: NVMeEnduranceGroupProperties.EndGrpLifetime. ErrorInformationLogEntryCount mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeEnduranceGroupP roperties.EndGrpLifetime.Er rorInformationLogEntryCount	"Number of Error Information Log Entries" in the Endurance Group Log. See 5.14.1.9
Type	Int64	
Des cription	This property contains the number of error information log entries over the life of the controller for the endurance group.	
LongDes cription	This property shall contain the number of error information log entries over the life of the controller for the endurance group.	
M andatory	Do Not Implement.	
Notes		Do not implement N VMeEnduranceGroupProperties as part of an NVM Set.

6.7.2.20 NVMeSetProperties.SetIdentifier

The mapping for `NVMeSetProperties.SetIdentifier` is summarized in Table 213.

Table 213: NVMeSetProperties.SetIdentifier

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMe SetProperties.SetIdentifier	**NVM Spec Property / Field:** NVMSETID **NVM Spec: Section:Figure** NVMe 1.4a: Section 5.15.2.5, Figure 253: NVM Set Attributes Entry
Type	String	**NVM Spec Property Type:** 16-bit value **Additional NVM Spec Identifying Information:** ByteOffset: 01:00
Des cription	A 16-bit hex value that contains the NVMe Set identifier.	
LongDes cription	This property shall contain a 16-bit hex value that contains the NVMe Set identifier. The NVM Set identifier is unique within a subsystem. Reserved values include 0.	
M andatory	Required	
Notes		Return as hex value as described in the Swordfish schema.

6.7.2.21 NVMeSetProperties.OptimalWriteSizeBytes The mapping for `NVMeSetProperties.OptimalWriteSizeBytes` is summarized in Table 214.

Table 214: NVMeSetProperties.OptimalWriteSizeBytes mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeSetPrope rties.OptimalWriteSizeBytes	**NVM Spec Property / Field:** OptimalWriteSize **NVM Spec: Section:Figure** NVMe 1.4a: Section 5.15.2.5, Figure 253: NVM Set Attributes Entry
Type	Int64	**NVM Spec Property Type:** Bytes **Additional NVM Spec Identifying Information:** ByteOffset: 15:12
Des cription	This property contains the Optimal Write Size in Bytes for this NVMe Set.	
LongDes cription	This property shall contain the Optimal Write Size in Bytes for this NVMe Set.	
M andatory	Required	
Notes		

6.7.2.22 NVMeSetProperties.EnduranceGroupIdentifier The mapping for `NVMeSetProperties.EnduranceGroupIdentifier` is summarized in Table 215.

Table 215: NVMeSetProperties.EnduranceGroupIdentifier mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeSetProperti es.EnduranceGroupIdentifier	**NVM Spec Property / Field:** EnduranceGroupIdentifier **NVM Spec: Section:Figure** NVMe 1.4a: Section 5.15.2.5, Figure 253: NVM Set Attributes Entry
Type	String	**NVM Spec Property Type:** 2 bytes **Additional NVM Spec Identifying Information:** ByteOffset: 03:02
Des cription	A 16-bit hex value that contains the endurance group identifier.	
LongDes cription	This property shall contain a 16-bit hex value that contains the endurance group identifier. The endurance group identifier is unique within a subsystem. Reserved values include 0.	
M andatory	Required	
Notes		

6.7.2.23 NVMeSetProperties.Random4kReadTypicalNanoSeconds The mapping for `NVMeSetProperties.Random4kReadTypicalNanoSeconds` is summarized in Table 216.

Table 216: NVMeSetProperties.Random4kReadTypicalNanoSeconds mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeSetProperties.Ran dom4kReadTypicalNanoSeconds	**NVM Spec Property / Field:** Random 4 KiB Read Typical **NVM Spec: Section:Figure** NVMe 1.4a: Section 5.15.2.5, Figure 253: NVM Set Attributes Entry
Type	Int64	**NVM Spec Property Type:** 4 bytes **Additional NVM Spec Identifying Information:** ByteOffset: 11:08
Des cription	Indicates the typical time to complete a 4k read in 100 nano-second units when the NVM Set is in a Predictable Latency Mode Deterministic Window and there is 1 outstanding command per NVM Set.	
LongDes cription	This property shall contain the typical time to complete a 4k read in 100 nano-second units when the NVM Set is in a Predictable Latency Mode Deterministic Window and there is 1 outstanding command per NVM Set.	
M andatory	Required	
Notes		Convert from 100 nanosecond units to nanosecond units.

6.7.2.24 NVMeSetProperties.UnallocatedNVMNamespaceCapacityBytes The mapping for `NVMeSetProperties.UnallocatedNVMNamespaceCapacityBytes` is summarized in Table 217.

Table 217: NVMeSetProperties.Random4kReadTypicalNanoSeconds mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeSetProperties.UnallocatedNVMNamespaceCapacityBytes	**NVM Spec Property / Field:** Random 4 KiB Read Typical **NVM Spec: Section:Figure** NVMe 1.4a: Section 5.15.2.5, Figure 253: NVM Set Attributes Entry
Type	Int64	**NVM Spec Property Type:** 4 bytes **Additional NVM Spec Identifying Information:** ByteOffset: 47:32
Des cription	Indicates the unallocated capacity of the NVMe Set in bytes.	
LongDes cription	This property shall contain the unallocated capacity of the NVMe Set in bytes.	
M andatory	Required	
Notes		Convert from 100 nanosecond units to nanosecond units.

