Swordfish NVMe Model Overview and Mapping Guide

Version: 1.2.7

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.

Working Draft

Publication of this Working Draft for review and comment has been approved by the Scalable Storage Management Technical Work Group. This draft represents a 'best effort' attempt by the Scalable Storage Management Technical Work Group to reach preliminary consensus, and it may be updated, replaced, or made obsolete at any time. This document should not be used as reference material or cited as other than a 'work in progress.' Suggestions for revision should be directed to http://www.snia.org/feedback.

Last Updated: 21 May 2024

Contents

List of Tables

List of Figures

USAGE

- Neither the name of Storage Networking Industry Association nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

DISCLAIMER

The information contained in this publication is subject to change without notice. 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 SNIA, 5201 Great America Parkway, Suite 320, Santa Clara, CA 95054, USA.

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

Date	Rev	Notes
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.
16 March 2023	1.2.5	Updated all references from NVMe 1.4 Specification to the NVM Express Base Specification 2.0c throughout. Update references to figures in NVMe 2.0 Base Specification as well as the NVM Express NVM Command Set Specification 1.0c, NVMe over PCIe Transport Specification 1.0c, and NVMe-MI 1.2c Specification.
		Document usage of SecuritySend and SecurityReceive Actions on StorageController
		Define usage for Links.Controllers in Namespace (volume)
		Errata Fixes:
		&nbps; Clarify references to CapacitySource collection vs instance
		Standardize nomenclature for Mandatory instances throughout
20 June 2023	1.2.5a	Release as SNIA Standard
22 January 2024	1.2.6	Release as Working Draft
		Updated cross-references.
		Copyrights updated to 2024.
		Updates to SNIA Front Matter.
		Updated Contributing members.

Date	Rev	Notes
		Added NVMeDomains Object, and updated corresponding mockups.
		Added missing reference to Drives and EBOFs in HTML version of document.
		Updated NVMeNamespaceProperties to new guidance for LBAFormats representation (in both mockups and mapping properties). Removed/deprecated NVMeNamespace-Properties.NumberLBAF ormats.
		Added new section Namespace Capabilities.
		Errata fixes - corrected references to PredictedMediaLifetLeftPercent and NVMeoFDiscoverySubsystems properties.
9 April 2024	1.2.6	Release as SNIA Standard
		Errata fix - include missing NVMeDomainds section in doc from build error.
21 May 2024	1.2.7	Release as Working Draft
		- Added `ConfigurationLock`, `TargetConfigurationLockLevel`, and
		`ConfigurationLockState` definitions, mappings, and guidance.
		- Added Domain and Implicit Domain examples and guidance.
		- Added NVMeDomains Object, and updated corresponding mockups.

Date	Rev	Notes
		- Added mapping for MaxNamespaceAttachmentsAllowed to controllers.
		- Added guidance for StoragePools in Subsystem
		- Added new NVMeSubsystemProperties for Namespace Management: `MaxNamespacesSupported` and `SharedNamespaceControllerAttachmentSupported`
		- Add updated mapping guidance for `DisplayName` to use Namespace Admin Label
		- Errata Fixes:
		- corrected references to PredictedMediaLifetLeftPercent and NVMeoFDiscoverySubsystems properties.

About SNIA

SNIA is a not-for-profit global organization made up of corporations, universities, startups, and individuals. The members collaborate to develop and promote vendor-neutral architectures, standards, and education for management, movement, and security for technologies related to handling and optimizing data. SNIA focuses on the transport, storage, acceleration, format, protection, and optimization of infrastructure for data. Learn more at www.snia.org.

Acknowledgements

Table 2: Contributors

Member	Representatives (* – prior employer)
Broadcom Inc.	Richelle Ahlvers*
Celestica	Krishnakumar Gowravaram
Cisco Systems, Inc.	Krishnakumar Gowravaram*
Code Construct	Jeremy Kerr
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

Member	Representatives (* – prior employer)
	Tom Rainey
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)	Autho r	URL
ISO-860 1	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

Tag	Title (Version)	Autho r	URL
Swordfi sh	Swordfish Scalable Storage Management API Specification (v1.2.5)	SNIA	<https://www.snia .org/tech_activit ies/standards/cur r_standards/sword fish>
NVMe-Ba se	NVM Express Base Specification 2.0c	NVM Expre ss	<https://nvmexpre ss.org/developers /nvme-specificati on/>
NVMe-Co mmand	NVM Express NVM Command Set Specification 1.0c	NVM Expre ss	<https://nvmexpre ss.org/developers /nvme-command-set -specifications/>
NVMe-Tr ansport	NVMe over PCIe Transport Specification 1.0c	NVM Expre ss	<https://nvmexpre ss.org/developers /nvme-transport-s pecifications/>
NVMe-MI	NVMe-MI 1.2c Specification	NVM Expre ss	<https://nvmexpre ss.org/developers /nvme-mi-specific ation/>
SPC-5	SCSI Primary Commands - 5	INCIT S	<https://webstore .ansi.org/standar ds/incits/incits5 022019>
URI	Uniform Resource Identifier (URI): Generic Syntax	IETF	<https://datatrac ker.ietf.org/doc/ html/rfc3986>

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 Subsystem 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.
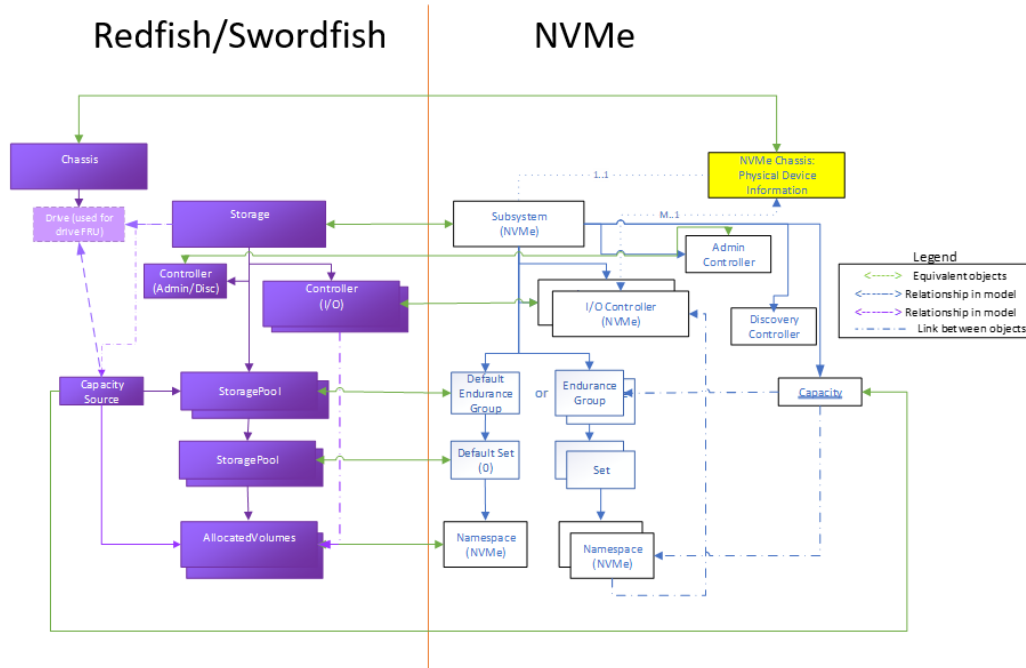
Figure 1: Subsystem model

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.
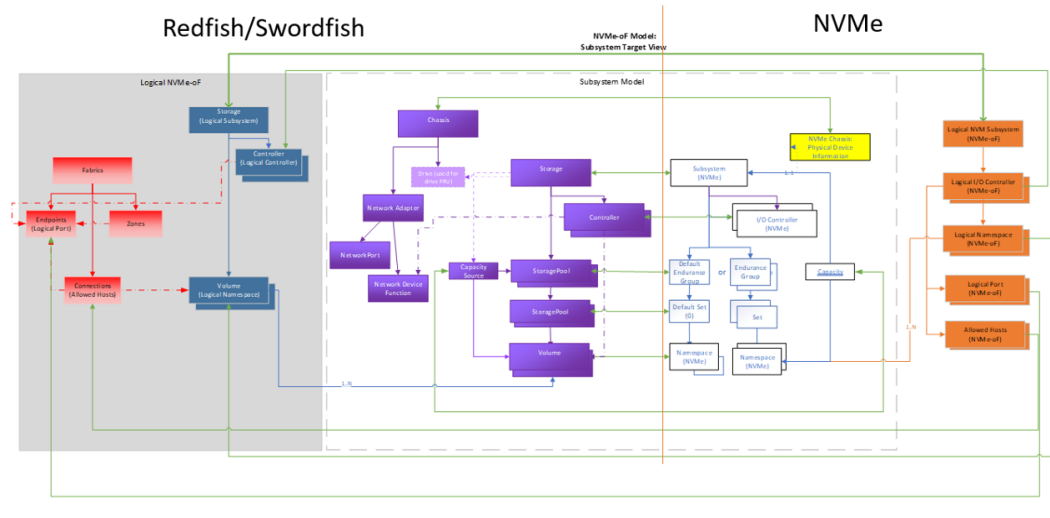
Figure 2: NVMe-oF Subsystem Model

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.
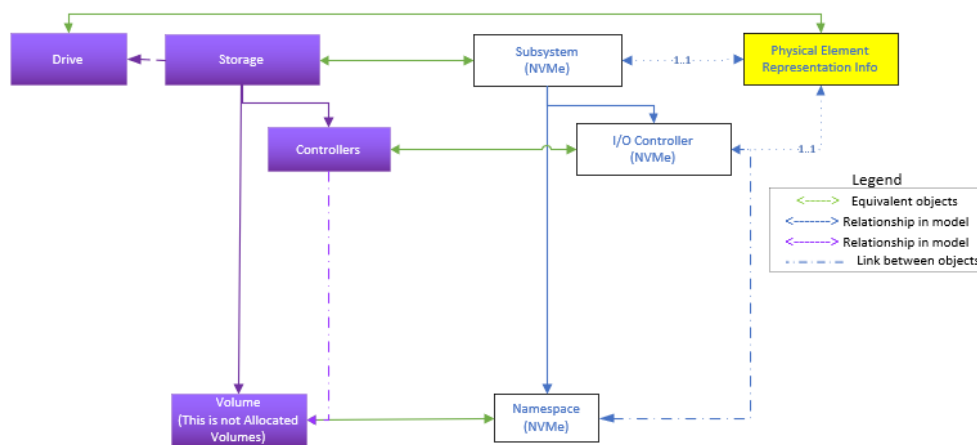
Figure 3: Simple SSD instance diagram

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.
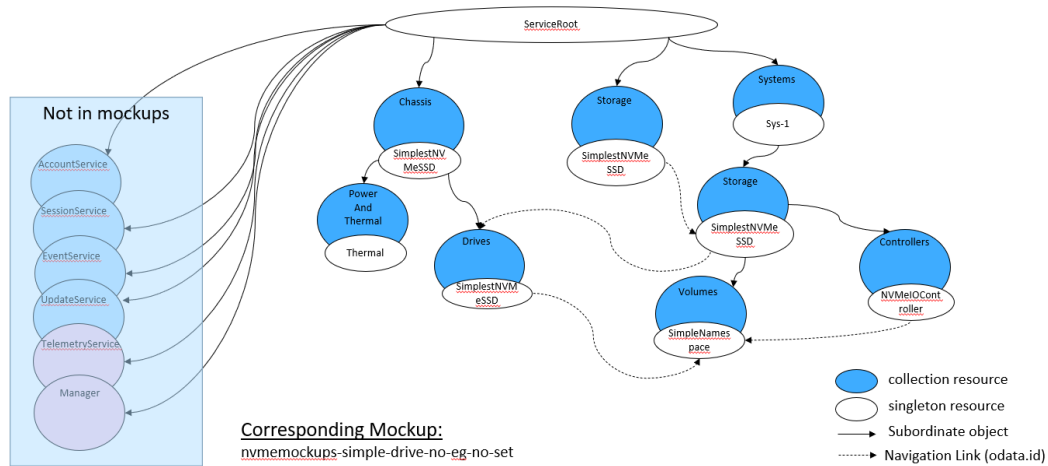
Figure 4: Simple SSD mockup example

5.2.4 Mockup

A corresponding mockup for this configuration can be found at http://simple-ssd-mockups.swordfishmockups.com/redfish/v1/.

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.
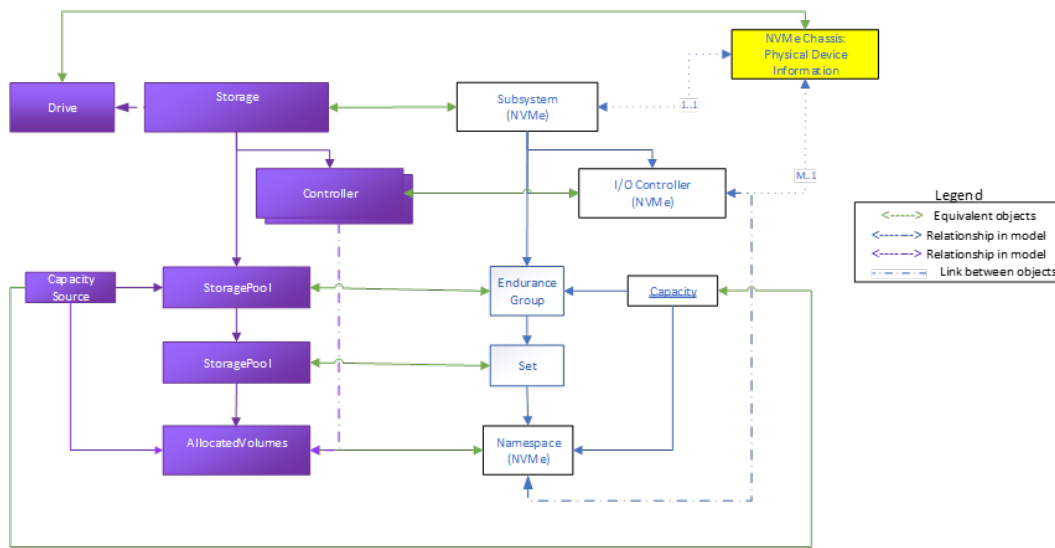
Figure 5: Complex SSD Model

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.
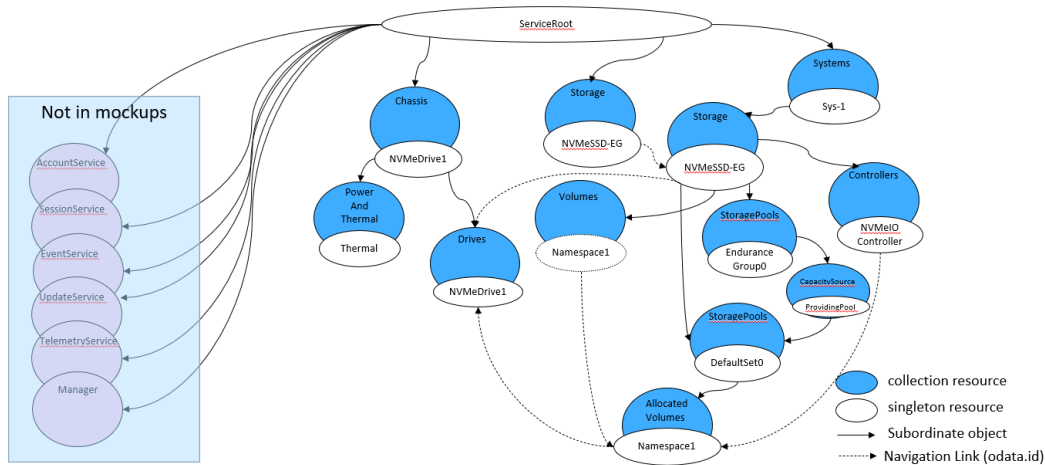
Figure 6: Complex SSD Model

5.3.3.1 Mockup A corresponding mockup for this configuration can be found at http://simple-ssd-eg-set-mockups.swordfishmockups.com/redfish/v1/.

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.
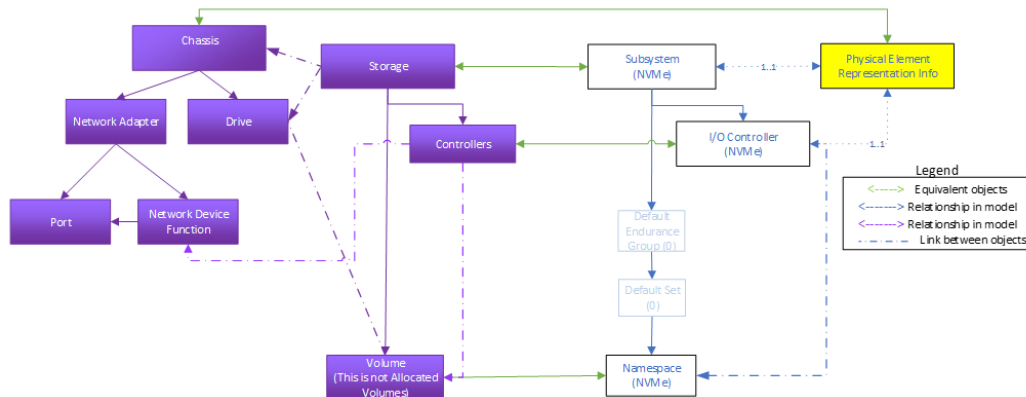
Figure 7: Simple IP-attached SSD

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.
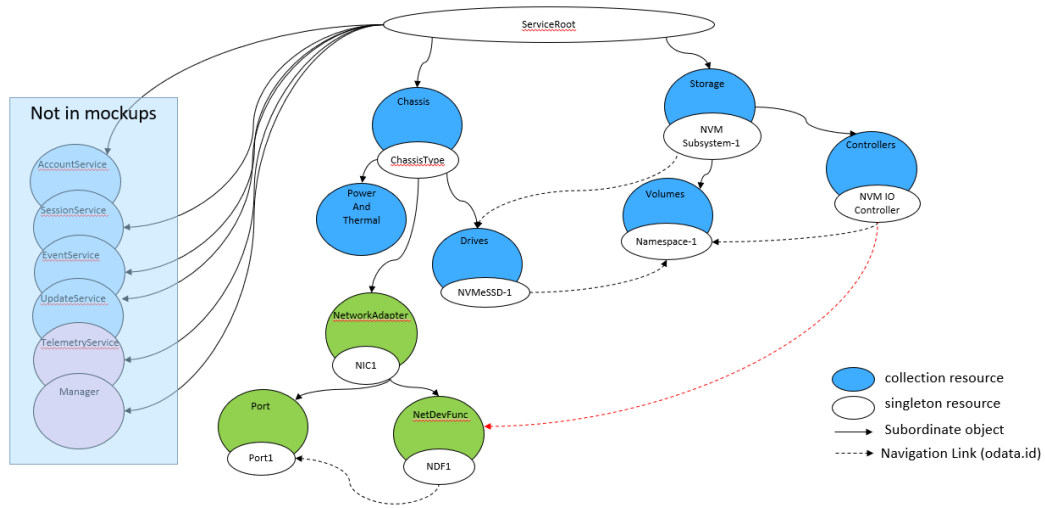
Figure 8: Simple IP-attached SSD mockup

5.4.4 Mockup

A corresponding mockup for this configuration can be found at http://ethernet-attach-drive-mockups.swordfishmockups.com/redfish/v1/.