6.7.2.25 Status.State The mapping for `Status.State` is summarized in Table 218.

Table 218: Status.State mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Status.State	**NVM Spec Property / Field:** N/A
Type	Resource.State (enum)	N/A

	Redfish/Swordfish	NVMe / NVMe-oF
Des cription	The known state of the resource, such as, enabled.	
LongDes cription	This property shall indicate whether and why this component is available. Enabled indicates the resource is available. Disabled indicates the resource has been intentionally made unavailable but it can be enabled. Offline indicates the resource is unavailable intentionally and requires action to make it available. InTest indicates that the component is undergoing testing. Starting indicates that the resource is becoming available. Absent indicates the resource is physically unavailable.	
M andatory		Do not implement.
Notes	Possible values: Enabled / Disabled / StandbyOffline / StandbySpare / InTest / Starting / ABsent / UnavaialableOffline / Deferring / Quiesced / Updating / Qualified	There is not a clear mapping for State of an NVM Set. Do not implement this property.

6.7.2.26 Status.Health The mapping for `Status.Health` is summarized in Table 219.

Table 219: Status.Health mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Status.Health	**NVM Spec Property / Field:** N/A
Type	Resource.Health	N/A
Des cription	The health state of this resource in the absence of its dependent resources.	
LongDes cription	This property shall represent the health state of the resource without considering its dependent resources. The values shall conform to those defined in the Redfish Specification.	
M andatory		Do not implement.
Notes	Possible Values: OK / Warning / Critical	There is not a clear mapping for health of an NVM Set. Do not implement this property.

6.8 Drive

The Redfish/Swordfish Drive schema represent the physical attributes of an NVMe drive object.

6.8.1 Mockup

The following mockup shows a sample representation of the Volume schema used to represent an NVM Namespace.

```
{
  "@odata.id": "/redfish/v1/Chassis/SimplestNVMeSSD/Drives/SimplestNVMeSSD",
  "@odata.type": "#Drive.v1_9_0.Drive",
  "IndicatorLED": "Lit",
  "Model": "ST9146802SS",
  "Revision": "S20A",
  "Status": {
    "State": "Enabled",
    "Health": "OK"
  },
  "CapacityBytes": 899527000000,
  "FailurePredicted": false,
  "Protocol": "NVMe",
  "MediaType": "SSD",
  "Manufacturer": "Contoso",
  "SerialNumber": "72D0A037FRD26",
  "PartNumber": "SG0GP8811253178M02GJA00",
  "Identifiers": [{
    "DurableNameFormat": "NAA",
    "DurableName": "500003942810D13A"
  }],
  "CapableSpeedGbs": 12,
  "NegotiatedSpeedGbs": 12,
  "Links": {
    "Volumes": [{
      "@odata.id": "/redfish/v1/Systems/Sys-
      ↪  1/Storage/SimplestNVMeSSD/Volumes/SimpleNamespace"
    }]
  },
  "Actions": {
    "#Drive.Reset": {
      "target": "/red-
      ↪  fish/v1/Chassis/SimplestNVMeSSD/Drives/SimplestNVMeSSD/Actions/Drive.Reset"
    }
  }
}
```

6.8.2 Property Mapping

6.8.2.1 Actions.#Drive.Reset The mapping for `Actions.#Drive.Reset` is summarized in Table 220.

Table 220: Actions.#Drive.Reset mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Actions.#Drive.Reset	**NVM Spec Property / Field:** Shutdown Notification (SHN) **NVM Spec: Section:Figure** Figure 78: Offset 14h, Bits 15:14
Type	Action (Special form of POST)	
Des cription	This action resets this drive.	
LongDes cription	This action shall reset this drive.	
M andatory	Required for NVMe Drives	
Notes	This action has a mandatory property of “ResetType”, which can be any of On/ForceOff/GracefulShutdown/GracefulRestart/Nmi/ForceRestart/ForceOn/PushPowerButton/PowerCycle.	Usage: A normal NVM Subysystem shutdown maps to GracefulShutdown; Subsystem Reset maps to ForceRestart; abrupt Subsystem Shutdown maps to ForceOff. If an implementation that supports the functionality, then they may implement PowerCycle.

6.8.2.2 Actions.#Drive.SecureErase The mapping for `Actions.#Drive.SecureErase` is summarized in Table 221.

Table 221: Actions.#Drive.SecureErase mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Actions.#Drive.SecureErase	**NVM Spec Property / Field:** Sanitize Operation **NVM Spec: Section:Figure** NVM Base Specification 1.4a: Section 9 “Admin Command Set”: Figure 141: Opcodes for Admin Commands - NVM Command Set Specific NVM Base Specification 1.4a: Section 5.24 Sanitize Command NVM Base Specification 1.4a: Section 8.15 Sanitize Operations NVM Base Specification 1.4a: Annex A “Sanitize Operation Considerations (Informative)”
Type	Action (Special form of POST)	NVMe Administrative command
Des cription	This action securely erases the contents of the drive.	The sanitize administrative command operation makes all user data previously written to the device inaccessible.
LongDes cription	This action shall securely erase the drive.	The Sanitize command is used to start a sanitize operation or to recover from a previously failed sanitize operation. The sanitize operation types that may be supported are Block Erase, Crypto Erase, and Overwrite. All sanitize operations are processed in the background (i.e., completion of the Sanitize command does not indicate completion of the sanitize operation)

	Redfish/Swordfish	NVMe / NVMe-oF
M andatory	Yes	No
Notes	The action parameter SanitizationType supported are: BlockErase, CryptographicErase, and Overwrite. When Overwrite is used, OverwritePasses must also be specified.	Maps to sanitize. Implementation can support any variant.

6.8.2.3 Assembly.BinaryDataURI The mapping for `Assembly.BinaryDataURI` is summarized in Table 222.

Table 222: Assembly.BinaryDataURI mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Assembly.BinaryDataURI	**NVMe-MI Spec Property / Field:** Vital Product Data (VPD) **NVM Spec: Section:Figure** NVMe-MI: Section 9.2
Type	String	
Des cription	The URI at which to access an image of the assembly information.	

	Redfish/Swordfish	NVMe / NVMe-oF
LongDes cription	This property shall contain the URI at which to access an image of the assembly information, using the Redfish protocol and authentication methods. The Service provides this URI for the download of the OEM-specific binary image of the assembly data. An HTTP GET from this URI shall return a response payload of MIME time `application/octet-stream`. If the service supports it, an HTTP PUT to this URI shall replace the binary image of the assembly.	
M andatory	Recommended	
Notes		Use to map binary blob via NVMe MI to the VPD.

6.8.2.4 BlockSizeBytes The mapping for `BlockSizeBytes` is summarized in Table 223.

Table 223: BlockSizeBytes mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	BlockSizeBytes	**NVM Spec Property / Field:** Formatted LBA Size (FLBAS) **NVM Spec: Section:Figure** 247: byte 26
Type	Int64	
Des cription	The size, in bytes, of the smallest addressable unit, or block.	

	Redfish/Swordfish	NVMe / NVMe-oF
LongDes cription	This property shall contain size of the smallest addressable unit of the associated drive.	
M andatory	Required	
Notes		Follow index in 247: Byte 26 (bits 3:0) to the LBA format structure to get the metadata size and LBA data size combination. These combined values are BlockSizeBytes.

6.8.2.5 CapableSpeedGpbs

The mapping for `CapableSpeedGpbs` is summarized in Table 224.

Table 224: CapableSpeedGpbs mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	CapableSpeedGpbs	**NVM Spec Property / Field:** PXCAP+Ch **NVM Spec: Section:Figure** 53
Type	Decimal	
Des cription	The speed, in gigabit per second (Gbit/s), at which this drive can communicate to a storage controller in ideal conditions.	
LongDes cription	This property shall contain fastest capable bus speed, in gigabit per second (Gbit/s), of the associated drive.	
M andatory	Required	

	Redfish/Swordfish	NVMe / NVMe-oF
Notes		For PCIe, this is in the PCIe link capabilities (PXCAP+Ch). Figure 53. For NVMe-oF this is not specified; use the value for the highest supported native capability.

6.8.2.6 CapacityBytes The mapping for `CapacityBytes` is summarized in Table 226.

For drives supporting only a single namespace (no or default endurance group / set only):

Table 225: CapacityBytes for single namespace mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	CapacityBytes	**NVM Spec Property / Field:** Namespace Size (NSZE) **NVM Spec: Section:Figure** NVMe 1.4a: Section 5.15.2.1 (Identify Namespace), Figure 247
Type	Int64	**NVM Spec Property Type:** int 64 **Additional NVM Spec Identifying Information:** ByteOffset: 07:00, Identify Namespace data structure
Des cription	The size, in bytes, of this drive.	The total size of the NVM allocated to this namespace.

	Redfish/Swordfish	NVMe / NVMe-oF
LongDes cription	This property shall contain the raw size, in bytes, of the associated drive.	The total size of the NVM allocated to this namespace. The value is in bytes. This field shall be supported if the Namespace Management capability (refer to NVMe Base Specification section 8.12) is supported. This field may not correspond to the logical block size multiplied by the Namespace Size field. Due to thin provisioning or other settings (e.g., endurance), this field may be larger or smaller than the Namespace Size reported.
M andatory	Required	No
Notes	Reporting capacity in bytes is the Redfish and Swordfish standard mechanism.	Returned in bytes 63:48 of the Identify Namespace Data Structure (NVM Command Set Specific). Reference NVMe Base Specification section in 5.15.2.1 and figure 247.

For drives supporting multiple namespaces:

Table 226: CapacityBytes for single namespace mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	CapacityBytes	The Total Endurance Group Capacity. Just the “Total Endurance Group Capacity” in the Endurance Group Log. See 5.14.1.9 + TP 4009

	Redfish/Swordfish	NVMe / NVMe-oF
Type	Int64	
Des cription	The number of bytes currently allocated by the storage system in this data store for this data type.	
LongDes cription	The value shall be the number of bytes currently allocated by the storage system in this data store for this data type.	
M andatory		
Notes		Note: This is not in 1.4a because TP 4009 was not integrated.