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.
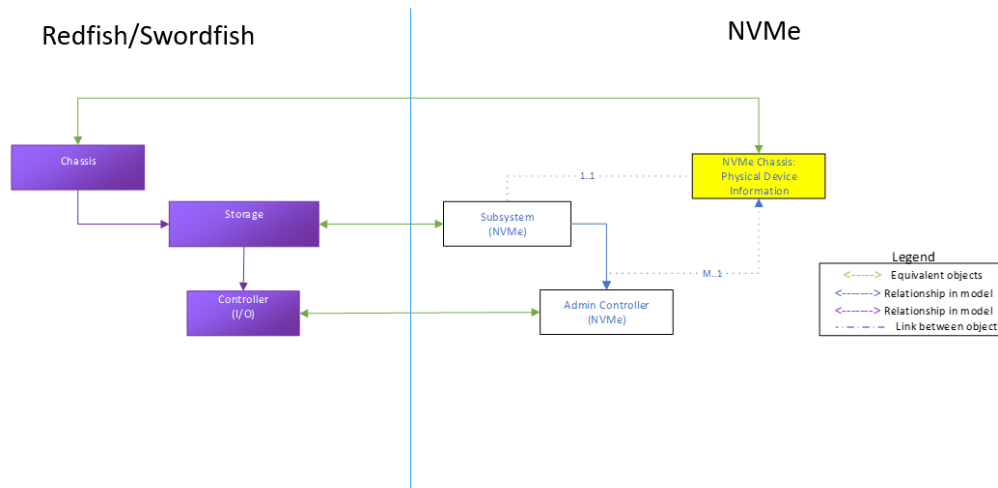
Figure 9: JBOF configuration controller object

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.
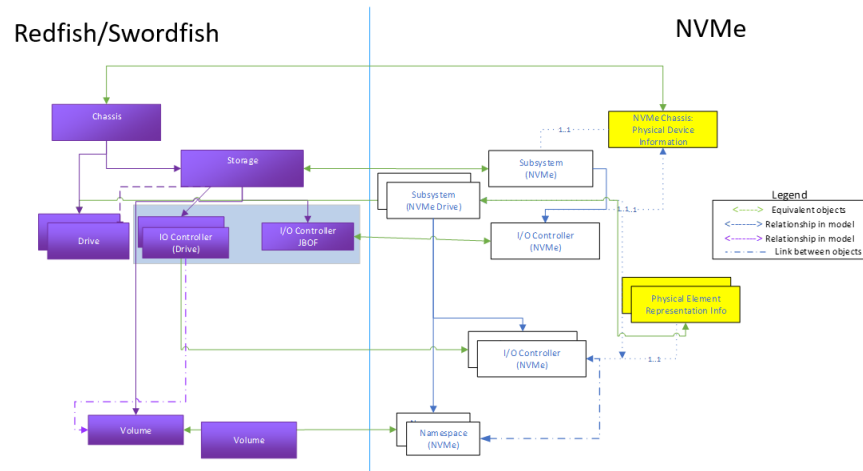
Figure 10: Full JBOF system

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://nvme-jbof-mockups.swordfishmockups.com/redfish/v1/
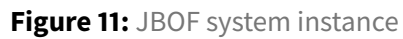
5.6 EBOF

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.
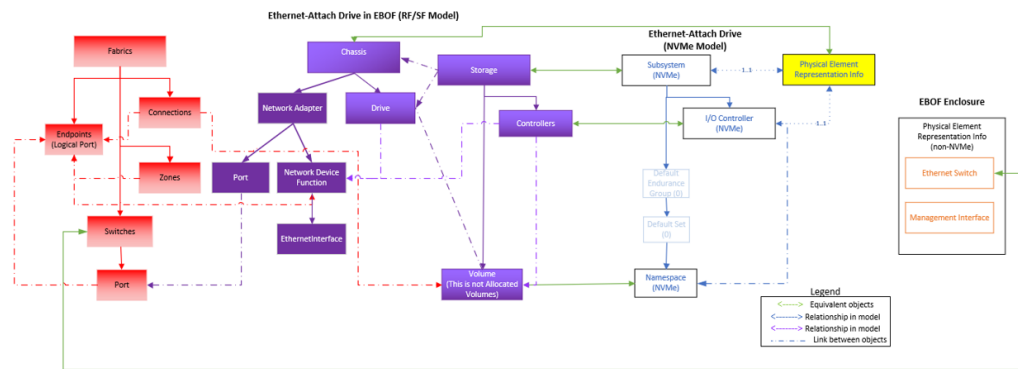
Figure 12: Full EBOF system

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.
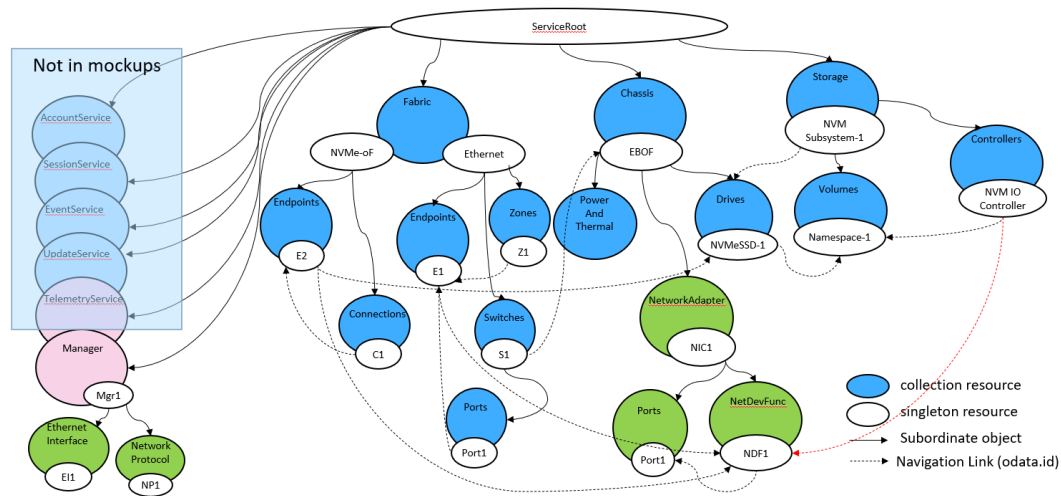
Figure 13: EBOF system instance

5.6.4 Mockup

A corresponding mockup for this configuration can be found at http://nvme-ebof-mockups.swordfishmockups.com/redfish/v1/

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.
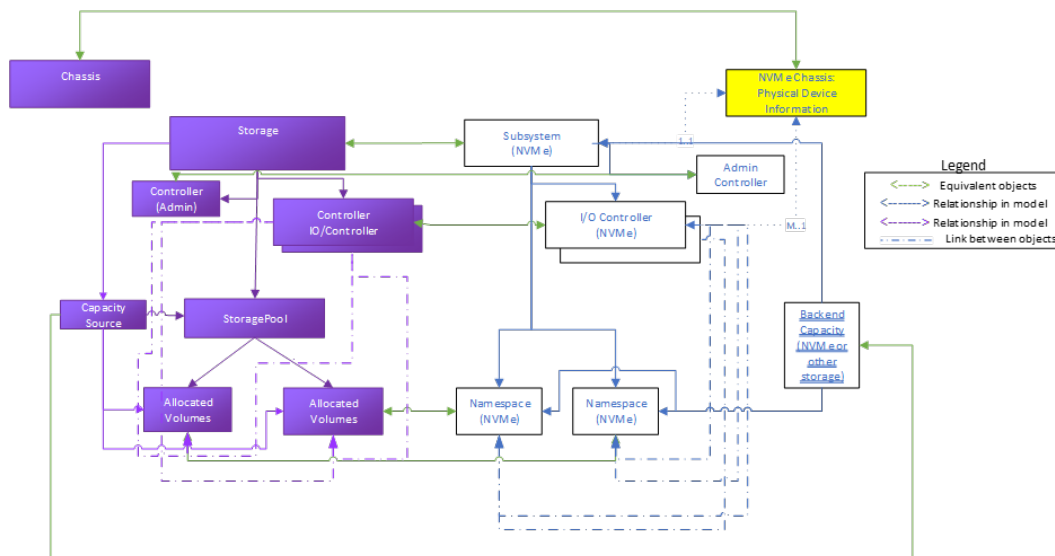
Figure 14: Opaque array example

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.
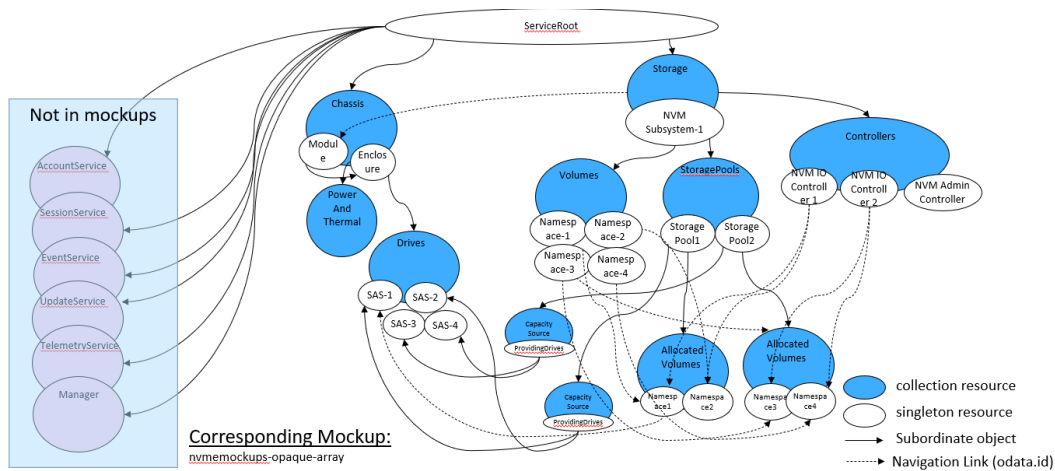
Figure 15: Sample opque system instance

5.7.4 Mockup

A corresponding mockup for this configuration can be found at http://nvme-opaque-array-mockups.swordfishmockups.com/redfish/v1/.

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.
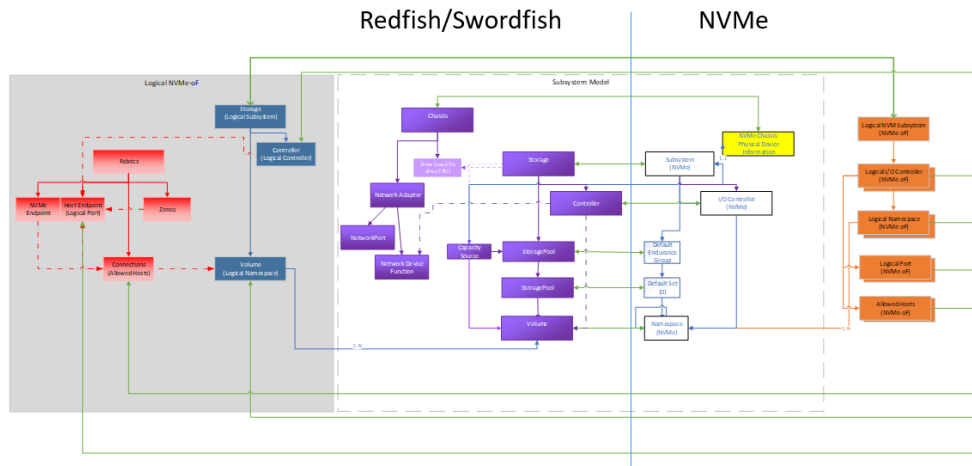
Figure 16: NVMe-OF subsystem example

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.
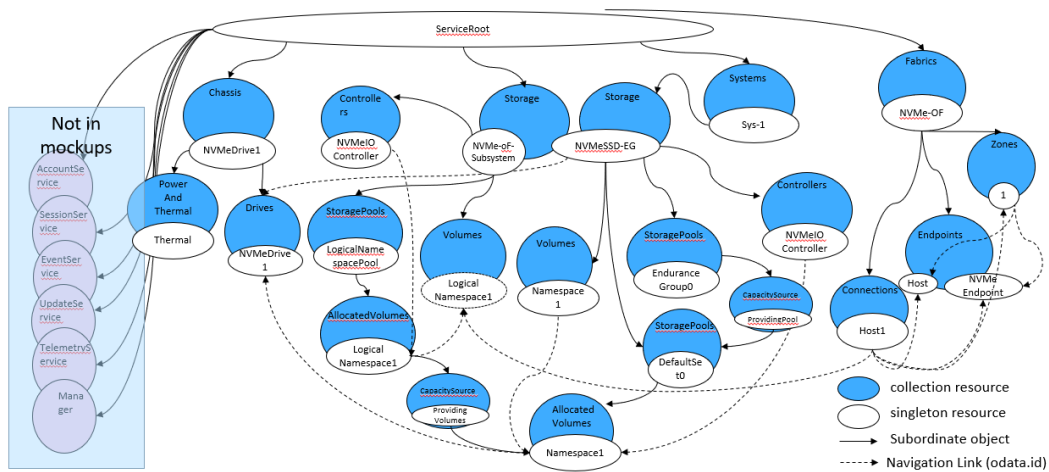
Figure 17: NVMe-oF system instance

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://nvmeof-mockups.swordfishmockups.com/redfish/v1/.

- A mockup for this configuration using an RDMA attach front-end can be found at http://nvmeof-RDMA-mockup.swordfishmockups.com/redfish/v1.

5.9 NVMe Domain Model

5.9.1 Overview

Figure 18 shows a representation of a simple domain representation, with a single IO controller and namespace in a domain.

> Note that as this configuration only has a single domain, it could be represented as an implicit domain.
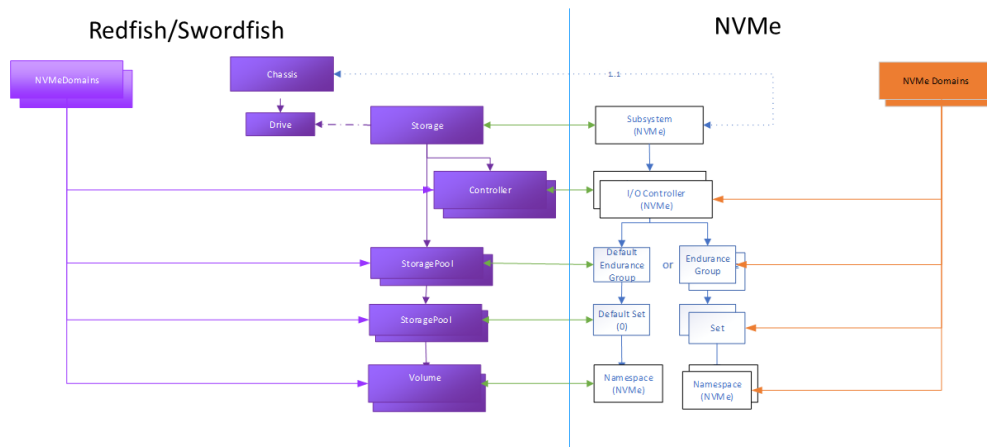
Figure 18: NVMe Domains example

5.9.2 Explanation of Object use

NVMeDomains contain a collection of domain members; these can be NVM controllers, endurance groups, NVM sets, or namespaces. Domains can constrain the set of firmware images available to apply to like members. Other domain-level properties include ALUA (asymmetric logical unit access) group definitions, indicating common access characteristics and path prioritization behaviors across the namespaces in the domain.

5.9.2.1 Implicit or Default Domains   In an NVM subsystem that does not support multiple domains, there is, by definition, a single domain. For single domain configurations, it is up to the Swordfish service implementer to determine whether or not to instrument the NVMe Domain objects.

5.9.3 Redfish / Swordfish Object Representation

Figure 19 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.
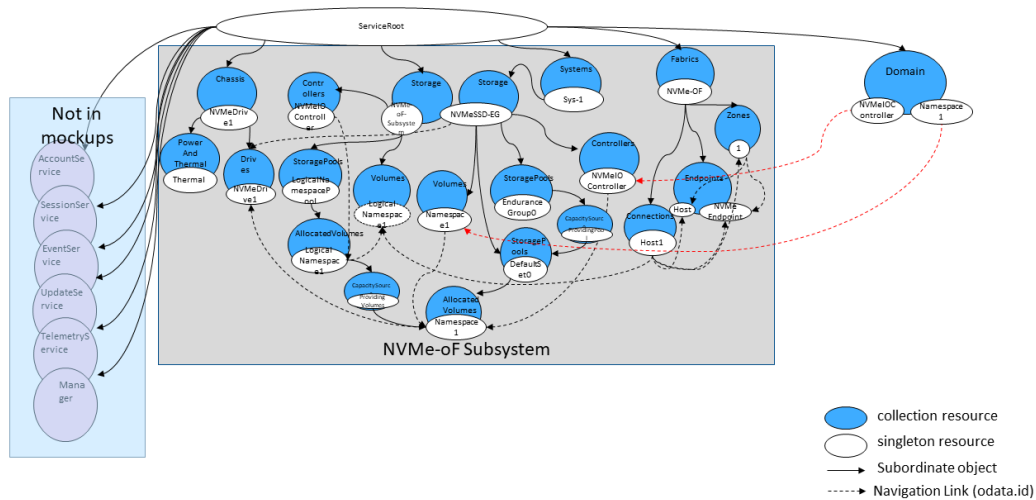
Figure 19: NVMe-oF system instance

> Note that as this configuration only has a single domain, it could be represented as an implicit domain.

5.9.4 Mockup

A mockup for this configuration using NVMe Domains can be found at http://nvme-domains-mockups.swordfishmockups.com/redfish/v1/.

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:** RecommendedArbitrationBurst Size	**NVM Spec Property / Field:** Identify Command / Identify Controller (CNS 01h) / RecommendedArbitrationBurst (RAB) **NVM Spec: Section:Figure** NVMe 2.0: Section 5.17.2.1, Figure 275 (Identify Controller Data Structure, I/O Command Set Independent) / RecommendedArbitrationBurst (RAB): Bytes 72
Type	**Redfish / Swordfish Schema Type:** String	Power of 2^n
Description	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.
LongDescription	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.	This is the recommended Arbitration Burst size. The value is in commands and is reported as a power of two (2^n). This is the same units as the Arbitration Burst size.