6.8.2.7 Description The mapping for `Description` is summarized in Table 227.

Table 227: Description mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Description	N/A
Type	String	N/A
Des cription	The description of this resource.	N/A
LongDes cription	This object represents the description of this resource. The resource values shall comply with the Redfish Specification-described requirements.	N/A
M andatory	Yes	N/A
Notes	In Redfish, Description is a read-only field.	

6.8.2.8 EncryptionAbility The mapping for `EncryptionAbility` is summarized in Table 228.

Table 228: EncryptionAbility mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	EncryptionAbility	N/A
Type	Enum (EncryptionAbility)	
Des cription	The encryption ability of this drive.	
LongDes cription	This property shall contain the encryption ability for the associated drive.	
M andatory	Required	
Notes	Required when encryption is supported. Available values: Non e/SelfEncryptingDrive/Other	Maps to vendor capabilities.

6.8.2.9 EncryptionStatus The mapping for `EncryptionStatus` is summarized in Table 229.

Table 229: EncryptionStatus mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	EncryptionStatus	N/A
Type	Drive.EncryptionStatus	
Des cription	The status of the encryption of this drive.	
LongDes cription	This property shall contain the encryption status for the associated drive.	
M andatory	DependsOn EncryptionAbility != None	

	Redfish/Swordfish	NVMe / NVMe-oF
Notes	Must be implemented and set when Encryption is enabled (EncryptionAbility will indicate encryption capability type.) Possible values: Unlocked /locked/foreign/unencrypted	Set according to vendor specs / mapping.

6.8.2.10 FailurePredicted The mapping for `FailurePredicted` is summarized in Table 230.

Table 230: FailurePredicted mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	FailurePredicted	**NVM Spec Property / Field:** SmartHealthLog **NVM Spec: Section:Figure** 196
Type	Boolean	
Des cription	An indication of whether this drive currently predicts a failure in the near future.	
LongDes cription	This property shall indicate whether this drive currently predicts a manufacturer-defined failure.	
M andatory	Required	
Notes	Implements the results of the SMART log data.	Trigger when the smart health errors indicate an issue (e.g.,): media and data integrity errors, and self-check.

6.8.2.11 Identifiers The mapping for `Identifiers` is summarized in Table 231.

Table 231: Identifiers mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Identifiers	NVM Subsystem NVMe Qualified Name (SUBNQN)
Type	Coll ection(Resource.Identifier)	An array of identifiers
Des cription	The Durable names for the subsystem.	An array of identifiers
LongDes cription	This property shall contain a list of all known durable names for the associated subsystem.	This specifies the NVM Subsystem NVMe Qualified Name as a UTF-8 null-terminated string. Refer to NVMe Base Specification, section 7.9, for the definition of NVMe Qualified Name. Support for this field is mandatory if the controller supports revision 1.2.1 or later as indicated in the Version register (refer to section 3.1.2).
M andatory	Yes	Yes
Notes	This is an array of unique identifiers for the NVM Subsystem.	There will only be one instance in this array for Subsystem. Refer to the Identify Controller data structure (CNS 01h) bits 1023:768 in figure 249 (Identify – Identify Controller Data Structure) of the NVMe Base Specification.

6.8.2.12 Identifiers.DurableNameFormat The mapping for `Identifiers.DurableNameFormat` is summarized in Table 232.

Table 232: Identifiers.DurableNameFormat mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Id entifiers.DurableNameFormat	NVM Subsystem NVMe Qualified Name (SUBNQN)
Type	Resourc e.v1_1_0.DurableNameFormat	There is a single value for this array in Subsystem. The property type is of type NVMe Qualified Name (NQN).
Des cription	The format of the Durable names for the subsystem.	NVM Subsystem NVMe Qualified Name (SUBNQN)
LongDes cription	This specifies the format of the associated NVM Subsystem NVMe Qualified Name of type NQN. Support for this field is mandatory if the controller supports revision 1.2.1 or later as indicated in the Version register (refer to section 3.1.2).	
M andatory	No	Yes
Notes	This is an enum with multiple potential values. For this particular usage in Subsystem, there will only be one instance populated, of type NQN.	There will only be one instance in this array for Subsystem. Refer to the Identify Controller data structure (CNS 01h) bits 1023:768 in figure 249 (Identify – Identify Controller Data Structure) of the NVMe Base Specification.

6.8.2.13 Identifiers.DurableName The mapping for `Identifiers.DurableName` is summarized in Table 233.

Table 233: Identifiers.DurableName mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Identifiers.DurableName	NVM Subsystem NVMe Qualified Name (SUBNQN)
Type	Edm.String	The NVM Subsystem NVMe Qualified Name as a UTF-8 null-terminated string
Des cription	The format of the Durable names for the subsystem.	NVM Subsystem NVMe Qualified Name (SUBNQN)
LongDes cription	This specifies the NVM Subsystem NVMe Qualified Name as a UTF-8 null-terminated string. Refer to NVMe Base Specification, section 7.9, for the definition of NVMe Qualified Name. Support for this field is mandatory if the controller supports revision 1.2.1 or later as indicated in the Version register (refer to section 3.1.2).	
M andatory	No	Yes
Notes	For this particular usage in Subsystem, there will only be one instance populated in the identifiers array.	There will only be one instance in this array for Subsystem. Refer to the Identify Controller data structure (CNS 01h) bits 1023:768 in figure 249 (Identify – Identify Controller Data Structure) of the NVMe Base Specification.

6.8.2.14 IndicatorLED The mapping for `IndicatorLED` is summarized in Table 234.

Table 234: IndicatorLED mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	IndicatorLED	N/A
Type	Boolean	N/A
Des cription	An indication of whether this drive currently predicts a failure in the near future.	
LongDes cription	This property shall indicate whether this drive currently predicts a manufacturer-defined failure.	N/A
M andatory	Do Not Implement	
Notes	This property has been deprecated. See guidance / implement the LocationIndicatorActive property instead.	

6.8.2.15 Links.Volumes The mapping for `Links.Volume` is summarized in Table 235.

Table 235: Links.Volume mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Links.Volume	
Type	Collection(Volume.Volume)	
Des cription	An array of links to the volumes that this drive either wholly or only partially contains.	

	Redfish/Swordfish	NVMe / NVMe-oF
LongDes cription	This property shall contain an array of links to resources of type Volume with which this drive is associated. This property shall include all volume resources of which this drive is a member and all volumes for which this drive acts as a spare if the hot spare type is `Dedicated`.	
M andatory	Required	
Notes	This array shall contain links to all namespaces associated with this physical drive.	One way to do this would be get all of the controllers, iterate through the list and get all of the namespace ids.

6.8.2.16 Links.Volumes@odata.count The mapping for `Links.Volumes@odata.count` is summarized in Table 236.

Table 236: Links.Volumes@odata.count mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Links.Volumes@odata.count	
Type	(odata property)int64	N/A
Des cription	Count of the number of items in the Links.Volume array.	
LongDes cription		
M andatory	Required	
Notes		The number of namespaces is available from NVMe on a per controller basis.

6.8.2.17 Location The mapping for `Location` is summarized in Table 237.

Table 237: Location mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Location	N/A
Type	Co llection(Resource.Location)	
Des cription	The location of the drive.	
LongDes cription	This property shall contain location information of the associated drive.	N/A
M andatory	Do Not Implement	
Notes	This property has been deprecated. See guidance / implement the PhysicalLocation property instead.	

6.8.2.18 LocationIndicatorActive The mapping for `LocationIndicatorActive` is summarized in Table 238.

Table 238: LocationIndicatorActive mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	LocationIndicatorActive	N/A
Type	Boolean	N/A
Des cription	An indicator allowing an operator to physically locate this resource.	

	Redfish/Swordfish	NVMe / NVMe-oF
LongDescription	This property shall contain the state of the indicator used to physically identify or locate this resource. A write to this property shall update the value of IndicatorLED in this resource, if supported, to reflect the implementation of the locating function.	N/A
M andatory	Recommended	
Notes	This property replaces the IndicatorLED, which has been deprecated.	Comes from vendor.

6.8.2.19 Manufacturer The mapping for `Manufacturer` is summarized in Table 239.

Table 239: Manufacturer mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Manufacturer	**NVM Spec Property / Field: NVM Spec: Section: Figure**
Type	String	
Des cription	The manufacturer of this drive.	
LongDescription	This property shall contain the name of the organization responsible for producing the drive. This organization might be the entity from whom the drive is purchased, but this is not necessarily true.	
M andatory	Required	

	Redfish/Swordfish	NVMe / NVMe-oF
Notes		End clients expect to see the name of the company (e.g,; Contoso, BestVendor). While the value may be filled from the IdentifyController PCI Vendor ID or SubsystemID field, it would be preferable to have this filled with the actual string value of the company name.

6.8.2.20 MediaType The mapping for `MediaType` is summarized in Table 240.

Table 240: MediaType mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	MediaType	N/A
Type	enum (MediaType)	
Des cription	The type of media contained in this drive.	
LongDes cription	This property shall contain the type of media contained in the associated drive.	N/A
M andatory	Required	
Notes	Possible values: HDD/SSD/SMR.	NVMe SSD Drives to report SSD.

6.8.2.21 Model The mapping for `Model` is summarized in Table 241.

Table 241: Model mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Model	**NVM Spec Property / Field:** IdentifyController / Model Number (MN) **NVM Spec: Section: Figure** 249 byte 24:63
Type	String	N/A
Des cription	The model number for the drive.	
LongDes cription	This property shall contain the name by which the manufacturer generally refers to the drive.	N/A
M andatory	Required	
Notes		

6.8.2.22 Multipath The mapping for `Multipath` is summarized in Table 242.

Table 242: Multipath mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Multipath	**NVM Spec Property / Field:** IdentifyNamespace: NMIC **NVM Spec: Section:Figure** 247: Byte 30, bit 0
Type	Boolean	N/A
Des cription	An indication of whether the drive is accessible from multiple paths.	