	Redfish/Swordfish	NVMe / NVMe-oF
Mandator y	Mandatory	I/O Controller: Mandatory Admin Controller: MandatoryDiscovery Controller: Restricted
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.

```
{
    "@odata.type": "#Storage.v1_16_0.Storage",
    "Id": "1",
    "Name": "NVMe Simple SSD Configuration",
    "Description": "An NVM Express Subsystem is an NVMe
     ↪   device that contains one or more NVM Express
     ↪   controllers and may contain one or more namespaces.",
    "Status": {
        "State": "Enabled",
        "Health": "OK",
        "HealthRollup": "OK"
    },
    "Identifiers": [
        {
            "DurableNameFormat": "NQN",
            "DurableName":
             ↪   "nqn.2014-08.org.nvmexpress:uuid:6c5fe566-
             ↪   10e6-4fb6-aad4-8b4159f50245"
        }
    ],
    "ConfigurationLock": "Partial",
    "TargetConfigurationLockLevel": "Standard",
    "NVMeSubsystemProperties": {
        "ConfigurationLockState": {
            "FirmwareCommit": "Unlocked",
            "Lockdown": "Locked",
            "SecureSend": "LockdownUnsupported",
```

```
            "FirmwareImageDownload": "LockdownUnsupported",
            "VPDWrite": "CommandUnsupported"
        },
        "MaxNamespacesSupported": 2,
        "SharedNamespaceControllerAttachmentSupported": true
    },
    "Controllers": {
        "@odata.id": "/redfish/v1/Systems/Sys-
        ↪   1/Storage/NVMeSSD-EG/Controllers"
    },
    "Drives": [
        {
            "@odata.id": "/redfish/v1/Chassis/Sys-
            ↪   1Chassis/Drives/NVMeDrive1"
        }
    ],
    "Volumes": {
        "@odata.id": "/redfish/v1/Systems/Sys-
        ↪   1/Storage/NVMeSSD-EG/Volumes"
    },
    "StoragePools": {
        "@odata.id": "/redfish/v1/Systems/Sys-
        ↪   1/Storage/NVMeSSD-EG/StoragePools"
    },
    "Links": {
        "Enclosures": [
            {
                "@odata.id":
                ↪   "/redfish/v1/Chassis/Sys-1Chassis"
            }
        ]
    },
    "@odata.id":
    ↪   "/redfish/v1/Systems/Sys-1/Storage/NVMeSSD-EG",
```

```
    "@Redfish.Copyright": "Copyright 2015-2024 SNIA. All
    ↪   rights reserved."
}
```

6.3.2 Property Mapping

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

Table 5: Actions.#StorageController.SetEncryptionKey mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Actions.#StorageController .SetEncryptionKey	N/A
Type	Action (Special form of POST)	N/A
Descript ion	The available OEM-specific actions for this resource.	N/A
LongDesc ription	This property shall contain the available OEM-specific actions for this resource.	N/A
Mandator y	Optional Do Not Implement for NVMe Drives, or for devices with an NVMe front end interface, e.g., opaque arrays.	DNI
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 ConfigurationLock   The mapping for `ConfigurationLock` is summarized in Table 6.

Table 6: ConfigurationLock mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	ConfigurationLock	**NVM Spec: Property / Field:**NVMe 2.0d: Identify Command / Identify Controller data structure (CNS 01h) - Optional Admin Command Support (OACS)**NVM Spec: Section:Figure**NVMe 2.0d: Section 5.17.2.1: Figure 275 - 257:256, Bit 10) Get log page specifying the Command and Feature Lockdown (Log Page Identifier 14h) **NVM Spec: Section:Figure**NVMe 2.0d: Section 5.16.1.20
Type	enum (ConfigurationLock)	Log Page

	Redfish/Swordfish	NVMe / NVMe-oF
Description	The `ConfigurationLock`, `TargetConfigura-tionLockLev el`, and `ConfigurationLockState` properties are jointly used to manage lockdown of NVMe devices.	The Lockdown command is used to control the Command and Feature Lockdown capability which configures the prohibition or allowance of execution of the specified command or Set Features command targeting a specific Feature Identifier. After a successful completion of a Lockdown command prohibiting a command or Feature Identifier, all controllers, if applicable, and all management endpoints, if applicable, in the NVM subsystem behave as requested.

	Redfish/Swordfish	NVMe / NVMe-oF
LongDesc ription	This has three states: Enabled, Disabled, and Partial. • `Enabled`: All supported properties, as reported in the `ConfigurationLockState`, that are part of the `Target-ConfigurationLockLev el` are locked. • `Disabled`: All supported properties, as reported in the `ConfigurationLockState`, that are part of the `Target-ConfigurationLockLev el` are unlocked.• `Partial`: The properties, as reported in the `ConfigurationLockState`, that are part of the `Target-ConfigurationLockLev el` are not consistently locked or unlocked. Services shall reject modification requests that contain the value `Partial`.	Command and Feature Lockdown is used to prohibit the execution of commands submitted to NVM Express controllers and/or Management Endpoints in an NVM subsystem. Commands and Feature Identifiers are defined with the following scopes: • Admin Command Set commands defined by the Opcode field;• Set Features command features defined by the Feature Identifier field;• Management Interface Command Set commands defined by the Opcode field; and• PCIe Command Set commands defined by the Opcode field (refer to the NVM Express Management Interface Specification).
Mandator y	Optional	Mandatory to implement OACS; Lockdown feature is optional.
Notes	See section `Mapping NVMe Command and F eature Lockdown to "Configu rationLocked"` for detailed behavior and mapping description.	

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

Table 7: Controllers mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Controllers	**NVM Spec Property / Field:** Identify Command / Controller List (CNS 13h)**NVM Spec: Section:Figure** NVMe 2.0: Section 5.17.2.12
Type	StorageControllerCollection .	Controller list.
Descript ion	The set of controllers instantiated by this storage subsystem.	A List of controllers that exist in the NVM subsystem.
LongDesc ription	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 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). Reference Section 4.4.1, Figure 134 (Controller List Format)
Mandator y	Mandatory.	No (see note)

	Redfish/Swordfish	NVMe / NVMe-oF
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.

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

Table 8: Description mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Description	N/A
Type	String	N/A
Descript ion	The description of this resource.	N/A
LongDesc ription	This object represents the description of this resource. The resource values shall comply with the Redfish Specification-described requirements.	N/A
Mandator y	Mandatory	
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."

6.3.2.5 Drives   The mapping for `Drives` is summarized in Table 9.

Table 9: Drives mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Drives	N/A
Type	Collection(Drive.Drive)	N/A
Descript ion	The set of drives attached to the storage controllers that this resource represents.	N/A
LongDesc ription	This property shall contain a set of the drives attached to the storage controllers that this resource represents.	N/A
Mandator y	Optional Mandatory for NVMe Drives.	
Notes	For NVMe Drive implementation, this links to "Drive" object, which contains the physical representation of NVMe Drive information.	

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

Table 10: Identifiers mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Identifiers	N/A
Type	Collection(Resource.Identif ier)	N/A
Descript ion	The Durable names for the subsystem.	N/A
LongDesc ription	This property shall contain a list of all known durable names for the associated subsystem.	N/A
Mandator y	Mandatory	
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 275 (Identify – Identify Controller Data Structure, I/O Command Set Independent) of the NVMe Base Specification.	

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

Table 11: Identifiers.DurableNameFormat mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Identifiers.DurableNameForm at	**NVM Spec Property / Field:** NVMe Qualified Names **NVM Spec: Section:Figure** NVMe 2.0: Section 4.5
Type	Resource.v1_1_0.DurableNa meFormat	UTF-8 null-terminated string
Descript ion	The format of the Durable names for the subsystem.	This specifies the NVM Subsystem NVMe Qualified Name as a UTF-8 null-terminated string.
LongDesc ription	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).	This specifies the NVM Subsystem NVMe Qualified Name as a UTF-8 null-terminated string.
Mandator y	Optional	Yes if the controller supports revision 1.2.1 or later as indicated in the Version register.
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 which is a UTF-8 null-terminated string.	

6.3.2.8 Identifiers.DurableName   The mapping for `Identifiers.DurableName` is summarized in Table 12.

Table 12: Identifiers.DurableName mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Identifiers.DurableName	**NVM Spec Property / Field:** Identify Command / Identify Controller (CNS 01h)**NVM Spec: Section:Figure** NVMe 2.0: Section 5.17.2.1, Figure 275 (Identify Controller Data Structure, I/O Command Set Independent), Bits 768-1023 - NVM Subsystem NVMe Qualified Name (SUBNQN)
Type	Edm.String	UTF-8 null-terminated string
Descript ion	The format of the Durable names for the subsystem.	This field specifies the NVM Subsystem NVMe Qualified Name (SUBNQN)
LongDesc ription	This specifies the NVM Subsystem NVMe Qualified Name as a UTF-8 null-terminated string.	This field specifies the NVM Subsystem NVMe Qualified Name as a UTF-8 null-terminated string. Refer to NVMe Base Specification, section 4.5, for the definition of NVMe Qualified Name.
Mandator y	Optional	Yes if the controller supports revision 1.2.1 or later as indicated in the Version register (refer to section 3.1.3.2).

	Redfish/Swordfish	NVMe / NVMe-oF
Notes	For this particular usage in Subsystem, there will only be one instance populated in the identifiers array.	

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

Table 13: Links.Enclosures mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Links.Enclosures	N/A
Type	Collection(Chassis.Chassis)	N/A
Descript ion	An array of links to the chassis to which this storage subsystem is attached.	N/A
LongDesc ription	This property shall contain an array of links to resources of type Chassis that represent the physical containers attached to this resource.	N/A
Mandator y	Mandatory	
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

6.3.2.10 Links.Enclosures@odata.count   The mapping for `Links.Enclosures@odata.count` is summarized in Table 14.

Table 14: Links.Enclosures@odata.count mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Links.Enclosures@odata.cou ntt	N/A
Type	(odata property)int64	N/A
Descript ion	Count of the number of items in the Links.Enclosures array.	N/A
LongDesc ription		N/A
Mandator y	Mandatory	
Notes		The functionality comes from the underlying implementation and does not originate in the NVMe specs

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

Table 15: Links.HostingStorageSystems mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Links.HostingStorageSystems	N/A
Type	Collection(ComputerSystem.ComputerSystem)	N/A
Descript ion	The storage systems that host this storage subsystem.	N/A
LongDesc ription	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
Mandator y	Optional Recommended for devices with an NVMe front end interface such as opaque arrays.	
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

6.3.2.12 Links.NVMeoFDiscoverySubsystems   The mapping for `Links.NVMeoFDiscoverySubsystems` is summarized in Table 16.

Table 16: Links.NVMeoFDiscoverySubsystems mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Links.NVMeoFDiscoverySubsystems	N/A
Type	Collection(Storage.Storage)	N/A
Descript ion	An array of links to the discovery subsystems that discovered this subsystem in an NVMe-oF environment.	N/A
LongDesc ription	This property shall contain an array of links to resources of type Storage that represent the discovery subsystems that discovered this subsystem in an NVMe-oF environment.	N/A
Mandator y	Optional Recommended for devices with an NVMe front end interface such as opaque arrays; required for use in NVMe-oF environments and environments using discovery controllers.	
Notes	This information should be constructed from the Swordfish service. There is not a known mapping to NVMe-oF information at this time.	

6.3.2.13 Links.SimpleStorage The mapping for `Links.SimpleStorage` is summarized in Table 17.

Table 17: Links.SimpleStorage mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Links.SimpleStorage	N/A
Type	SimpleStorage.SimpleStorage	N/A
Descript ion	The link to the simple storage instance that corresponds to this storage.	N/A
LongDesc ription	This property shall contain a link to a resource of type SimpleStorage that represents the same storage subsystem as this resource.	N/A
Mandator y	Do Not Implement	
Notes		

6.3.2.14 NVMeSubsystemProperties.NVMeConfigurationLockState The map-
ping for `NVMeSubsystemProperties.NVMeConfigurationLockState` is
summarized in Table 18.

Table 18: NVMeSubsystemProperties.NVMeConfigurationLockState mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeSubsystemProperties.NVMeConfigurationLockState	N/A (Sub-properties map to NVMe Command and Feature Lockdown)
Type	ComplexType	N/A
Descript ion	Set of configurable features that are able to be locked on an NVMe Subsystem, and their current lock state.	
LongDesc ription		
Mandator y	Optional	
Notes	See section `Mapping NVMe Command and F eature Lockdown` to `"Configu rationLocked"` for detailed behavior and mapping description.	

6.3.2.15 NVMeSubsystemProperties.NVMeConfigurationLockState.FirmwareCommit

The mapping for NVMeSubsystemProperties.NVMeConfigurationLockState.FirmwareCommit is summarized in Table 19.

Table 19: NVMeSubsystemProperties.NVMeConfigurationLockState.FirmwareCommit mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeSubsystemProperties.NVMeConfigurationLockState.FirmwareCommit	**NVM Spec: Property / Field:**NVMe 2.0d: Identify Command / Identify Controller data structure (CNS 01h) - Optional Admin Command Support (OACS)**NVM Spec: Section:Figure**NVMe 2.0d: Section 5.17.2.1: Figure 275 - 257:256, Bytes 13:08) Get log page specifying the Command and Feature Lockdown (Log Page Identifier 14h) **NVM Spec: Section:Figure**NVMe 2.0d: Section 5.16.1.20 (Figure 260)

	Redfish/Swordfish	NVMe / NVMe-oF
Type	Storage.v1_17_0.ConfigLoc kOptions • `Unlocked`: Command is supported, able to be locked, and the current state is unlocked. • `Locked`: Command is supported, able to be locked, and the current state is locked. • `LockdownUnsupported`: Command is supported, able to be locked, and the current state is locked. • `CommandUnsupported`: Command is not supported, therefore lockdown does not apply.	Log Page
Descript ion	Ability to perform the NVMe defined FirmwareCommit command.	If the FirmwareCommit feature is able to be locked down, it's corresponding opcode, 10h, will be reported in the command and feature lockdown log page (bytes 13:08).

	Redfish/Swordfish	NVMe / NVMe-oF
LongDesc ription	This has three states: Enabled, Disabled, and Partial. • `Enabled`: All supported properties, as reported in the `ConfigurationLockState`, that are part of the `Target-ConfigurationLockLev el` are locked. • `Disabled`: All supported properties, as reported in the `ConfigurationLockState`, that are part of the `Target-ConfigurationLockLev el` are unlocked.• `Partial`: The properties, as reported in the `ConfigurationLockState`, that are part of the `Target-ConfigurationLockLev el` are not consistently locked or unlocked. Services shall reject modification requests that contain the value `Partial`.	Command and Feature Lockdown is used to prohibit the execution of commands submitted to NVM Express controllers and/or Management Endpoints in an NVM subsystem. Commands and Feature Identifiers are defined with the following scopes: • Admin Command Set commands defined by the Opcode field;• Set Features command features defined by the Feature Identifier field;• Management Interface Command Set commands defined by the Opcode field; and• PCIe Command Set commands defined by the Opcode field (refer to the NVM Express Management Interface Specification).
Mandator y	Optional	Mandatory to implement OACS; FirmwareCommit feature lockdown support is optional. If able to be supported (in log page 14), use the lockdown command to lock and unlock correspond to client requests.

	Redfish/Swordfish	NVMe / NVMe-oF
Notes	See section `Mapping NVMe Command and F eature Lockdown to "Configu rationLocked"` for detailed behavior and mapping description.	

6.3.2.16 NVMeSubsystemProperties.NVMeConfigurationLockState.FirmwareImageDownload

The mapping for NVMeSubsystemProperties.NVMeConfigurationLockState.FirmwareImageDo
is summarized in Table 20.

Table 20:

NVMeSubsystemProperties.NVMeConfigurationLockState.FirmwareImageDownload
mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeSubsystemProperties.NVMeConfigurationLockState.FirmwareImageDownload	**NVM Spec: Property / Field:**NVMe 2.0d: Lockdown - **NVM Spec: Section:Figure**NVMe 2.0d: Section 5.19: Figure 292 - 06:00) Get log page specifying and controlling the Command and Feature Lockdown (Log Page Identifier 14h) **NVM Spec: Section:Figure**NVMe 2.0d: Section 5.16.1.20 (Figure 260), bytes 13:08 Log page indicating which commands and features are supported and currently set.

	Redfish/Swordfish	NVMe / NVMe-oF
Type	Storage.v1_17_0.ConfigLoc kOptions • `Unlocked:` Command is supported, able to be locked, and the current state is unlocked. • `Locked:` Command is supported, able to be locked, and the current state is locked. • `LockdownUnsupported:` Command is supported, able to be locked, and the current state is locked. • `CommandUnsupported:` Command is not supported, therefore lockdown does not apply.	Log Page
Descript ion	Ability to perform the NVMe defined FirmwareImageDownload command.	If the FirmwareImageDownload feature is able to be locked down, it's corresponding opcode, 11h, will be reported in the command and feature lockdown log page (bytes 13:08).

	Redfish/Swordfish	NVMe / NVMe-oF
LongDescription	This has three states: Enabled, Disabled, and Partial. • `Enabled`: All supported properties, as reported in the `ConfigurationLockState`, that are part of the `Target-ConfigurationLockLev el` are locked. • `Disabled`: All supported properties, as reported in the `ConfigurationLockState`, that are part of the `Target-ConfigurationLockLev el` are unlocked.• `Partial`: The properties, as reported in the `ConfigurationLockState`, that are part of the `Target-ConfigurationLockLev el` are not consistently locked or unlocked. Services shall reject modification requests that contain the value `Partial`.	Command and Feature Lockdown is used to prohibit the execution of commands submitted to NVM Express controllers and/or Management Endpoints in an NVM subsystem. Commands and Feature Identifiers are defined with the following scopes: • Admin Command Set commands defined by the Opcode field;• Set Features command features defined by the Feature Identifier field;• Management Interface Command Set commands defined by the Opcode field; and• PCIe Command Set commands defined by the Opcode field (refer to the NVM Express Management Interface Specification).
Mandatory	Optional	Mandatory to implement OACS; FirmwareImageDownload feature lockdown support is optional. If able to be supported (in log page 14), use the lockdown command to lock and unlock correspond to client requests.

	Redfish/Swordfish	NVMe / NVMe-oF
Notes	See section `Mapping NVMe Command and F eature Lockdown to "Configu rationLocked"` for detailed behavior and mapping description.	

6.3.2.17 NVMeSubsystemProperties.NVMeConfigurationLockState.Lockdown

The mapping for NVMeSubsystemProperties.NVMeConfigurationLockState.Lockdown is summarized in Table 21.