	Redfish/Swordfish	NVMe / NVMe-oF
LongDes cription	This property shall indicate whether the drive is accessible by an initiator from multiple paths allowing for failover capabilities upon a path failure.	N/A
M andatory	Recommended to implement. Required property if drive is dual-ported.	
Notes		

6.8.2.23 Name The mapping for `Name` is summarized in Table 243.

Table 243: Name mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Name	**NVM Spec Property / Field:** Namespace ID (NSID) **NVM Spec: Section:Figure** NVMe 1.4a
Type	String	
Des cription	The name of the resource or array member.	N/A
LongDes cription	This object represents the name of this resource or array member. The resource values shall comply with the Redfish Specification-described requirements. This string value shall be of the 'Name' reserved word format.	
M andatory	Yes	N/A

	Redfish/Swordfish	NVMe / NVMe-oF
Notes	In Redfish, Name is a read-only field.	

6.8.2.24 NegotiatedSpeedGbps The mapping for `NegotiatedSpeedGbps` is summarized in Table 244.

Table 244: NegotiatedSpeedGbps mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NegotiatedSpeedGbps	**NVM Spec Property / Field:** N/A **NVM Spec: Section:Figure** N/A
Type	Decimal	N/A
Des cription	The speed, in gigabit per second (Gbit/s), at which this drive currently communicates to the storage controller.	
LongDes cription	This property shall contain current bus speed, in gigabit per second (Gbit/s), of the associated drive.	N/A
M andatory	Required	
Notes		For PCIe, this is in the PCIe link capabilities (PXCAP+12h). Figure 55. For NVMe-oF this is not specified; use the value for the native capability.

6.8.2.25 PhysicalLocation.Info The mapping for `PhysicalLocation.Info` is summarized in Table 245.

Table 245: PhysicalLocation.Info mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	PhysicalLocation.Info	**NVM Spec Property / Field:** N/A **NVM Spec: Section:Figure** N/A
Type	String	
M andatory	Do Not Implement	
Notes	This property has been deprecated.	

6.8.2.26 PhysicalLocation.InfoFormat The mapping for `PhysicalLocation.InfoFormat` is summarized in Table 246.

Table 246: PhysicalLocation.InfoFormat mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	PhysicalLocation.InfoFormat	**NVM Spec Property / Field:** N/A **NVM Spec: Section:Figure** N/A
Type	String	
M andatory	Do Not Implement	
Notes	This property has been deprecated.	

6.8.2.27 PhysicalLocation.PartLocation The mapping for `PhysicalLocation.PartLocation` is summarized in Table 247.

Table 247: PhysicalLocation.PartLocation mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Ph ysicalLocation.PartLocation	**NVM Spec Property / Field:** N/A **NVM Spec: Section:Figure** N/A
Type	Boolean	N/A

	Redfish/Swordfish	NVMe / NVMe-oF
Des cription	An indication of whether the drive is accessible from multiple paths.	
LongDes cription	This property shall indicate whether the drive is accessible by an initiator from multiple paths allowing for failover capabilities upon a path failure.	N/A
M andatory	Recommended to implement. Required property if drive is dual-ported.	
Notes	The drive should support this property to be filled in by a layered process (e.g., BMC). Support for any other properties in PhysicalLocation are at the discretion of the vendor.	

6.8.2.28 PredictedMediaLifetimeLeftPercent

The mapping for `PredictedMediaLifetimeLeftPercent` is summarized in Table 248.

Table 248: PredictedMediaLifetimeLeftPercent mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Predic tedMediaLifetimeLeftPercent	**NVM Spec Property / Field:** Get Log Page – SMART / Health Information Log **NVM Spec: Section:Figure** Figure 196: 05
Type	Decimal	
Des cription	The percentage of reads and writes that are predicted to be available for the media.	

	Redfish/Swordfish	NVMe / NVMe-oF
LongDes cription	This property shall contain an indicator of the percentage of life remaining in the drive's media.	N/A
M andatory	Required.	
Notes	Maps to percentage used in SMART information log	Calculate as 100 - value reported (PercentageUsed).

6.8.2.29 Protocol

The mapping for `Protocol` is summarized in Table 249.

Table 249: Protocol mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Protocol	
Type	Protocol.Protocol	
Des cription	The protocol that this drive currently uses to communicate to the storage controller.	
LongDes cription	This property shall contain the protocol that the associated drive currently uses to communicate to the storage controller for this system.	
M andatory	Required.	
Notes	Possible values (long list)	NVMe Drives shall report "NVMe".

6.8.2.30 Revision

The mapping for `Revision` is summarized in Table 250.

Table 250: Revision mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Revision	**NVM Spec Property / Field:** IdentifyController: Firmare Revision (FR) **NVM Spec: Section:Figure** 249: 71:64
Type	String	
Des cription	The revision of this drive. This is typically the firmware or hardware version of the drive.	
LongDes cription	This property shall contain the manufacturer-defined revision for the associated drive.	
M andatory	Required.	
Notes		Return the currently active firmware revision information.

6.8.2.31 RotationSpeedRPM The mapping for `RotationSpeedRPM` is summarized in Table 251.

Table 251: RotationSpeedRPM mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	RotationSpeedRPM	
Type	Decimal	
Des cription	An indication of whether the drive is accessible from multiple paths.	
LongDes cription	This property shall indicate whether the drive is accessible by an initiator from multiple paths allowing for failover capabilities upon a path failure.	