Table 21: NVMeSubsystemProperties.NVMeConfigurationLockState.Lockdown mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeSubsystemProperties.NVMeConfigurationLockState.Lockdown	**NVM Spec: Property / Field:**NVMe 2.0d: Identify Command / Identify Controller data structure (CNS 01h) - Optional Admin Command Support (OACS)**NVM Spec: Section:Figure**NVMe 2.0d: Section 5.17.2.1: Figure 275 - 257:256, Bit 10) Get log page specifying the Command and Feature Lockdown (Log Page Identifier 14h) **NVM Spec: Section:Figure**NVMe 2.0d: Section 5.16.1.20

	Redfish/Swordfish	NVMe / NVMe-oF
Type	Storage.v1_17_0.ConfigLoc kOptions • `Unlocked`: Command is supported, able to be locked, and the current state is unlocked. • `Locked`: Command is supported, able to be locked, and the current state is locked. • `LockdownUnsupported`: Command is supported, able to be locked, and the current state is locked. • `CommandUnsupported`: Command is not supported, therefore lockdown does not apply.	Log Page
Descript ion	Ability to perform the NVMe defined Lockdown command.	The Lockdown command is used to control the Command and Feature Lockdown capability which configures the prohibition or allowance of execution of the specified command or Set Features command targeting a specific Feature Identifier. After a successful completion of a Lockdown command prohibiting a command or Feature Identifier, all controllers, if applicable, and all management endpoints, if applicable, in the NVM subsystem behave as requested.

	Redfish/Swordfish	NVMe / NVMe-oF
LongDesc ription	This has three states: Enabled, Disabled, and Partial. • `Enabled`: All supported properties, as reported in the `ConfigurationLockState`, that are part of the `Target-ConfigurationLockLev el` are locked. • `Disabled`: All supported properties, as reported in the `ConfigurationLockState`, that are part of the `Target-ConfigurationLockLev el` are unlocked.• `Partial`: The properties, as reported in the `ConfigurationLockState`, that are part of the `Target-ConfigurationLockLev el` are not consistently locked or unlocked. Services shall reject modification requests that contain the value `Partial`.	Command and Feature Lockdown is used to prohibit the execution of commands submitted to NVM Express controllers and/or Management Endpoints in an NVM subsystem. Commands and Feature Identifiers are defined with the following scopes: • Admin Command Set commands defined by the Opcode field;• Set Features command features defined by the Feature Identifier field;• Management Interface Command Set commands defined by the Opcode field; and• PCIe Command Set commands defined by the Opcode field (refer to the NVM Express Management Interface Specification).
Mandator y	Optional	Mandatory to implement OACS; Lockdown feature is optional. If able to be supported (in log page 14), use the lockdown command to lock and unlock correspond to client requests.

	Redfish/Swordfish	NVMe / NVMe-oF
Notes	See section `Mapping NVMe Command and F eature Lockdown to "Configu rationLocked"` for detailed behavior and mapping description.	

6.3.2.18 NVMeSubsystemProperties.NVMeConfigurationLockState.SecuritySend

The mapping for NVMeSubsystemProperties.NVMeConfigurationLockState.SecuritySend is summarized in Table 22.

Table 22: NVMeSubsystemProperties.NVMeConfigurationLockState.SecuritySend mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeSubsystemProperties.NVMeConfigurationLockState.SecuritySend	**NVM Spec: Property / Field:**NVMe 2.0d: Identify Command / Identify Controller data structure (CNS 01h) - Optional Admin Command Support (OACS)**NVM Spec: Section:Figure**NVMe 2.0d: Section 5.17.2.1: Figure 275 - 257:256, Bytes 13:08) Get log page specifying the Command and Feature Lockdown (Log Page Identifier 14h) **NVM Spec: Section:Figure**NVMe 2.0d: Section 5.16.1.20 (Figure 260)

	Redfish/Swordfish	NVMe / NVMe-oF
Type	Storage.v1_17_0.ConfigLoc kOptions • `Unlocked`: Command is supported, able to be locked, and the current state is unlocked. • `Locked`: Command is supported, able to be locked, and the current state is locked. • `LockdownUnsupported`: Command is supported, able to be locked, and the current state is locked. • `CommandUnsupported`: Command is not supported, therefore lockdown does not apply.	Log Page
Descript ion	Ability to perform the NVMe defined SecuritySend command.	If the SecuritySend feature is able to be locked down, it's corresponding opcode, 81h, will be reported in the command and feature lockdown log page (bytes 13:08).

	Redfish/Swordfish	NVMe / NVMe-oF
LongDesc ription	This has three states: Enabled, Disabled, and Partial. • `Enabled`: All supported properties, as reported in the `ConfigurationLockState`, that are part of the `Target- ConfigurationLockLev el` are locked. • `Disabled`: All supported properties, as reported in the `ConfigurationLockState`, that are part of the `Target- ConfigurationLockLev el` are unlocked.• `Partial`: The properties, as reported in the `ConfigurationLockState`, that are part of the `Target- ConfigurationLockLev el` are not consistently locked or unlocked. Services shall reject modification requests that contain the value `Partial`.	Command and Feature Lockdown is used to prohibit the execution of commands submitted to NVM Express controllers and/or Management Endpoints in an NVM subsystem. Commands and Feature Identifiers are defined with the following scopes: • Admin Command Set commands defined by the Opcode field;• Set Features command features defined by the Feature Identifier field;• Management Interface Command Set commands defined by the Opcode field; and• PCIe Command Set commands defined by the Opcode field (refer to the NVM Express Management Interface Specification).
Mandator y	Optional	Mandatory to implement OACS; SecuritySend feature lockdown support is optional. If able to be supported (in log page 14), use the lockdown command to lock and unlock correspond to client requests.

	Redfish/Swordfish	NVMe / NVMe-oF
Notes	See section `Mapping NVMe Command and F eature Lockdown to "Configu rationLocked"` for detailed behavior and mapping description.	

6.3.2.19 NVMeSubsystemProperties.NVMeConfigurationLockState.VPDWrite   The
mapping for NVMeSubsystemProperties.NVMeConfigurationLockState.VPDWrite
is summarized in Table 23.

Table 23: NVMeSubsystemProperties.NVMeConfigurationLockState.VPDWrite mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeSubsystemProperties.NVMeConfigurationLockState.VPDWrite	**NVM Spec: Property / Field:**NVMe 2.0d: Identify Command / Identify Controller data structure (CNS 01h) - Optional Admin Command Support (OACS)**NVM Spec: Section:Figure**NVMe 2.0d: Section 5.17.2.1: Figure 275 - 257:256) Get log page specifying the Command and Feature Lockdown (Log Page Identifier 14h) **NVM Spec: Section:Figure**NVMe 2.0d: Section 5.16.1.20 (Figure 260)

	Redfish/Swordfish	NVMe / NVMe-oF
Type	Storage.v1_17_0.ConfigLoc kOptions • `Unlocked:` Command is supported, able to be locked, and the current state is unlocked. • `Locked:` Command is supported, able to be locked, and the current state is locked. • `LockdownUnsupported:` Command is supported, able to be locked, and the current state is locked. • `CommandUnsupported:` Command is not supported, therefore lockdown does not apply.	Log Page
Descript ion	Ability to perform the NVMe defined VPDWrite command.	If the VPDWrite feature is able to be locked down, it's corresponding opcode, 81h, will be reported in the command and feature lockdown log page (bytes 13:08).

	Redfish/Swordfish	NVMe / NVMe-oF
LongDesc ription	This has three states: Enabled, Disabled, and Partial. • `Enabled`: All supported properties, as reported in the `ConfigurationLockState`, that are part of the `Target-ConfigurationLockLev el` are locked. • `Disabled`: All supported properties, as reported in the `ConfigurationLockState`, that are part of the `Target-ConfigurationLockLev el` are unlocked.• `Partial`: The properties, as reported in the `ConfigurationLockState`, that are part of the `Target-ConfigurationLockLev el` are not consistently locked or unlocked. Services shall reject modification requests that contain the value `Partial`.	Command and Feature Lockdown is used to prohibit the execution of commands submitted to NVM Express controllers and/or Management Endpoints in an NVM subsystem. Commands and Feature Identifiers are defined with the following scopes: • Admin Command Set commands defined by the Opcode field;• Set Features command features defined by the Feature Identifier field;• Management Interface Command Set commands defined by the Opcode field; and• PCIe Command Set commands defined by the Opcode field (refer to the NVM Express Management Interface Specification).
Mandator y	Optional	Mandatory to implement OACS; VPDWrite feature lockdown support is optional. If able to be supported (in log page 14), use the lockdown command to lock and unlock correspond to client requests.

	Redfish/Swordfish	NVMe / NVMe-oF
Notes	See section `Mapping NVMe Command and F eature Lockdown to "Configu rationLocked"` for detailed behavior and mapping description.	

6.3.2.20 NVMeSubsystemProperties.MaxNamespacesSupported   The mapping for NVMeSubsystemProperties.MaxNamespacesSupported is summarized in Table 24.

Table 24: NVMeSubsystemProperties.MaxNamespacesSupported mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeSubsystemProperties.Max NamespacesSupported	**NVM Spec Property / Field:** Identify Controller Data Structure (CNS 01h)**NVM Spec: Section:Figure** NVMe 2.0: Section 5.17.2.1, Figure 276 (Identify Controller Data Structure, I/O Command Set Independent), Bytes 543:540 - Maximum Number of Allowed Namespaces (MNAN)
Type	Edm.Decimal	Integer
Descript ion	The maximum number of namespace attachments supported by this NVMe Subsystem.	This field indicates the maximum number of namespaces supported by the NVM subsystem.

	Redfish/Swordfish	NVMe / NVMe-oF
LongDescription	This property shall contain the maximum number of namespace attachments supported by this NVMe Subsystem. If there are no limits imposed, this property should not be implemented.	This field indicates the maximum number of namespaces supported by the NVM subsystem. If this field is cleared to 0h, then the maximum number of namespaces supported by the NVM subsystem is less than or equal to the value in the NN field. If the controller supports Asymmetric Namespace Access Reporting, then this field shall be set to a non-zero value that is less than or equal to the NN value.
Mandatory	Optional Used when Namespace Management is implemented.	Optional
Notes	If MNAN is 0h, the NVMeSubsystemProperties.MaxNamespacesSupported property should not be implemented.	

6.3.2.21 NVMeSubsystemProperties.SharedNamespaceControllerAttachmentSupported

The mapping for NVMeSubsystemProperties.SharedNamespaceControllerAttachmentSupport
is summarized in Table 25.

Table 25:

NVMeSubsystemProperties.SharedNamespaceControllerAttachmentSupported
mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeSubsystemProperties.SharedNamespaceControllerAttachmentSupported	**NVM Spec Property / Field:** I/O Command Set Independent Identify Namespace data structure (CNS 08h)**NVM Spec: Section:Figure** NVMe 2.0: Section 5.17.2.8, Figure 281 (Identify Controller Data Structure, I/O Command Set Independent), Byte 01, bit 0 - Namespace Multi-path I/O and Namespace Sharing Capabilities (NMIC)
Type	Edm.Boolean	Bool
Descript ion	Indicates whether the subsystem supports shared namespace controller attachment.	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). If cleared to "0", then the namespace is a private namespace and is able to be attached to only one controller at a time.

	Redfish/Swordfish	NVMe / NVMe-oF
LongDescription	This property shall indicate whether the subsystem supports shared namespace controller attachment. A shared namespace may be attached to two or more controllers in a Subsystem concurrently.	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). If cleared to "0", then the namespace is a private namespace and is able to be attached to only one controller at a time.
Mandator y Notes	Optional Used when Namespace Management is implemented.	Optional

6.3.2.22 Name   The mapping for Name is summarized in Table 26.

Table 26: Name mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Name	N/A
Type	String	N/A
Descript ion	The name of the resource or array member.	N/A
LongDesc ription	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.	N/A
Mandator y	Mandatory	
Notes	In Redfish, Name is a read-only field.	

6.3.2.23 Status.State   The mapping for `Status.State` is summarized in Table 27.

Table 27: Status.State mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Status.State	N/A
Type	Resource.State (enum)	N/A
Descript ion	The known state of the resource, such as, enabled.	N/A
LongDesc ription	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.	N/A
Mandator y	Optional Recommended to implement for NVMe front end devices such as opaque arrays.	

	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.

6.3.2.24 Status.Health The mapping for `Status.Health` is summarized in Table 28.

Table 28: Status.Health mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Status.Health	**NVM Spec Property / Field:** Smart / Health Information Log (Log Identifier 02h) / Critical Warning / the NVM subsystem reliability has been degraded due to significant media related errors or any internal error that degrades NVM subsystem reliability**NVM Spec: Section:Figure** NVMe 2.0: Section 5.16.1.3, Figure 207 (Get Log Page – SMART / Health Information Log), Byte 00 (Critical Warning), Bit 02
Type	Resource.Health	8 Bit value
Descript ion	The health state of this resource in the absence of its dependent resources.	Indicates if the NVM subsystem reliability has been degraded due to significant media related errors or any internal error that degrades NVM subsystem reliability
LongDesc ription	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.	Indicates if the NVM subsystem reliability has been degraded due to significant media related errors or any internal error that degrades NVM subsystem reliability
Mandator y	Mandatory	Yes

	Redfish/Swordfish	NVMe / NVMe-oF
Notes	Possible Values: OK / Warning / Critical	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.

6.3.2.25 Status.HealthRollup The mapping for `Status.HealthRollup` is summarized in Table 29.

Table 29: Status.HealthRollup mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Status.HealthRollup	N/A
Type	Resource.Health	N/A
Descript ion	The overall health state from the view of this resource.	N/A
LongDesc ription	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
Mandator y	Mandatory	
Notes	Possible Values: OK / Warning / Critical See [Swordfish	
Specific ation](#Normat ive_Ref erences) for more informat ion regardin g Status.H ealthRol lup		

6.3.2.26 StorageControllers   The mapping for `StorageControllers` is summarized in Table 30.

Table 30: StorageControllers mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	StorageControllers	N/A
Type	Storage.StorageControllers	N/A
Descript ion	The set of storage controllers that this resource represents.	N/A
LongDesc ription	This property shall contain a set of the storage controllers that this resource represents.	N/A
Mandator y	Do Not Implement	
Notes	Deprecated for NVMe use - replaced by Controllers (type StorageController.StorageController).	

6.3.2.27 StorageGroups   The mapping for `StorageGroups` is summarized in Table 31.

Table 31: StorageGroups mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	StorageGroups	N/A
Type	StorageGroup.StorageGroup	N/A
Descript ion	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
LongDesc ription	This property shall contain a link to a resource collection of type StorageGroupsCollection. This property shall be used when implementing mapping and masking.	N/A
Mandator y	Do Not Implement	
Notes	N/A for NVMe use cases. Deprecated by Connections.	

6.3.2.28 StoragePools   The mapping for `StoragePools` is summarized in Table 32.

Table 32: StoragePools mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	StoragePools	N/A
Type	StoragePoolCollection.StoragePoolCollection	N/A
Descript ion	The set of all storage pools that are allocated by this storage subsystem. A storage pool is the set of storage capacity that can be used to produce volumes or other storage pools.	N/A
LongDesc ription	This property shall contain a link to a resource collection of type StoragePoolCollection. This property shall be used when an abstraction of media, rather than references to individual media, are used as the storage data source.	N/A
Mandator y	Optional; Use when the subsystem contains endurance groups and sets, represented by StoragePool resources.	
Notes	This collection contains a pointer to the resources in the StoragePool collection, including all defined endurance groups and sets in the subsystem.	

6.3.2.29 TargetConfigurationLockLevel   The mapping for `TargetConfigura-`
`tionLockLevel` is summarized in Table 33.

Table 33: TargetConfigurationLockLevel mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	TargetConfigurationLockLeve l	N/A
Type	enum (TargetConfigurationLockLev el)	N/A
Descript ion	Indicates the target configuration lock level for the drive resource. This corresponds to the `ConfigurationLock` and `ConfigurationLockState` for the storage subsystem type.	N/A
LongDesc ription	This property shall indicate the target configuration lock level for the drive resource. This corresponds to the `ConfigurationLock` and `ConfigurationLockState` for the storage subsystem type.	N/A
Mandator y	Optional; Use when the subsystem contains endurance groups and sets, represented by StoragePool resources.	

	Redfish/Swordfish	NVMe / NVMe-oF
Notes	The `Standard` enum is defined as the standard configuration lock level, corresponding to applying firmware, and updating security keys. See section `Mapping NVMe Command and F eature Lockdown to "Configu rationLocked"` for detailed behavior and mapping description.	

6.3.2.30 Volumes   The mapping for `Volumes` is summarized in Table 34.

Table 34: Volumes mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Volumes	N/A
Type	VolumeCollection.VolumeCollection	N/A
Descript ion	The set of volumes instantiated by this storage subsystem.	N/A
LongDesc ription	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.	N/A
Mandator y	Optional Required for NVMe drives, as well as opaque arrays and other similar devices with an NVMe front end.	
Notes	This is a collection of Namespaces that belong to this NVM Subsystem. Refer to the Volume schema for details of the instance information.	

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.

```
{
    "@odata.type":
    ↪  "#StorageController.v1_3_0.StorageController",
    "Name": "NVMe Admin Controller",
    "Description": "Single NVMe Admin Controller for in-band
    ↪  admin command access.",
    "Status": {
        "State": "Enabled",
        "Health": "OK"
    },
    "Id": "NVMeAdminController",
    "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,
            "SupportsReservations": false
        },
    "NVMeSMARTCriticalWarnings": {
        "MediaInReadOnly": false,
        "OverallSubsystemDegraded": false,
        "SpareCapacityWornOut": false
    },
    "MaxQueueSize": 1
},

"Links": {
    "NetworkDeviceFunctions": [
        {
            "@odata.id": "/red-
            ↪    fish/v1/Chassis/NVMeOpaqueArray/NetworkAdapters/OpaqueArr
        },
        {
            "@odata.id": "/red-
            ↪    fish/v1/Chassis/NVMeOpaqueArray/NetworkAdapters/OpaqueArr
        },
        {
            "@odata.id": "/red-
            ↪    fish/v1/Chassis/NVMeOpaqueArray/NetworkAdapters/OpaqueArr
        },
        {
            "@odata.id": "/red-
            ↪    fish/v1/Chassis/NVMeOpaqueArray/NetworkAdapters/OpaqueArr
        }
    ]
},
"@odata.id": "/red-
↪    fish/v1/Storage/OpaqueArray/Controllers/NVMeAdminController",
```

```
    "@Redfish.Copyright": "Copyright 2015-2024 SNIA. All
    ↪    rights reserved."
}
```

6.4.1.2 Property Mapping

6.4.1.2.1 Actions.#StorageController.RunSelfTest   The mapping for `Actions.#StorageController.R` is summarized in Table 35.