	Redfish/Swordfish	NVMe / NVMe-oF
M andatory	DependsOn MediaType value. If MediaType == SSD, (not required to) do not implement.	
Notes	Future proofing - will be a relevant property for NVMe HDDs.	

6.8.2.32 SKU The mapping for `SKU` is summarized in Table 252.

Table 252: SKU mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	SKU	N/A
Type	String	
Des cription	The SKU for this drive.	
LongDes cription	This property shall contain the stock-keeping unit (SKU) number for this drive.	
M andatory	Required.	
Notes	The drive should support this property to be filled in by a layered process (e.g., OEM manufacturing).	

6.8.2.33 SerialNumber The mapping for `SerialNumber` is summarized in Table 253.

Table 253: SerialNumber mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	SerialNumber	**NVM Spec Property / Field:** IdentifyController / SerialNumber (SN) **NVM Spec: Section: Figure** 249 byte 04:23
Type	String	
Des cription	The serial number for this drive.	
LongDes cription	This property shall contain the manufacturer-allocated number that identifies the drive.	
M andatory	Required.	
Notes		

6.8.2.34 Status.State

The mapping for `Status.state` is summarized in Table 254.

Table 254: Status.State mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Status.State	Enable (EN)
Type	Resource.State (enum)	
Des cription	The known state of the resource, such as, enabled.	

	Redfish/Swordfish	NVMe / NVMe-oF
LongDescription	This property shall indicate whether and why this component is available. Enabled indicates the resource is available. Disabled indicates the resource has been intentionally made unavailable but can be enabled. Offline indicates the resource is unavailable intentionally and requires action to make it available. InTest indicates that the component is undergoing testing. Starting indicates that the resource is becoming available. Absent indicates the resource is physically unavailable.	
M andatory	Required (for NVM Drives)	
Notes	Possible values: Enabled / Disabled / StandbyOffline / StandbySpare / InTest / Starting / Absent / UnavailableOffline / Deferring / Quiesced / Updating / Qualified	The drive should support this property to be filled in by a higher level client (e.g., BMC). The drive can self-set this drive to Enabled / Disabled / InTest /Updating. If any controller in the drive is set to Enabled, set to “Enabled”. If all controllers are set to disabled, set to “Disabled”. If a firmware update is in progress, set to “Updating”. If the drive is running a self-test, set to “InTest”.

6.8.2.35 Status.Health The mapping for `Status.Health` is summarized in Table 255.

Table 255: Status.Health mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Status.Health	**NVM Spec Property / Field:** CSTS – Controller Status **NVM Spec: Section:Figure** NVMe 1.4a: Section 3.1.6, Figure 79 **NVM Spec Property / Field:** Critical Warning **NVM Spec: Section:Figure** NVMe 1.4a: Section 5.14.1.2, SMART / Health Information, Figure 196
Type	Resource.Health	
Des cription	The health state of this resource in the absence of its dependent resources.	
LongDes cription	This property shall represent the health state of the resource without considering its dependent resources. The values shall conform to those defined in the Redfish Specification.	
M andatory	Required for NVM Drives.	
Notes	Possible Values: OK / Warning / Critical	Report to same value as set for worst-case controller Status.Health.

6.8.2.36 StatusIndicator The mapping for `StatusIndicator` is summarized in Table 256.

Table 256: StatusIndicator mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	StatusIndicator	N/A
Type	enum (StatusIndicator)	
Des cription	An indication of whether the drive is accessible from multiple paths.	
LongDes cription	This property shall indicate whether the drive is accessible by an initiator from multiple paths allowing for failover capabilities upon a path failure.	
M andatory	Required.	
Notes	Multiple values possible. Relevant values for NVMe standalone drives: OK, Fail, PredictiveFailureAnalysis.	The drive should support this property to be filled in by a higher level client (e.g., BMC). The drive can self-set this drive only to OK (or potentially fail, but only if the property has not been set by higher-level software).

6.8.2.37 WriteCacheEnabled The mapping for `WriteCacheEnabled` is summarized in Table 257.

Table 257: WriteCacheEnabled mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	SerialNumber	**NVM Spec Property / Field:** Volatile Write Cache Enable (WCE) **NVM Spec: Section:Figure** 283: 00
Type	Boolean	

	Redfish/Swordfish	NVMe / NVMe-oF
Description	An indication of whether the drive write cache is enabled.	
LongDescription	This property shall indicate whether the drive write cache is enabled.	
M andatory	Required (for NVMe Drives).	
Notes		Bits 00 of WCE indicate whether the WCE is enabled or disabled.

7 Other Feature Mapping

7.1 Introduction

There are additional features and functionality that are also desirable to be exposed via the Redfish/Swordfish management ecosystems; however, there is not necessarily a clear mapping for these to a specific portion of the NVMe specification, or the functionality has a large component provided by the NVMe device implementer. This section provides an overview of the required functionality and presentation required to present a common, standardized interface to the Redfish/Swordfish clients.

7.2 Firmware Update

Firmware update is a required function to present comprehensive management. This is presented in Redfish through the “Update” service. The Update Service provides a standardized interface to all software and firmware update mechanisms throughout the server, storage, and networking environment, and as such, provides an abstracted, standardized interface that largely manages the acquisition from the client of the image files (and subsequent distribution of those to the designated targets), as well as very coarse-grained scheduling.