Table 35: Actions.#StorageController.RunSelfTest mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Prope rty	Actions.#StorageController.R unSelfTest	**NVM Spec Property / Field:** Device Self-test Command **NVM Spec: Section:Figure** NVM Base Specification 2.0: Section 5.9
Type	Action (Special form of POST)	NVMe Administrative command
Descr iptio n	This action is used to initiate a self-test against this controller. This action should return either the `ResourceSelfTestFailed` or `ResourceSelfTestCom-pleted` event.	The Device Self-test command is used to start a device self-test operation or abort a device self-test operation
LongD escri ption	This action is used to initiate a self-test against this controller. This action should return either the `ResourceSelfTestFailed` or `ResourceSelfTestCom-pleted` event.	The Device Self-test command is used to start a device self-test operation or abort a device self-test operation (refer to NVMe 2.0 Base Specification section 8.6).
Manda tory	Optional	Optional

Redfish/Swordfish	NVMe / NVMe-oF
Notes	The Device Self-test command is used specifically to: a) start a short device self-test operation;b) start an extended device self-test operation;c) start a vendor specific device self-test operation; ord) abort a device self-test operation already in process.

6.4.1.2.2 Assembly   The mapping for `Assembly` is summarized in Table 36.

Table 36: Assembly mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Descript ion	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
LongDesc ription	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
Mandator y	Optional Do Not Implement for NVMe drives, or devices with NVMe front ends, such as opaque arrays.	
Notes		

6.4.1.2.3 AssetTag   The mapping for `AssetTag` is summarized in Table 37.

Table 37: Assembly mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	AssetTag	N/A
Type	Edm.String	N/A
Descript ion	The user-assigned asset tag for this storage controller.	N/A
LongDesc ription	This property shall track the storage controller for inventory purposes.	N/A
Mandator y	Opitonal Do Not Implement for NVMe Drives, or devices with NVMe front ends, such as opaque arrays.	
Notes		

6.4.1.2.4 CacheSummary   The mapping for CacheSummary is summarized in Table 38.

Table 38: CacheSummary mapping

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

6.4.1.2.5 ControllerRates   The mapping for `ControllerRates` is summarized in Table 39.

Table 39: ControllerRates mapping

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

6.4.1.2.6 Description   The mapping for `Description` is summarized in Table 40.

Table 40: Description mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Description	N/A
Type	String	N/A
Descript ion	The description of this resource.	N/A
LongDesc ription	This object represents the description of this resource. The resource values shall comply with the Redfish Specification-described requirements.	N/A
Mandator y	Optional	
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."

6.4.1.2.7 FirmwareVersion   The mapping for `FirmwareVersion` is summarized in Table 41.

Table 41: FirmwareVersion mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	FirmwareVersion	**NVM Spec Property / Field:** Identify Command / Identify Controller Data structure (CNS 01h): Firmware Revision (FR) **NVM Spec: Section:Figure** NVMe 2.0: Section 5.17.2.1:Figure 275: Bytes 71:64
Type	String	String
Descript ion	The firmware version of this storage controller.	The currently active firmware revision for the domain of which this controller is a part.
LongDesc ription	This property shall contain the firmware version as defined by the manufacturer for the associated storage controller.	The currently active firmware revision for the domain of which this controller is a part.
Mandator y	Mandatory	Mandatory
Notes		

6.4.1.2.8 Identifiers   The mapping for `Identifiers` is summarized in Table 42.

Table 42: Identifiers mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Identifiers	N/A
Type	Collection(Resource.Identif ier)	N/A
Descript ion	The Durable names for the storage controller.	N/A
LongDesc ription	This property shall contain a list of all known durable names for the associated storage controller.	N/A
Mandator y	Optional	
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.	

6.4.1.2.9 Identifiers.DurableName   The mapping for `Identifiers.DurableName` is summarized in Table 43.

Table 43: Identifiers.DurableName mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Identifiers.DurableName	N/A
Type	Variable - see notes	N/A
Descript ion	The world-wide, persistent name of the resource.	N/A
LongDesc ription	This property shall contain the world-wide unique identifier for the resource. The string shall be in the Identifier.DurableNameForma t property value format.	N/A
Mandator y	Optional	
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.	

6.4.1.2.10 Identifiers.DurableNameFormat   The mapping for `Identifiers.DurableNameFormat` is summarized in Table 44.

Table 44: Identifiers.DurableNameFormat mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Identifiers.DurableNameForm at	N/A
Type	enum (DurableNameFormat)	N/A
Descript ion	The format of the durable name property.	N/A
LongDesc ription	This property shall represent the format of the DurableName property.	N/A
Mandator y	Do Not Implement.	
Notes	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.	

6.4.1.2.11 Links.AttachedVolumes   The mapping for `Links.AttachedVolumes` is summarized in Table 45.

Table 45: Links.AttachedVolumes mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Links.AttachedVolumes	N/A
Type	Collection(Volume.Volume)	N/A
Descript ion	An array of links to volumes that are attached to this controller instance.	N/A
LongDesc ription	This property shall contain a link to the Resources of type Volume that are attached to this instance of storage controller.	N/A
Mandator y	Optional Do Not Implement for admin controllers.	
Notes		

6.4.1.2.12 Links.NetworkDeviceFunctions   The mapping for `Links.NetworkDeviceFunctions` is summarized in Table 46.

Table 46: Links.NetworkDeviceFunctions mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Links.NetworkDeviceFunction s	N/A
Type	Collection(NetworkDeviceFun ction.NetworkDeviceFunction)	N/A
Descript ion	The network device functions that provide connectivity to this controller.	N/A
LongDesc ription	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
Mandator y	Optional. Recommended for NVMe-oF configurations and more complex devices with NVMe front ends, such as opaque arrays.	
Notes		

6.4.1.2.13 Location   The mapping for `Location` is summarized in Table 47.

Table 47: Location mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Location	N/A
Type	Collection(Resource.Locatio n)	N/A
Descript ion	The location of the storage controller.	N/A
LongDesc ription	This property shall contain location information of the associated storage controller.	N/A
Mandator y	Optional Do Not Implement for NVM Drives or more complex devices with an NVMe front end, such as opaque arrays.	
Notes		

6.4.1.2.14 Manufacturer   The mapping for `Manufacturer` is summarized in Table 48.

Table 48: Manufacturer mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Manufacturer	**NVM Spec Property / Field:** Identify Command / Identify Controller Datastructure (CNS 01h) / PCI Vendor ID (VID) **NVM Spec: Section: Figure** NVMe 2.0: Section 5.17.2.1: Figure 275: Bytes 00:01
Type	String	16-bit number in little endian format.
Descript ion	The manufacturer of this storage controller.	The company vendor identifier
LongDesc ription	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
Mandator y	Optional	NVMe: Mandatory NVMe-oF: DNI

	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, it would be preferable to have this filled with the actual string value of the company name.	

6.4.1.2.15 Model   The mapping for Model is summarized in Table 49.

Table 49: Model mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Model	**NVM Spec Property / Field:** Identify Command / Identify Controller Datastructure (CNS 01h) / Model Number (MN) **NVM Spec: Section: Figure** NVMe 2.0: Section 5.17.2.1: Figure 275: Bytes 24:63
Type	String	String
Description	The model number for the storage controller.	Model Number (MN)
LongDescription	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 NVMe 2.0 Section 7.10 for unique identifier requirements. Refer to NVMe 2.0 Section 1.5 for ASCII string requirements
Mandatory Notes	Recommended	Mandatory

6.4.1.2.16 Name   The mapping for Name is summarized in Table 50.

Table 50: Name mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Name	**NVM Spec Property / Field:** Identify Command / Identify Controller Data Data structure (CNS 01h) / Controller ID (CNTLID) **NVM Spec: Section:Figure** NVMe 2.0: Section 5.17.2.1: Figure 275: Bytes 78:79
Type	String	16-bit hex value
Descript ion	The name of the resource or array member.	Controller ID
LongDesc ription	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 unique controller identifier associated with the controller.
Mandator y	Mandatory	Mandatory
Notes	In Redfish, Name is a read-only field.	Map the CNTLID field to a string with the format: "0xABCD"

6.4.1.2.17 NVMeControllerProperties.ControllerType The mapping for NVMeCon‐
troller`Properties.ControllerType` is summarized in Table 51.

Table 51: NVMeControllerProperties.ControllerType mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeControllerProperties.Co ntrollerType	**NVM Spec Property / Field:** Identify Command / Identify Controller Datastructure (CNS 01h) / Controller Type (CNTRL_TYPE) **NVM Spec: Section:Figure** NVMe 2.0: Section 5.17.2.1: Figure 275: Bytes 111
Type	StorageController.v1_0_0. NVMeControllerType	Hex value
Descript ion	This property specifies the type of NVMe Controller.	Controller Type
LongDesc ription	This property shall specify the type of NVMe Controller.	This field specifies the controller type. Values reported: 0h Reserved (controller type not reported)1h I/O controller2h Discovery controller3h Administrative controller
Mandator y	Mandatory	Mandatory
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"

6.4.1.2.18 NVMeControllerProperties.NVMeVersion   The mapping for `NVMeCon-trollerProperties.NVMeVersion` is summarized in Table 52.

Table 52: NVMeControllerProperties.NVMeVersion mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeControllerProperties.NVMeVersion	**NVM Spec Property / Field:** Identify Command / Identify Controller Datastructure (CNS 01h) / Version (VER) **NVM Spec: Section:Figure** NVMe 2.0: Section 5.17.2.1: Figure 275: Bytes 80:83
Type	String	32-bit value.
Descript ion	The version of the NVMe Base Specification supported.	Version (VER)
LongDesc ription	This property shall specify the type of NVMe Controller.	Indicates the major, minor, and tertiary version of the NVM Express base specification that the controller implementation supports. bits 31:16 indicate the major versionbits 15:08 indicate the minor version bits 07:00 indicate the tertiary version number
Mandator y	Optional	Mandatory
Notes		Implementations compliant to NVM Express Base Specification revision 1.2 or later shall report a non-zero value in this field. Valid versions of the specification are: 1.0, 1.1, 1.2, 1.2.1, 1.3, 1.4, and 2.0.

6.4.1.2.19 NVMeControllerProperties.NVMeControllerAttributes.ReportsNamespaceGranularity

The mapping for `NVMeControllerProperties.NVMeControllerAttributes.ReportsNamespace` is summarized in Table 53.

Table 53:
NVMeControllerProperties.NVMeControllerAttributes.ReportsNamespaceGranularity mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeControllerProperties.NVMeControllerAttributes.ReportsNamespaceGranularity	**NVM Spec Property / Field:** Identify Command / Identify Controller Datastructure (CNS 01h) / Controller Attributes (CTRATT): Bit 07 (Namespace Granularity) **NVM Spec: Section:Figure** NVMe 2.0: Section 5.17.2.1: Figure 275: Bytes 96:99: Bit 07
Type	Boolean	Single bit (bool)
Descript ion	Indicates whether or not the controller supports reporting of Namespace Granularity.	Indicates whether or not the controller supports reporting of Namespace Granularity.
LongDesc ription	This property shall indicate whether or not the controller supports reporting of Namespace Granularity.	This property shall indicate whether or not the controller supports reporting of Namespace Granularity.
Mandator y	Optional Recommended for NVM Drives and more complex devices with NVMe front ends, such as opaque arrays.	
Notes		

6.4.1.2.20 NVMeControllerProperties.NVMeControllerAttributes.SupportsSQAssociations

The mapping for NVMeControllerProperties.NVMeControllerAttributes.SupportsSQAssoci
is summarized in Table 54.

Table 54:

NVMeControllerProperties.NVMeControllerAttributes.SupportsSQAssociations Mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeControllerProperties.NVMeControllerAttributes.SupportsSQAssociations	**NVM Spec Property / Field:** Identify Command / Identify Controller Datastructure (CNS 01h) / Controller Attributes (CTRATT): Bit 08 (SQ Associations) **NVM Spec: Section:Figure** NVMe 2.0: Section 5.17.2.1: Figure 275: Bytes 96:99: Bit 08
Type	Boolean	Single bit (bool)
Description	Indicates whether or not the controller supports SQ Associations.	Indicates whether or not the controller supports SQ Associations.
LongDescription	This property shall indicate whether or not the controller supports SQ Associations.	This property shall indicate whether or not the controller supports SQ Associations.
Mandatory	Optional	Mandatory
Notes		

6.4.1.2.21 **NVMeControllerProperties.NVMeControllerAttributes.SupportsTrafficBasedKeepAlive**

The mapping for NVMeControllerProperties.NVMeControllerAttributes.TrafficBasedKeep
is summarized in Table 55.

Table 55:
NVMeControllerProperties.NVMeControllerAttributes.SupportsTrafficBasedKeepAlive
mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeControllerProperties.NVMeControllerAttributes.SupportsTrafficBasedKeepAlive	**NVM Spec Property / Field:** Identify Command / Identify Controller Datastructure (CNS 01h) / Controller Attributes (CTRATT): Bit 06 (Traffic Based Keep Alive Support – TBKAS) **NVM Spec: Section:Figure** NVMe 2.0: Section 5.17.2.1: Figure 275: Bytes 96:99: Bit 06
Type	Boolean	Single bit (bool)
Descript ion	Indicates whether or not the controller supports restarting KeepAlive Timer if traffic is processed from an admin command or IO during KeepAlive Timeout interval.	Indicates if the the host and controller are allowed to restart the Traffic Based Keep Alive Timer in the presence of Admin or I/O command processing.
LongDesc ription	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.	This property shall indicate whether or not the the host and controller are allowed to restart the Traffic Based Keep Alive Timer in the presence of Admin or I/O command processing.

	Redfish/Swordfish	NVMe / NVMe-oF
Mandator y	Optional Mandatory for Ethernet-Attach Drives; Mandatory for more complex devices with NVMe front ends, such as opaque arrays.	Mandatory
Notes		

6.4.1.2.22 NVMeControllerProperties.NVMeControllerAttributes.SupportsExceedingPowerOfNonOperatic

The mapping for NVMeControllerProperties.NVMeControllerAttributes.SupportsExceedin
is summarized in Table 56.

Table 56: NVMeControllerProper-
ties.NVMeControllerAttributes.SupportsExceedingPowerOfNonOperationalState
mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeControllerProperties.NVMeControllerAttributes. SupportsExceedingPowerOfNonOperationalState	**NVM Spec Property / Field:** Identify Command / Identify Controller Datastructure (CNS 01h) / Controller Attributes (CTRATT): Bit 01 (Non-Operational Power State Permissive Mode) **NVM Spec: Section:Figure** NVMe 2.0: Section 5.17.2.1: Figure 275: Bytes 96:99: Bit 01
Type	Boolean	Single bit (bool)
Descript ion	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.	Indicates whether or not the controller supports host control of whether the controller may temporarily exceed the power of a non-operational power state for the purpose of executing controller initiated background operations in a non-operational power state (i.e., Non-Operational Power State Permissive Mode supported).

	Redfish/Swordfish	NVMe / NVMe-oF
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.	This property indicates whether or not the controller supports host control of whether the controller may temporarily exceed the power of a non-operational power state for the purpose of executing controller initiated background operations in a non-operational power state (i.e., Non-Operational Power State Permissive Mode supported).
Mandatory Notes	Optional	Mandatory

6.4.1.2.23 **NVMeControllerProperties.NVMeControllerAttributes.Supports128BitHostId**

The mapping for NVMeControllerProperties.NVMeControllerAttributes.Supports128BitHo
is summarized in Table 57.

Table 57: NVMeControllerProperties.NVMeControllerAttributes.Supports128BitHostId
mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeControllerProperties.NVMeControllerAttributes.Supports128BitHostId	**NVM Spec Property / Field:** Identify Command / Identify Controller Datastructure (CNS 01h) / Controller Attributes (CTRATT): Bit 00 **NVM Spec: Section:Figure** NVMe 2.0: Section 5.17.2.1: Figure 275: Bytes 96:99: Bit 00
Type	Boolean	Single bit (bool)
Descript ion	Indicates whether or not the controller supports a 128-bit Host Identifier.	Indicates whether or not the controller supports a 128-bit Host Identifier.
LongDesc ription	This property shall indicate whether or not the controller supports a 128-bit Host Identifier.	This property shall indicate whether or not the controller supports a 128-bit Host Identifier.
Mandator y	Mandatory	Mandatory
Notes		

6.4.1.2.24 NVMeControllerProperties.MaxQueueSize   The mapping for NVMeCon-trollerProperties.MaxQueueSize is summarized in Table 58.

Table 58: NVMeControllerProperties.ANACharacteristics mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeControllerProperties.MaxQueueSize	**NVM Spec Property / Field:** Property Get Command / Controller Capabilities (CAP) - Offset 0h / Maximum Queue Entries Supported (MQES) **NVM Spec: Section:Figure** NVMe 2.0: Section 3.1.3.1: Figure 36 : Bits 15:00
Type	Int64	16-bit value
Descript ion	Indicates the maximum individual queue size that an NVMe IO Controller supports.	Indicates the maximum individual queue size that the controller supports.
LongDesc ription	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.	Indicates the maximum individual queue size that the controller supports. This is a 0's based value. The minimum value is 1h, indicating two entries.
Mandator y	Optional Mandatory for more complex devices with NVMe front ends, such as opaque arrays.	Mandatory

	Redfish/Swordfish	NVMe / NVMe-oF
Notes	For PCIe, this applies to both submission and completion queues. For NVMe-oF, this applies to only submission queues.	For NVMe over PCIe implementations, this value applies to the I/O Submission Queues and I/O Completion Queues that the host creates. For NVMe over Fabrics implementations, this value applies to only the I/O Submission Queues that the host creates.

6.4.1.2.25 NVMeControllerProperties.ANACharacteristics The mapping for NVMe‐
`ControllerProperties.ANACharacteristics` is summarized in Table 59.

Table 59: NNVMeControllerProperties.ANACharacteristics mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeControllerProperties.ANACharacteristics	**NVM Spec Property / Field:** Asymmetric Namespace Access Log (Log ID 0ch) **NVM Spec: Section:Figure** NVMe 2.0: Section 5.16.1.13: Figures 221 and 222
Type	Collection(StorageController.v1_0_0.ANACharacteristi cs)	Log Page.
Descript ion	This property contains the combination of ANA type and volume information.	This log consists of a header describing the log and descriptors containing the asymmetric namespace access information for ANA Groups
LongDesc ription	This property shall contain the combination of ANA type and volume information.	This log consists of a header describing the log and descriptors containing the asymmetric namespace access information for ANA Groups that contain namespaces that are attached to the controller processing the command. ANA Group Descriptors shall be returned in ascending ANA Group Identifier order
Mandator y Notes	Do Not Implement	Prohibited

6.4.1.2.26 NVMeControllerProperties.NVMeSMARTCriticalWarnings.OverallSystemDegraded

The mapping for NVMeControllerProperties.NVMeSMARTCriticalWarnings.OverallSystemDe
is summarized in Table 60.

Table 60:

NVMeControllerProperties.NVMeSMARTCriticalWarnings.OverallSystemDegraded
mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeControllerProperties.NV MeSMARTCriticalWarnings.Ove rallSystemDegraded	**NVM Spec Property / Field:** SMART / Health Information Log (Log ID 02h) / Critical Warning / Subsystem Reliability has been degraded **NVM Spec: Section:Figure** NVMe 2.0: Section 5.16.1.3: Figure 207: Bytes 00, Bits 02
Type	Boolean	Single bit (bool)
Descript ion	Indicates that the NVM subsystem reliability has been compromised.	Indicates that the NVM subsystem reliability has been degraded due to significant media related errors or any internal error that degrades NVM subsystem reliability.
LongDesc ription	This property shall indicate that the NVM subsystem reliability has been compromised.	This property Indicates that the NVM subsystem reliability has been degraded due to significant media related errors or any internal error that degrades NVM subsystem reliability.
Mandator y Notes	Mandatory	Optional

6.4.1.2.27 **NVMeControllerProperties.NVMeSMARTCriticalWarnings.PowerBackupFailed**

The mapping for `NVMeControllerProperties.NVMeSMARTCriticalWarnings.PowerBackupFail` is summarized in Table 61.

Table 61: NVMeControllerProperties.NVMeSMARTCriticalWarnings.PowerBackupFailed mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeControllerProperties.NVMeSMARTCriticalWarnings.PowerBackupFailed	**NVM Spec Property / Field:** SMART / Health Information Log (Log ID 02h) / Critical Warning / volatile memory backup device has failed **NVM Spec: Section:Figure** NVMe 2.0: Section 5.16.1.3: Figure 207: Bytes 00, Bits 04
Type	Boolean	Single bit (bool)
Descript ion	Indicates that the volatile memory backup device has failed.	Indicates that the volatile memory backup device has failed.
LongDesc ription	This property shall indicate that the volatile memory backup device has failed.	This warning indicates that the volatile memory backup device has failed.
Mandator y	Optional Recommended for NVM Drives.Required for more complex devices with NVMe front ends, such as opaque arrays.	Optional
Notes		This field is only valid if the controller has a volatile memory backup solution.

6.4.1.2.28 SKU   The mapping for SKU is summarized in Table 62.

Table 62: SKU mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	SKU	N/A
Type	Edm.String	N/A
Descript ion	The SKU for this storage controller.	N/A
LongDesc ription	This property shall contain the stock-keeping unit number for this storage storage controller.	N/A
Mandator y	Do Not Implement	
Notes		

6.4.1.2.29 SpeedGbps   The mapping for SpeedGbps is summarized in Table 63.

Table 63: SpeedGbps mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	SpeedGbps	N/A
Type	Decimal	N/A
Descript ion	The maximum speed of the storage controller's device interface.	N/A
LongDesc ription	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
Mandator y	Do Not Implement	
Notes		

6.4.1.2.30 Status.Health   The mapping for `Status.Health` is summarized in Table 64.

Table 64: Status.Health mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Status.Health	**NVM Spec Property / Field:** Property Get command / Controller Status (CSTS): - Offset 1Ch **NVM Spec: Section:Figure** NVMe 2.0: Section 3.1.3.6: Figure 47**NVM Spec Property / Field:** Get Log Page command / SMART/Health Information Log (Log Identifier 02h)**NVM Spec: Section:Figure** NVMe 2.0: Section 5.16.1.3: Figure 207
Type	Resource.Health	32-bit value
Descript ion	The health state of this resource in the absence of its dependent resources.	This field indicates critical warnings for the state of the controller from both the SMART & general health information log and the Controller Status Property
LongDesc ription	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.	This field indicates critical warnings for the state of the controller from the SMART and general health information . The information provided is over the life of the controller and is retained across power cycles.
Mandator y	Mandatory	Optional

	Redfish/Swordfish	NVMe / NVMe-oF
Notes	Possible Values: OK / Warning / Critical	This comes from CSTS Controller Status. Redfish/Swordfish "OK" corresponds to NVMe/NVMe-oF "Ready" (RDY) bit set to "1" and no warnings in the CSTS or SMART/Health information log pageRedfish /Swordfish "Warning" corresponds to "Volatile Memory Backup Device has failed", "NVM subsystem reliability has been degraded", "Temperature greater than or equal to an over temperature threshold", "temperature less than or equal to an under temperature threshold", or "available spare capacity has fallen below the threshold"Redfish/Swordfish "Critical" corresponds to NVMe/NVMe-oF "Controller Fatal Status" (CSTS.CFS) bit set to "1" which indicates that a serious error condition has occurred.

6.4.1.2.31 Status.State   The mapping for `Status.State` is summarized in Table 65.

Table 65: Status.State mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Status.State	**NVM Spec Property / Field:** Property Get Command / CSTS – Controller Status (Offset 1Ch) **NVM Spec: Section:Figure** NVMe 2.0: Section 3.1.3.6: Figure 47
Type	Resource.State (enum)	32-bit value
Descript ion	The known state of the resource, such as, enabled.	Controller Status
LongDesc ription	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.	Controller Status
Mandator y	Mandatory	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), Processing Paused (CSTS.PP) maps to Deferring. If both Ready and Shutdown are indicated, then the system should indicate StandbyOffline. If both Ready and Processing Paused are indicated, then the system should indicate Deferring.

6.4.1.2.32 SupportedControllerProtocols The mapping for `SupportedCon-trollerProtocols` is summarized in Table 66.

Table 66: SupportedControllerProtocols mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	SupportedControllerProtocol s	N/A
Type	Collection(Protocol.Protoco l)	N/A
Descript ion	The supported set of protocols for communicating to this storage controller.	N/A
LongDesc ription	This property shall contain the supported set of protocols for communicating to this storage controller.	N/A
Mandator y	Mandatory	
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.	

6.4.1.2.33 SupportedDeviceProtocols   The mapping for `SupportedDevicePro-`
`tocols` is summarized in Table 67.

Table 67: SupportedDeviceProtocols mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	SupportedDeviceProtocols	N/A
Type	Collection(Protocol.Protoco l)	N/A
Descript ion	The protocols that the storage controller can use to communicate with attached devices.	N/A
LongDesc ription	This property shall contain the set of protocols this storage controller can use to communicate with attached devices.	N/A
Mandator y	Do Not Implement	
Notes		

6.4.2 Discovery Controller

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

```
{
  "@odata.id": "/redfish/v1/Storage/NVMeoF-
  ↪   Discovery/Controllers/NVMeDiscoveryController",
  "@odata.type":
  ↪   "#StorageController.v1_7_0.StorageController",
  "Id": "9",
  "Name": "NVMe Discovery Controller",
  "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.",

  "Status": {
    "State": "Enabled",
    "Health": "OK"
  },
  "Id": "NVMeDiscoveryController",

  "SupportedControllerProtocols": [
    "TCP", "RoCEv2", "FC"
  ],
  "SupportedDeviceProtocols": [
    "NVMeOverFabrics"
  ],

  "NVMeControllerProperties": {
    "NVMeVersion": "1.3",
    "NVMeControllerAttributes": {
      "ReportsUUIDList": false,
      "SupportsSQAssociations": false,
```

```
        "ReportsNamespaceGranularity": false,
        "SupportsTrafficBasedKeepAlive": false,
        "SupportsPredictableLatencyMode": false,
        "SupportsEnduranceGroups": false,
        "SupportsReadRecoveryLevels": false,
        "SupportsNVMSets": false,
        "SupportsExceedingPowerOfNonOperationalState": false,
        "Supports128BitHostId": false
      }
    },
    "Links": {
      "Endpoints": [{
        "@odata.id": "/redfish/v1/Fabrics/NVMe-
          ↪  oF/Endpoints/NVMeSubsystemEndpoint1"
      }],
      "NVMeDiscoveredSubsystems": [{
        "@odata.id": "/redfish/v1/Storage/NVMeoF-SS1"
      },
      {
        "@odata.id": "/redfish/v1/Storage/NVMeoF-SS2"
      }
    ]
    },
    "@Redfish.Copyright": "Copyright 2015-2024 SNIA. All
      ↪  rights reserved."
}
```

6.4.2.2 Property Mapping

6.4.2.2.1 Assembly   The mapping for `Assembly` is summarized in Table 68.

Table 68: Assembly mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Assembly	N/A
Type	ComplexType	N/A
Descript ion	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
LongDesc ription	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
Mandator y	Optional 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 69.

Table 69: AssetTag mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	AssetTag	N/A
Type	Edm.String	N/A
Descript ion	The user-assigned asset tag for this storage controller.	N/A
LongDesc ription	This property shall track the storage controller for inventory purposes.	N/A
Mandator y	Optional 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 70.

Table 70: CacheSummary mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	CacheSummary	N/A
Type	ComplexType	N/A
Description	The cache memory of the storage controller in general detail.	N/A
LongDescription	This property shall contain properties that describe the cache memory for this resource.	N/A
Mandatory	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 71.

Table 71: ControllerRates mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	ControllerRates	N/A
Type	ComplexType	N/A
Descript ion	This property describes the various controller rates used for processes such as volume rebuild or consistency checks.	N/A
LongDesc ription	This object shall contain all the rate settings available on the controller.	N/A
Mandator y	Do Not Implement	
Notes		

6.4.2.2.5 Description   The mapping for `Description` is summarized in Table 72.

Table 72: Description mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Description	N/A
Type	String	N/A
Descript ion	The description of this resource.	N/A
LongDesc ription	This object represents the description of this resource. The resource values shall comply with the Redfish Specification-described requirements.	N/A
Mandator y	Optional	
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 73.

Table 73: FirmwareVersion mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	FirmwareVersion	**NVM Spec Property / Field:** Identify Command / Identify Controller Datastructure (CNS 01h): Firmware Revision (FR) **NVM Spec: Section:Figure** NVMe 2.0: Section 5.17.2.1:Figure 275: Bytes 71:64
Type	String	String
Descript ion	The firmware version of this storage controller.	The currently active firmware revision for the domain of which this controller is a part.
LongDesc ription	This property shall contain the firmware version as defined by the manufacturer for the associated storage controller.	The currently active firmware revision for the domain of which this controller is a part.
Mandator y	Mandatory	Mandatory
Notes		Return the currently active firmware revision information.

6.4.2.2.7 Identifiers   The mapping for `Identifiers` is summarized in Table 74.

Table 74: Identifiers mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Identifiers	N/A
Type	Collection(Resource.Identif ier)	N/A
Descript ion	The Durable names for the storage controller.	N/A
LongDesc ription	This property shall contain a list of all known durable names for the associated storage controller.	N/A
Mandator y	Optional	
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.	

6.4.2.2.8 Identifiers.DurableName The mapping for `Identifiers.DurableName` is summarized in Table 75.

Table 75: Identifiers.DurableName mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Identifiers.DurableName	N/A
Type	Variable - see notes	N/A
Descript ion	The world-wide, persistent name of the resource.	N/A
LongDesc ription	This property shall contain the world-wide unique identifier for the resource. The string shall be in the Identifier.DurableNameForma t property value format.	N/A
Mandator y	Optional	
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.	

6.4.2.2.9 Identifiers.DurableNameFormat   The mapping for `Identifiers.DurableNameFormat` is summarized in Table 76.

Table 76: Identifiers.DurableNameFormat mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Identifiers.DurableNameForm at	N/A
Type	enum (DurableNameFormat)	N/A
Descript ion	The Durable names for the storage controller.	N/A
LongDesc ription	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
Mandator y	Optional	
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.	

6.4.2.2.10 Links.AttachedVolumes   The mapping for `Links.AttachedVolumes` is summarized in Table 77.

Table 77: Links.AttachedVolumes mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Links.AttachedVolumes	N/A
Type	Collection(Volume.Volume)	N/A
Descript ion	An array of links to volumes that are attached to this controller instance.	N/A
LongDesc ription	This property shall contain a link to the Resources of type Volume that are attached to this instance of storage controller.	N/A
Mandator y	Do Not Implement	
Notes		

6.4.2.2.11 Links.Endpoints   The mapping for `Links.Endpoints` is summarized in Table 78.

Table 78: Links.Endpoints mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Links.Endpoints	N/A
Type	Collection(Endpoint.Endpoin t)	N/A
Descript ion	An array of links to the endpoints that connect to this controller.	N/A
LongDesc ription	This property shall contain an array of links to the Resources of type Endpoint associated with this controller.	N/A
Mandator y	Mandatory	
Notes		

6.4.2.2.12 Links.Connections   The mapping for `Links.Connections` is summarized in Table 79.

Table 79: Links.Connections mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Links.Connections	N/A
Type	Collection(Connection.Connection)	N/A
Descript ion	An array of links to volumes that are attached to this controller instance.	N/A
LongDesc ription	This property shall contain a link to the Resources of type Volume that are attached to this instance of storage controller.	N/A
Mandator y	Optional	
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 80.

Table 80: Links.NetworkDeviceFunctions mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Links.NetworkDeviceFunction s	N/A
Type	Collection(NetworkDeviceFun ction.NetworkDeviceFunction)	N/A
Descript ion	The network device functions that provide connectivity to this controller.	N/A
LongDesc ription	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
Mandator y	Optional 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 Links.NVMeDiscoveredSubsystems   The mapping for `Links.NVMeDiscoveredSubsystems` is summarized in Table 81.

Table 81: Links.NVMeDiscoveredSubsystems mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Links.NVMeDiscoveredSubsyst ems	**NVM Spec Property / Field:** NVMe 2.0 Base Specification: 5.16.1.23 Discovery Log Page (Log Identifier 70h) GetLogPage
Type	Collection(Storage.Storage)	N/A
Descript ion	The NVMe subsystems discovered by this discovery controller.	The Discovery Log Page provides an inventory of NVM subsystems with which a host may attempt to form an association. The Discovery Log Page may be specific to the host requesting the log.
LongDesc ription	This property shall contain an array of links to resources of type Storage that represent the NVMe subsystems discovered by this discovery controller. This property shall only be present if ControllerType in NVMeControllerProperties contains `Discovery`.	
Mandator y	Mandatory	Mandatory
Notes	This contains the list of all Subsystems discovered by this Discovery Controller.	The Discovery Log Page is persistent across power cycles.

6.4.2.2.15 Location The mapping for `Location` is summarized in Table 82.

Table 82: Location mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Location	N/A
Type	Collection(Resource.Locatio n)	N/A
Descript ion	The location of the storage controller.	N/A
LongDesc ription	This property shall contain location information of the associated storage controller.	N/A
Mandator y	Optional Do Not Implement for NVM Drives, or more complex devices with NVMe front ends, such as opaque arrays.	
Notes		

6.4.2.2.16 Manufacturer The mapping for `Manufacturer` is summarized in Table 83.

Table 83: Manufacturer mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Manufacturer	N/A
Type	String	N/A
Descript ion	The manufacturer of this storage controller.	N/A
LongDesc ription	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.	N/A
Mandator y	Optional	Reserved (DNI)
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.	

6.4.2.2.17 Model   The mapping for Model is summarized in Table 84.

Table 84: Model mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Model	N/A
Type	String	N/A
Descript ion	The model number for the storage controller.	N/A
LongDesc ription	This property shall contain the name by which the manufacturer generally refers to the storage controller.	N/A
Mandator y	Do Not Implement	Reserved (DNI) for Discovery Controllers
Notes		

6.4.2.2.18 Name   The mapping for Name is summarized in Table 85.

Table 85: Name mapping

		Redfish/Swordfish
Property	Name	**NVM Spec Property / Field:** Identify Commmand / Identify Controller Datastructure (CNS 01h) / Controller ID (CNTLID) **NVM Spec: Section:Figure** NVMe 2.0: Section 5.17.2.1: Figure 275: Bytes 78:79
Type	String	16-bit hex value
Descript ion	The name of the resource or array member.	Controller ID
LongDesc ription	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 unique controller identifier associated with the controller.
Mandator y	Mandatory	Mandatory
Notes	In Redfish, Name is a read-only field.	Map the CNTLID field to a string with the format: "0xABCD"

6.4.2.2.19 NVMeControllerProperties.ControllerType   The mapping for NVMeCon-
trollerProperties.ControllerType is summarized in Table 86.

Table 86: NVMeControllerProperties.ControllerType mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeControllerProperties.Co ntrollerType	**NVM Spec Property / Field:** Identify Command / Identify Conntroller Datastructure (CNS 01h) / Controller Type (CNTRL_TYPE) **NVM Spec: Section:Figure** NVMe 2.0: Section 5.17.2.1: Figure 275: Bytes 111
Type	StorageController.v1_0_0. NVMeControllerType	Hex value
Descript ion	This property specifies the type of NVMe Controller.	Controller Type
LongDesc ription	This property shall specify the type of NVMe Controller.	This field specifies the controller type. Values reported: 0h Reserved (controller type not reported)1h I/O controller2h Discovery controller3h Administrative controller
Mandator y	Mandatory	Mandatory
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"

6.4.2.2.20 NVMeControllerProperties.NVMeVersion   The mapping for `NVMeCon-trollerProperties.NVMeVersion` is summarized in Table 87.

Table 87: NVMeControllerProperties.NVMeVersion mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeControllerProperties.NVMeVersion	**NVM Spec Property / Field:** Identify Command / Identify Controller Datastructure (CNS 01h) / Version (VER) **NVM Spec: Section:Figure** NVMe 2.0: Section 5.17.2.1: Figure 275: Bytes 80:83
Type	String	32-bit value.
Descript ion	The version of the NVMe Base Specification supported.	Version (VER)
LongDesc ription	This property shall contain the version of the NVMe Base Specification supported.	Indicates the major, minor, and tertiary version of the NVM Express base specification that the controller implementation supports.
Mandator y	Recommended Mandatory if NVMe version is 1.2 or later.	Mandatory
Notes		Implementations compliant to NVM Express Base Specification revision 1.2 or later shall report a non-zero value in this field. Valid versions of the specification are: 1.0, 1.1, 1.2, 1.2.1, 1.3, 1.4, and 2.0. bits 31:16 indicate the major versionbits 15:08 indicate the minor version bits 07:00 indicate the tertiare version number

6.4.2.2.21 NVMeControllerProperties.NVMeControllerAttributes.SupportsTrafficBasedKeepAlive

The mapping for NVMeControllerProperties.NVMeControllerAttributes.TrafficBasedKeer
is summarized in Table 88.