Note that this mechanism does not include any “active image” management capability (e.g., rollback).

7.2.1 Firmware update for NVMe Drives

NVMe drives shall support a standard implementation of the Redfish Update service. This is fundamentally a single-step invocation process on the update service that gets the image file from a specified URI using the SimpleUpdate action.

One parameter that shall be specified is the OperationApplyTime; this indicates when to apply the image. If not explicitly specified, the implementation should assume the request is to be applied immediately.

The values of the OperationApplyTime that shall be supported by NVMe drives are: - Immediate - OnReset

Example simple update request:

```
POST /redfish/v1/UpdateService/Actions/UpdateService.SimpleUpdate HTTP/1.1
```

```
Content-Type: application/json
Content-Length: <computed-length>

{
    "ImageURI": "https://192.168.1.250/images/dev1_update.bin"
    "@Redfish.OperationApplyTime": "OnReset"
}
```

The `username` and `password` parameters summarized in Table 258 may be required to access the image. Implementations should support these. Additionally, the implementation may support the `Targets` parameter to specify the list of devices on which to apply the image. If `Targets` is not specified, the image will be applied to all applicable devices.

Table 258: Additional parameters

Parameter	Type	Description
`Username`	String	The user name to access the URI specified by the `ImageURI` parameter.
`Password`	String	The password to access the URI specified by the `ImageURI` parameter.
`Targets`	Array	An array of strings that are URIs to resources that indicate where to apply the image.

The SimpleUpdate mechanism is fundamentally a “pull” mechanism (the implementation pulls a file from a target); the UpdateService also supports a “push” mechanism (the client “pushes” the image to the implementation as part of the request payload) that may also be implemented. For more information this mechanism, using the MultipartHttpPushUri property, is documented in the Redfish Specification.

7.2.1.1 Mapping Images to NVMe Capabilities It will be up to the implementation to determine the NVMe specific parameters to use upon receipt of the image from the update service. However, the following general guidelines should be used to ensure behavioral consistency across implementations:

When receiving image: - If open slots, the implementation can choose what open slot to use, and load the image to that slot. - If all slots are full, the implementation can choose (typically any slot except the currently active image).

After loaded to a slot, the implementation will mark the image as active. Based on the “apply time” setting, the implementation setting will activate the image either immediately, or set to activate on (the next) reset.

Appendix A: Objects without a direct mapping to the NVMe model

A.1: Overview

There are a number of objects that are required for the proper integration of NVMe support within Redfish and Swordfish, but which are not supported by an entity that can be mapped directly from the various NVM Specifications. Information about these related but un-mapped objects are defined elsewhere in the documentation provided with each Swordfish release.

They are summarized in Table A.1 for ease of use, including:

- Object Name
- NVMe Device Expected Usage

Object	NVMe Device Expected Usage
NetworkAdapter	NVMe-oF Subsystems, Network-Attach Drives
Port (on NetworkAdapter)	NVMe-oF Subsystems, Network-Attach Drives
Netw orkDeviceFunction	NVMe-oF Subsystems, Network-Attach Drives
Fabric	NVMe-oF Subsystems, Network-Attach Drives and arrays
Connection	NVMe-oF Subsystems, Network-Attach Drives and arrays
Endpoint	NVMe-oF Subsystems, Network-Attach Drives and arrays
Endpoint Group	NVMe-oF Subsystems, Network-Attach Drives and arrays
Switch	NVMe-oF Subsystems, Network-Attach Drives and arrays
Port (on switch)	NVMe-oF Subsystems, Network-Attach Drives and arrays
EthernetInterface	NVMe-oF Subsystems, Network-Attach Drives and arrays, EBOF/JBOF
Manager	EBOF/JBOF, Complex devices, Arrays
NetworkProtocol	EBOF/JBOF, Complex devices, Arrays

Table A.1: Related Objects

Required usage of objects and their properties are contained in the relevant Swordfish profiles.
For example, for Ethernet-Attach drives, the NetworkAdapter, Port, and NetworkDeviceFunction required properties are specified as part of the “Ethernet-Attach NVMe Drive” profile.

A.2: Related Use Cases

The Swordfish Scalable Storage Management API User’s Guide contains real-world examples of how clients will interact with an implementation. Table A.2 summarizes the NVMe use cases as of 12 July 2022.

Title	Description
Attach a Namespace	Attach a Namespace
Deprovision a Namespace	Deprovision a Namespace
Detach a Namespace	Detach a Namespace
Provision a Namespace	Provision a Namespace
Provision an NVM Set with a Namespace	Provision an NVM Set associated with a Namespace
Report capacity for a Namespace	Report capacity for a Namespace
Report remaining life for a Namespace	Report remaining life for a Namespace

Table A.2: NVMe Use Case Summary

Annex B: Bibliography

B.1 Overview

The following referenced documents provide important support for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

B.2 Informational references

The informational references are summarized in Table B.1.

Tag	Title (Version)	Author	URL
Pro files	Swordfish Profile Bundle Working Draft	SNIA	<https://www.snia.org/forums/smi/swordfish>
Users Guide	wordfish Scalable Storage Management API User's Guide	SNIA	<https://www.snia.org/forums/smi/swordfish>

Table B.1: Informational References