Table 88:
NVMeControllerProperties.NVMeControllerAttributes.SupportsTrafficBasedKeepAlive
SupportsTrafficBasedKeepAlive mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeControllerProperties.NV MeControllerAttributes.Supp ortsTrafficBasedKeepAlive	N/A
Type	Boolean	N/A
Descript ion	Indicates whether or not the controller supports restarting KeepAlive Timer if traffic is processed from an admin command or IO during KeepAlive Timeout interval.	N/A
LongDesc ription	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.	N/A
Mandator y Notes	Do Not Implement	DNI (Reserved)

6.4.2.2.22 **NVMeControllerProperties.NVMeControllerAttributes.SupportsExceedingPowerOfNonOperatio**

The mapping for NVMeControllerProperties.NVMeControllerAttributes.SupportsExceedir
is summarized in Table 89.

Table 89: NVMeControllerProper-
ties.NVMeControllerAttributes.SupportsExceedingPowerOfNonOperationalState
SupportsExceedingPowerOfNonOperationalState mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeControllerProperties.NV MeControllerAttributes.Supp ortsExceedingPowerOfNonOper ationalState	N/A
Type	Boolean	N/A
Descript ion	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.	N/A
LongDesc ription	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.	N/A
Mandator y Notes	Optional	DNI (Reserved))

6.4.2.2.23 NVMeControllerProperties.NVMeControllerAttributes.Supports128BitHostId

The mapping for NVMeControllerProperties.NVMeControllerAttributes.Supports128BitHo
is summarized in Table 90.

Table 90: NVMeControllerProperties.NVMeControllerAttributes.Supports128BitHostId
Supports128BitHostId mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeControllerProperties.NVMeControllerAttributes.Supports128BitHostId	N/A
Type	Boolean	N/A
Descript ion	Indicates whether or not the controller supports a 128-bit Host Identifier.	N/A
LongDesc ription	This property shall indicate whether or not the controller supports a 128-bit Host Identifier.	N/A
Mandator y Notes	Do Not Implement	DNI (Reserved)

6.4.2.2.24 NVMeControllerProperties.MaxQueueSize   The mapping for NVMeControllerProperties.MaxQueueSize is summarized in Table 91.

Table 91: NVMeControllerProperties.MaxQueueSize mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeControllerProperties.MaxQueueSize	**NVM Spec Property / Field:** Property Get command / Controller Capabilities (CAP) - Offset 0h: Maximum Queue Entries Supported (MQES) **NVM Spec: Section:Figure** NVMe 2.0: Section 3.1.3.1: Figure 36: Bits 15:00
Type	Int64	16-bit value
Descript ion	Indicates the maximum individual queue size that an NVMe IO Controller supports.	Indicates the maximum individual queue size that the controller supports.
LongDesc ription	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.	indicates the maximum individual queue size that the controller supports. For NVMe over PCIe implementations, this value applies to the I/O Submission Queues and I/O Completion Queues that the host creates. For NVMe over Fabrics implementations, this value applies to only the I/O Submission Queues that the host creates. This is a 0's based value. The minimum value is 1h, indicating two entries.

	Redfish/Swordfish	NVMe / NVMe-oF
Mandator y	Optional Mandatory for more complex devices with NVMe front ends, such as opaque arrays.	Mandatory
Notes		

6.4.2.2.25 NVMeControllerProperties.NVMeSMARTCriticalWarnings.OverallSubsystemDegraded

The mapping for NVMeControllerProperties.NVMeSMARTCriticalWarnings.OverallSubsyste
is summarized in Table 92.

Table 92:

NVMeControllerProperties.NVMeSMARTCriticalWarnings.OverallSubsystemsDegraded
OverallSubsystemDegraded mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeControllerProperties.NV MeSMARTCriticalWarnings.Ove rallSubsystemDegraded	N/A
Type	Boolean	N/A
Descript ion	Indicates that the NVM subsystem reliability has been compromised.	N/A
LongDesc ription	This property shall indicate that the NVM subsystem reliability has been compromised.	N/A
Mandator y	Do Not Implement	
Notes		

6.4.2.2.26 NVMeControllerProperties.NVMeSMARTCriticalWarnings.SpareCapacityWornOut

The mapping for `NVMeControllerProperties.NVMeSMARTCriticalWarnings.SpareCapacityWo`
is summarized in Table 93.

Table 93:
NVMeControllerProperties.NVMeSMARTCriticalWarnings.SpareCapacityWornOut
SpareCapacityWornOut mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeControllerProperties.NV MeSMARTCriticalWarnings.Spa reCapacityWornOut	N/A
Type	Boolean	N/A
Descript ion	Indicates that the available spare capacity has fallen below the threshold.	N/A
LongDesc ription	This property shall indicate that the available spare capacity has fallen below the threshold.	N/A
Mandator y	Do Not Implement	
Notes		

6.4.2.2.27 NVMeControllerProperties.NVMeSMARTCriticalWarnings.PowerBackupFailed

The mapping for `NVMeControllerProperties.NVMeSMARTCriticalWarnings.PowerBackupFail`
is summarized in Table **??**.

Table 94: NVMeControllerProperties.NVMeSMARTCriticalWarnings.PowerBackupFailed
mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeControllerProperties.NVMeSMARTCriticalWarnings.PowerBackupFailed	N/A
Type	Boolean	N/A
Descript ion	Indicates that the volatile memory backup device has failed.	N/A
LongDesc ription	This property shall indicate that the volatile memory backup device has failed.	N/A
Mandator y Notes	Do Not Implement	

6.4.2.2.28 Status.Health   The mapping for `Status.Health` is summarized in Table 95.

Table 95: Status.Health mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Status.Health	N/A
Type	Resource.Health	N/A
Descript ion	The health state of this resource in the absence of its dependent resources.	N/A
LongDesc ription	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.	N/A
Mandator y	Mandatory	
Notes	Possible Values: OK / Warning / Critical	

6.4.2.2.29 Status.State   The mapping for `Status.State` is summarized in Table 96.

Table 96: Status.State mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Status.State	**NVM Spec Property / Field:** Property Get Command / CSTS – Controller Status (Offset 1Ch) **NVM Spec: Section:Figure** NVMe 2.0: Section 3.1.3.6: Figure 47
Type	Resource.State (enum)	32-bit value
Descript ion	The known state of the resource, such as, enabled.	Controller Status
LongDesc ription	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.	Controller Status
Mandator y	Mandatory	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.

6.4.2.2.30 SupportedControllerProtocols The mapping for `SupportedCon-trollerProtocols` is summarized in Table 97.

Table 97: SupportedControllerProtocols mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	SupportedControllerProtocol s	N/A
Type	Collection(Protocol.Protoco l)	N/A
Descript ion	The supported set of protocols for communicating to this storage controller.	N/A
LongDesc ription	This property shall contain the supported set of protocols for communicating to this storage controller.	N/A
Mandator y	Mandatory	
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.	

6.4.2.2.31 SupportedDeviceProtocols   The mapping for `SupportedDevicePro-`
`tocols` is summarized in Table 98.

Table 98: SupportedDeviceProtocols mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	SupportedDeviceProtocols	N/A
Type	Collection(Protocol.Protoco l)	N/A
Descript ion	The protocols that the storage controller can use to communicate with attached devices.	N/A
LongDesc ription	This property shall contain the set of protocols this storage controller can use to communicate with attached devices.	N/A
Mandator y	Do Not Implement	
Notes		

6.4.3 IO Controller

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

```
{
  "@odata.type":
  ↪  "#StorageController.v1_7_0.StorageController",
  "Id": "NVMeIOController",
  "Name": "NVMe I/O Controller",
  "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.",
  "Status": {
    "State": "Enabled",
    "Health": "OK"
  },
  "Manufacturer": "Best NVMe Vendor",
  "Model": "Simple NVMe Device",
  "SerialNumber": "NVME123456",
  "PartNumber": "NVM44",
  "FirmwareVersion": "1.0.0",
  "SupportedControllerProtocols": [
    "PCIe"
  ],
  "SupportedDeviceProtocols": [
    "NVMe"
  ],
  "SupportedRAIDTypes": [
    "None"
  ],
  "NVMeControllerProperties": {
    "ControllerType": "IO",
    "NVMeVersion": "1.3",
```

```
    "NVMeControllerAttributes": {
      "ReportsUUIDList": false,
      "SupportsSQAssociations": false,
      "ReportsNamespaceGranularity": false,
      "SupportsTrafficBasedKeepAlive": false,
      "SupportsPredictableLatencyMode": false,
      "SupportsEnduranceGroups": false,
      "SupportsReadRecoveryLevels": false,
      "SupportsNVMSets": false,
      "SupportsExceedingPowerOfNonOperationalState": false,
      "Supports128BitHostId": false
    },
    "NVMeSMARTCriticalWarnings": {
      "MediaInReadOnly": false,
      "OverallSubsystemDegraded": false,
      "SpareCapacityWornOut": false
    },
    "MaxQueueSize": 1
  },
  "Links": {
    "AttachedVolumes": [{
      "@odata.id": "/redfish/v1/Systems/Sys-
      ↪  1/Storage/SimplestNVMeSSD/Volumes/SimpleNamespace"
    }]
  },
  "Actions": {
    "#StorageController.SecuritySend": {
      "target": "/redfish/v1/Systems/Sys-
      ↪  1/Storage/SimplestNVMeSSD/Controllers/NVMeIOController/Actions/
      ↪  StorageController.SecuritySend"
    },
    "#StorageController.SecurityReceive": {
      "target": "/redfish/v1/Systems/Sys-
      ↪  1/Storage/SimplestNVMeSSD/Controllers/NVMeIOController/Actions/
      ↪  StorageController.SecurityReceive"
```

```
        }
    },
    "@odata.id": "/redfish/v1/Systems/Sys-
    ↪   1/Storage/SimplestNVMeSSD/Controllers/NVMeIOController",
    "@Redfish.Copyright": "Copyright 2015-2024 SNIA. All
    ↪   rights reserved."
}
```

6.4.3.2 Property Mapping

6.4.3.2.1 Actions.#StorageController.RunSelfTest   The mapping for `Actions.#StorageController.R`
is summarized in Table 99.

Table 99: Actions.#StorageController.RunSelfTest mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Prope rty	Actions.#StorageController.R unSelfTest	**NVM Spec Property / Field:** Device Self-test Command **NVM Spec: Section:Figure** NVM Base Specification 2.0: 5.9 Device Self-test Command
Type	Action (Special form of POST)	NVMe Administrative command
Descr iptio n	This action is used to initiate a self-test against this controller. This action should return either the `ResourceSelfTestFailed` or `ResourceSelfTestCom‐ pleted` event.	The Device Self-test command is used to start a device self-test operation or abort a device self-test operation
LongD escri ption	This action is used to initiate a self-test against this controller. This action should return either the `ResourceSelfTestFailed` or `ResourceSelfTestCom‐ pleted` event.	The Device Self-test command is used to start a device self-test operation or abort a device self-test operation (refer to NVMe 2.0 Base Specification section 8.6).
Manda tory	Optional	Optional

	Redfish/Swordfish	NVMe / NVMe-oF
Notes		The Device Self-test command is used specifically to: a) start a short device self-test operation;b) start an extended device self-test operation;c) start a vendor specific device self-test operation; ord) abort a device self-test operation already in process.

The mapping for `Assembly` is summarized in Table 100.

Table 100: Assembly mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Assembly	N/A
Type	ComplexType	N/A
Descript ion	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

	Redfish/Swordfish	NVMe / NVMe-oF
LongDescription	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
Mandatory	Optional Do Not Implement for NVMe drives, or devices with NVMe front ends, such as opaque arrays.	
Notes		

6.4.3.2.2 AssetTag The mapping for `AssetTag` is summarized in Table 101.

Table 101: Assembly mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	AssetTag	N/A
Type	Edm.String	N/A
Descript ion	The user-assigned asset tag for this storage controller.	N/A
LongDesc ription	This property shall track the storage controller for inventory purposes.	N/A
Mandator y	Optional Do Not Implement for NVMe Drives, or devices with NVMe front ends, such as opaque arrays.	
Notes		

6.4.3.2.3 CacheSummary   The mapping for CacheSummary is summarized in Table 102.

Table 102: CacheSummary mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	CacheSummary	N/A
Type	ComplexType	N/A
Descript ion	The cache memory of the storage controller in general detail.	N/A
LongDesc ription	This property shall contain properties that describe the cache memory for this resource.	N/A
Mandator y	Do Not Implement	
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 103.

Table 103: ControllerRates mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	ControllerRates	N/A
Type	ComplexType	N/A
Descript ion	This property describes the various controller rates used for processes such as volume rebuild or consistency checks.	N/A
LongDesc ription	This object shall contain all the rate settings available on the controller.	N/A
Mandator y	Do Not Implement	
Notes		

6.4.3.2.5 Description   The mapping for `Description` is summarized in Table 104.

Table 104: Description mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Description	N/A
Type	String	N/A
Descript ion	The description of this resource.	N/A
LongDesc ription	This object represents the description of this resource. The resource values shall comply with the Redfish Specification-described requirements.	N/A
Mandator y	Mandatory	
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 105.

Table 105: FirmwareVersion mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	FirmwareVersion	**NVM Spec Property / Field:** Identify Command / Identify Controller Data structure (CNS 01h) / Firmware Revision (FR) **NVM Spec: Section:Figure** NVMe 2.0: Section 5.17.2.1:Figure 275: Bytes 64 - 71
Type	String	String
Descript ion	The firmware version of this storage controller.	The currently active firmware revision for the domain of which this controller is a part.
LongDesc ription	This property shall contain the firmware version as defined by the manufacturer for the associated storage controller.	The currently active firmware revision for the domain of which this controller is a part.
Mandator y	Mandatory	Mandatory
Notes		Return the currently active firmware revision information.

6.4.3.2.7 Identifiers   The mapping for `Identifiers` is summarized in Table 106.

Table 106: Identifiers mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Identifiers	N/A
		N/A
Descript ion	The Durable names for the storage controller.	N/A

	Redfish/Swordfish	NVMe / NVMe-oF
LongDesc ription	This property shall contain a list of all known durable names for the associated storage controller.	N/A
Mandator y	Optional	
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.	

6.4.3.2.8 Identifiers.DurableName   The mapping for `Identifiers.DurableName` is summarized in Table 107.

Table 107: Identifiers.DurableName mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Identifiers.DurableName	N/A
Type	Variable	N/A
Descript ion	The Durable names for the storage controller.	N/A
LongDesc ription	The Durable names for the storage controller.	N/A
Mandator y	Optional	
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.	

6.4.3.2.9 Identifiers.DurableNameFormat   The mapping for `Identifiers.DurableNameFormat` is summarized in Table 108.

Table 108: Identifiers.DurableNameFormat mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Identifiers.DurableNameForm at	N/A
Type	enum (DurableNameFormat)	N/A
Descript ion	The Durable names for the storage controller.	N/A
LongDesc ription	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
Mandator y	Optional	
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.	

6.4.3.2.10 Links.AttachedVolumes   The mapping for `Links.AttachedVolumes` is summarized in Table 109.

Table 109: Links.AttachedVolumes mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Links.AttachedVolumes	**NVM Spec Property / Field:** Identify Command / Active Namespace ID list (CNS 07h) **NVM Spec: Section:Figure** NVMe 2.0: Section 5.17.2.7
Type	Collection(Volume.Volume)	Namespace LIst
Descript ion	An array of links to volumes that are attached to this controller instance.	A list of 1,024 namespace IDs is returned to the host containing active NSIDs in increasing order that are greater than the value specified in the Namespace Identifier (NSID) field of the command.
LongDesc ription	This property shall contain a link to the Resources of type Volume that are attached to this instance of storage controller.	A list of 1,024 namespace IDs is returned to the host containing active NSIDs in increasing order that are greater than the value specified in the Namespace Identifier (NSID) field of the command. The NSID field may be cleared to 0h to retrieve a Namespace List including the namespace starting with NSID of 1h. Reference NVMe 2.0: Section 4.4.2: Figure 135 (Namespace List Format)
Mandator y	Mandatory	Required

	Redfish/Swordfish	NVMe / NVMe-oF
Notes	This contains a pointer to the set of namespaces attached to this IO Controller.	

6.4.3.2.11 Links.Endpoints   The mapping for `Links.Endpoints` is summarized in
Table 110.

Table 110: Links.Endpoints mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Links.Endpoints	N/A
Type	Collection(Endpoint.Endpoin t)	N/A
Descript ion	An array of links to the endpoints that connect to this controller.	N/A
LongDesc ription	This property shall contain an array of links to the Resources of type Endpoint associated with this controller.	N/A
Mandator y	Optional	
Notes	For NVMe-oF configurations.	

6.4.3.2.12 Links.Connections   The mapping for `Links.Connections` is summarized in Table 112.

Table 111: Links.Connections mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Links.Connections	N/A
Type	Collection(Connection.Connection)	N/A
Descript ion	An array of links to volumes that are attached to this controller instance.	N/A
LongDesc ription	This property shall contain a link to the Resources of type Volume that are attached to this instance of storage controller.	N/A
Mandator y	Optional Do Not Implement 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

Table 112: Links.NetworkDeviceFunctions mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Links.NetworkDeviceFunction s	N/A

	Redfish/Swordfish	NVMe / NVMe-oF
Type	Collection(NetworkDeviceFun ction.NetworkDeviceFunction)	N/A
Descript ion	The network device functions that provide connectivity to this controller.	N/A
LongDesc ription	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
Mandator y	Optional Recommended for more complex devices with NVMe front ends, such as opaque arrays.	
Notes	For NVMe-oF configurations.	

6.4.3.2.14 Location   The mapping for `Location` is summarized in Table 113.

Table 113: Location mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Location	N/A
Type	Collection(Resource.Locatio n)	N/A
Descript ion	The location of the storage controller.	N/A
LongDesc ription	This property shall contain location information of the associated storage controller.	N/A

	Redfish/Swordfish	NVMe / NVMe-oF
Mandator y	Optional Do Not Implement for NVM Drives, or more complex devices with NVMe front ends, such as opaque arrays.	
Notes		

6.4.3.2.15 Manufacturer   The mapping for `Manufacturer` is summarized in Table 114.

Table 114: Manufacturer mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Manufacturer	NVM Spec Property / Field:** Identify Command / Identify Controller Datastructure (CNS 01h) / PCI Vendor ID (VID) **NVM Spec: Section: Figure** NVMe 2.0: Section 5.17.2.1: Figure 275: Bytes 00:01
Type	String	16-bit number in little endian format.
Descript ion	The manufacturer of this storage controller.	The company vendor identifier
LongDesc ription	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

	Redfish/Swordfish	NVMe / NVMe-oF
Mandator y Notes	Optional	NVMe: Mandatory NVMe-oF: DNI
		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, 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 115.

Table 115: Model mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Model	**NVM Spec Property / Field:** Identify Command / Identify Controller Datastructure (CNS 01h) / Model Number (MN) **NVM Spec: Section: Figure** NVMe 2.0: Section 5.17.2.1: Figure 275: Bytes 24:63
Type	String	String
Descript ion	The model number for the storage controller.	Model Number (MN)

	Redfish/Swordfish	NVMe / NVMe-oF
LongDescription	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
Mandatory Notes	Recommended	Mandatory

6.4.3.2.17 Name   The mapping for Name is summarized in Table 116.

Table 116: Name mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Name	**NVM Spec Property / Field:** Identify Command / Identify Controller Datastructure (CNS 01h) / Controller ID (CNTLID) **NVM Spec: Section:Figure** NVMe 2.0: Section 5.17.2.1: Figure 275: Bytes 78:79
Type	String	16-bit hex value
Description	The name of the resource or array member.	Controller ID

	Redfish/Swordfish	NVMe / NVMe-oF
LongDesc ription	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 unique controller identifier associated with the controller.
Mandator y	Mandatory	Mandatory
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 NVMeCon-
trollerProperties.ControllerType is summarized in Table 117.

Table 117: NVMeControllerProperties.ControllerType mapping

Redfish/ Swordfis h	NVMe / NVMe-oF	
Property	NVMeControllerProperties.Co ntrollerType	**NVM Spec Property / Field:** Identify Command / Identify Controller Datastructure (CNS 01h) / Controller Type (CNTRL_TYPE) **NVM Spec: Section:Figure** NVMe 2.0: Section 5.17.2.1: Figure 275: Bytes 111
Type	StorageController.v1_0_0. NVMeControllerType	Hex value
Descript ion	This property specifies the type of NVMe Controller.	Controller Type
LongDesc ription	This property shall specify the type of NVMe Controller.	This field specifies the controller type. Supported values: 0h Reserved (controller type not reported)1h I/O controller2h Discovery controller3h Administrative controller
Mandator y	Mandatory	Mandatory
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 `NVMeCon-trollerProperties.NVMeVersion` is summarized in Table 118.

Table 118: NVMeControllerProperties.NVMeVersion mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeControllerProperties.NVMeVersion	**NVM Spec Property / Field:** Identify Command / Identify Controller Datastructure (CNS 01h) / Version (VER) **NVM Spec: Section:Figure** NVMe 2.0: Section 5.17.2.1: Figure 275: Bytes 80:83
Type	String	32-bit value. bits 31:16 indicate the major versionbits 15:08 indicate the minor version bits 07:00 indicate the tertiare version number
Descript ion	The version of the NVMe Base Specification supported.	Version (VER)
LongDesc ription	This property shall specify the type of NVMe Controller.	Indicates the major, minor, and tertiary version of the NVM Express base specification that the controller implementation supports.
Mandator y	Optional	Mandatory
Notes		Implementations compliant to NVM Express Base Specification revision 1.2 or later shall report a non-zero value in this field. Valid versions of the specification are: 1.0, 1.1, 1.2, 1.2.1, 1.3, 1.4, and 2.0.

6.4.3.2.20 **NVMeControllerProperties.NVMeControllerAttributes.ReportsUUIDList**

The mapping for `NVMeControllerProperties.NVMeControllerAttributes.ReportsUUIDList` is summarized in Table 119.

Table 119: NVMeControllerProperties.NVMeControllerAttributes.ReportsUUIDList mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeControllerProperties.NVMeControllerAttributes.ReportsUUIDList	**NVM Spec Property / Field:** Identify Command / Identify Controller Datastructure (CNS 01h) / Controller Attributes (CTRATT) : UUID List (Bit 9) **NVM Spec: Section:Figure** NVMe 2.0: Section 5.17.2.1: Figure 275: Bytes 99:96 (CTRATT): Bit 9 (UUID List)
Type	Boolean	Single bit (bool)
Descript ion	Indicates whether or not the controller supports reporting of a UUID list.	Indicated whether or not the controller supports reporting of a UUID List.
LongDesc ription	This property shall indicate whether or not the controller supports reporting of a UUID list.	Indicated whether or not the controller supports reporting of a UUID List.
Mandator y Notes	Optional	Mandatory for IO Controllers

6.4.3.2.21 **NVMeControllerProperties.NVMeControllerAttributes.SupportsSQAssociations**

The mapping for `NVMeControllerProperties.NVMeControllerAttributes.SupportsSQAssoci`
is summarized in Table 120.

Table 120: SupportsSQAssociations mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeControllerProperties.NV MeControllerAttributes.Supp ortsSQAssociations	**NVM Spec Property / Field:** Identify Command / Identify Controller Datastructure / Controller Attributes (CTRATT): Bit 08 (SQ Associations) **NVM Spec: Section:Figure** NVMe 2.0: Section 5.17.2.1: Figure 275: Bytes 96:99: Bit 08 (SQ Associations)
Type	Boolean	Single bit (bool)
Descript ion	Indicates whether or not the controller supports SQ Associations.	Indicates whether or not the controller supports SQ Associations.
LongDesc ription	This property shall indicate whether or not the controller supports SQ Associations.	This property shall indicate whether or not the controller supports SQ Associations.
Mandator y	Optional	Mandatory
Notes		

6.4.3.2.22 **NVMeControllerProperties.NVMeControllerAttributes.ReportsNamespaceGranularity**

The mapping for NVMeControllerProperties.NVMeControllerAttributes.ReportsNamespace
is summarized in Table 121.

Table 121: NVMeControllerProperties.NVMeControllerAttributes.

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeControllerProperties.NV MeControllerAttributes.Repo rtsNamespaceGranularity	**NVM Spec Property / Field:** Identify Comand / Identify Controller Datastructure (CNS 01h) / Controller Attributes (CTRATT): Bit 07 (Namespace Granularity) **NVM Spec: Section:Figure** NVMe 2.0: Section 5.17.2.1: Figure 275: Bytes 96:99: Bit 07
Type	Boolean	Single bit (bool)
Descript ion	Indicates whether or not the controller supports reporting of Namespace Granularity.	Indicates whether or not the controller supports reporting of Namespace Granularity.
LongDesc ription	This property shall indicate whether or not the controller supports reporting of Namespace Granularity.	This property shall indicate whether or not the controller supports reporting of Namespace Granularity.
Mandator y	Optional Recommended for NVM Drives and more complex devices with NVMe front ends, such as opaque arrays.	Mandatory
Notes		

6.4.3.2.23 **NVMeControllerProperties.NVMeControllerAttributes.SupportsReservations**

The mapping for `NVMeControllerProperties.NVMeControllerAttributes.SupportsReservat`
is summarized in Table 122.

Table 122: NVMeControllerProperties.NVMeControllerAttributes.

	Redfish/Swordfish	NVMe / NVMe-oF
Prope rty	NVMeControllerProperties.NVMe ControllerAttributes.Supports Reservations	**NVM Spec Property / Field:** Identify Command / Identify Controller Datastructure (CNS 01h) / Optional NVM Command Support (ONCS): Bit 05 **NVM Spec: Section:Figure** NVMe 2.0: Section 5.17.2.1: Figure 275: Bytes 520 : 521: Bit 05
Type	Boolean	Single bit (bool)
Descr iptio n	Indicates whether or not the controller supports namespace reservations.	Indicates whether or not the controller supports 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.	Indicates whether or not the controller supports reservations.
Manda tory	Optional Recommended for NVM Drives and more complex devices with NVMe front ends, such as opaque arrays.	Mandatory
Notes		

6.4.3.2.24 **NVMeControllerProperties.NVMeControllerAttributes.SupportsTrafficBasedKeepAlive**

The mapping for NVMeControllerProperties.NVMeControllerAttributes.TrafficBasedKeep is summarized in Table 123.

Table 123: SupportsTrafficBasedKeepAlive mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeControllerProperties.NV MeControllerAttributes.Supp ortsTrafficBasedKeepAlive	**NVM Spec Property / Field:** Identify Command / Identify Controller Datastructure (CNS 01h) / Controller Attributes (CTRATT): Bit 06 - Traffic Based Keep Alive Support (TBKAS) **NVM Spec: Section:Figure** NVMe 2.0: Section 5.17.2.1: Figure 275: Bytes 96 : 99: Bit 06
	Boolean	Single bit (bool)
Descript ion	Indicates whether or not the controller supports restarting KeepAlive Timer if traffic is processed from an admin command or IO during KeepAlive Timeout interval.	Indicates if the controller supports restarting the Keep Alive Timer if an Admin command or an I/O command is processed during the Keep Alive Timeout Interval.
LongDesc ription	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.	controller supports restarting the Keep Alive Timer if an Admin command or an I/O command is processed during the Keep Alive Timeout Interval.

	Redfish/Swordfish	NVMe / NVMe-oF
Mandator y	Optional Mandatory for Ethernet-Attach Drives; Mandatory for more complex devices with NVMe front ends, such as opaque arrays.	Mandatory

6.4.3.2.25 NVMeControllerProperties.NVMeControllerAttributes.SupportsPredictableLatencyMode

The mapping for NVMeControllerProperties.NVMeControllerAttributes.PredictableLater
is summarized in Table 124.

Table 124: NVMeControllerProperties.NVMeControllerAttributes.

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeControllerProperties.NV MeControllerAttributes.Supp ortsPredictableLatencyMode	**NVM Spec Property / Field:** Identify Command / Identify Controller Datastructure (CNS 01h) / Controller Attributes (CTRATT): Bit 05 - (Predictable Latency Mode) **NVM Spec: Section:Figure** NVMe 2.0: Section 5.17.2.1: Figure 275: Bytes 96:99: Bit 05
Type	Boolean	Single bit (bool)
Descript ion	Indicates whether or not the controller supports Predictable Latency Mode.	Indicates if the controller supports Predictable Latency Mode.
LongDesc ription	This property shall indicate whether or not the controller supports Predictable Latency Mode.	Indicates if the controller supports Predictable Latency Mode
Mandator y	Optional	Mandatory
Notes		

6.4.3.2.26 NVMeControllerProperties.NVMeControllerAttributes.SupportsEnduranceGroups

The mapping for `NVMeControllerProperties.NVMeControllerAttributes.EnduranceGroups` is summarized in Table 125.

Table 125: NVMeControllerProperties.NVMeControllerAttributes.

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeControllerProperties.NVMeControllerAttributes.SupportsEnduranceGroups	**NVM Spec Property / Field:** Identify Command / Identify Controller Datastructure (CNS 01h) / Controller Attributes (CTRATT): Bit 04 (Endurance Groups) **NVM Spec: Section:Figure** NVMe 2.0: Section 5.17.2.1: Figure 275: Bytes 96 : 99: Bit 04
Type	Boolean	Single bit (bool)
Descript ion	Indicates whether or not the controller supports Endurance Groups.	Indicates if the controller supports Endurance Groups
LongDesc ription	This property shall indicate whether or not the controller supports Endurance Groups.	Indicates if the controller supports Endurance Groups
Mandator y	Optional Mandatory when EnduranceGroups/Sets are supported.	Mandatory
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.SupportsReadReco
is summarized in Table 126.

Table 126:

NVMeControllerProperties.NVMeControllerAttributes.SupportsReadRecoveryLevels

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeControllerProperties.NV MeControllerAttributes.Supp ortsReadRecoveryLevels	**NVM Spec Property / Field:** Identify Command / Identify Controller Datastructure (CNS 01h) / Controller Attributes (CTRATT): Bit 03 (Read Recovery Levels) **NVM Spec: Section:Figure** NVMe 2.0: Section 5.17.2.1: Figure 275: Bytes 9 6: 99: Bit 03
Type	Boolean	Single bit (bool)
Descript ion	Indicates whether or not the controller supports Read Recovery Levels.	Indicates whether or not the controller supports Read Recovery Levels.
LongDesc ription	This property shall indicate whether or not the controller supports Read Recovery Levels.	Indicates whether or not the controller supports Read Recovery Levels.
Mandator y	Optional	Mandatory
Notes		

6.4.3.2.28 NVMeControllerProperties.NVMeControllerAttributes.SupportsNVMSets

The mapping for `NVMeControllerProperties.NVMeControllerAttributes.SupportsNVMSets` is summarized in Table 127.

Table 127: NVMeControllerProperties.NVMeControllerAttributes.SupportsNVMSets

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeControllerProperties.NVMeControllerAttributes.SupportsNVMSets	**NVM Spec Property / Field:** Identify Command / Identify Controller Datastructure (CNS 01h) / Controller Attributes (CTRATT): Bit 02 (NVM Sets) **NVM Spec: Section:Figure** NVMe 2.0: Section 5.17.2.1: Figure 275: Bytes 96 : 99: Bit 02
Type	Boolean	Single bit (bool)
Descript ion	Indicates whether or not the controller supports NVM Sets.	Indicates whether or not the controller supports NVM Sets.
LongDesc ription	This property shall indicate whether or not the controller supports NVM Sets.	This property shall indicate whether or not the controller supports NVM Sets.
Mandator y	Optional Mandatory when EnduranceGroups/Sets are supported.	Mandatory
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.SupportsExceedingPowerOfNonOperatio

The mapping for `NVMeControllerProperties.NVMeControllerAttributes.SupportsExceedir`
is summarized in Table 128.

Table 128: SupportsExceedingPowerOfNonOperationalState

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeControllerProperties.NVMeControllerAttributes.SupportsExceedingPowerOfNonOperationalState	**NVM Spec Property / Field:** Identify Command / Identify Controller Datastructure (CNS 01h) / Controller Attributes (CTRATT): Bit 01 (Non-Operational Power State Permissive Mode) **NVM Spec: Section:Figure** NVMe 2.0: Section 5.17.2.1: Figure 275: Bytes 96 : 99: Bit 01
Type	Boolean	Single bit (bool)
Descript ion	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.	Indicates whether or not the controller supports host control of whether the controller may temporarily exceed the power of a non-operational power state for the purpose of executing controller initiated background operations in a non-operational power state (i.e., Non-Operational Power State Permissive Mode supported)

	Redfish/Swordfish	NVMe / NVMe-oF
LongDesc ription	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.	The controller supports host control of whether the controller may temporarily exceed the power of a non-operational power state for the purpose of executing controller initiated background operations in a non-operational power state (i.e., Non-Operational Power State Permissive Mode supported)
Mandator y Notes	Optional	Mandatory

6.4.3.2.30 **NVMeControllerProperties.NVMeControllerAttributes.Supports128BitHostId**

The mapping for NVMeControllerProperties.NVMeControllerAttributes.Supports128BitHo
is summarized in Table 129.

Table 129: Supports128BitHostId

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeControllerProperties.NVMeControllerAttributes.Supports128BitHostId	**NVM Spec Property / Field:** Identify Command / Identify Controller (CNS 01h) / Controller Attributes (CTRATT): Bit 00 (Host Identifier Support) **NVM Spec: Section:Figure** NVMe 2.0: Section 5.17.2.1: Figure 275: Bytes 96 : 99: Bit 00
Type	Boolean	Single bit (bool)
Descript ion	Indicates whether or not the controller supports a 128-bit Host Identifier.	Indicates whether or not the controller supports a 128-bit Host Identifier
LongDesc ription	This property shall indicate whether or not the controller supports a 128-bit Host Identifier.	Indicates whether or not the controller supports a 128-bit Host Identifier
Mandator y	Mandatory	Mandatory
Notes		

6.4.3.2.31 NVMeControllerProperties.MaxNamespaceAttachmentsAllowed

	Redfish/Swordfish	NVMe / NVMe-oF
Prope rty	MaxNamespaceAttachmentsAllowe d	**Retrieved in the Identify CommandNVM Spec Property / Field:** Identify Controller Data Structure, I/O Command Set Independent / Maximum I/O Controller Namespace Attachments (MAXCNA) NVMe Base Specification 2.0: Section 5.17.2.1: Figure 275: Identify – Identify Controller Data Structure, I/O Command Set Independent: Bytes 560:563 - Maximum I/O Controller Namespace Attachments (MAXCNA)
Type	Edm.Decimal	Bytes (4)
Descr iptio n	The maximum number of namespace attachments allowed for this namespace.	This field Indicates the maximum number of namespaces that are allowed to be attached to this I/O controller.
LongD escri ption	This property shall contain the maximum number of namespace attachments supported to this IO controller. If there are no limits imposed, this property should not be implemented.	This field Indicates the maximum number of namespaces that are allowed to be attached to this I/O controller.
Optio nal		Optional

Redfish/Swordfish	NVMe / NVMe-oF
Notes	The value of this field shall be less than or equal to the number of namespaces supported by the NVM subsystem - refer to the MNAN field in the NVMe Base Specification 2.0: Section 5.17.2.1: Figure 275: Identify – Identify Controller Data Structure, I/O Command Set Independent: Bytes 540:543 - Maximum Number of Allowed Namespaces (MNAN). If this field is cleared to 0h, then no maximum is specified.

6.4.3.2.32 NVMeControllerProperties.MaxQueueSize   The mapping for NVMeCon-
trollerProperties.MaxQueueSize is summarized in Table 131.

Table 131: NVMeControllerProperties.MaxQueueSize mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeControllerProperties.MaxQueueSize	**NVM Spec Property / Field:** JProperty Get / Controller Properties / Controller Capabilities offset 0h (CAP) / Maximum Queues Entries Supported (MQES) **NVM Spec: Section:Figure** NVMe 2.0; Section 3.1.3.1 Controller Capabilities: Figure 36: Bits 00:15
Type	Int64	16 bit value
Descript ion	Indicates the maximum individual queue size that an NVMe IO Controller supports.	This field indicates the maximum individual queue size that the controller supports.
LongDesc ription	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.	This field indicates the maximum individual queue size that the controller supports. For NVMe over PCIe implementations, this value applies to the I/O Submission Queues and I/O Completion Queues that the host creates. For NVMe over Fabrics implementations, this value applies to only the I/O Submission Queues that the host creates. This is a 0's based value. The minimum value is 1h, indicating two entries.

	Redfish/Swordfish	NVMe / NVMe-oF
Mandator y	Optional Mandatory for more complex devices with NVMe front ends, such as opaque arrays.	Required
Notes		

6.4.3.2.33 NVMeControllerProperties.ANACharacteristics   The mapping for NVMe‑ `ControllerProperties.ANACharacteristics` is summarized in Table 132.

Table 132: NNVMeControllerProperties.ANACharacteristics mapping

	RRedfish/Swordfish	NVMe / NVMe-oF
Property	NVMeControllerProperties.AN ACharacteristics	**NVM Spec Property / Field:** Asymmetric Namespace Access Log (Log ID 0ch) **NVM Spec: Section:Figure** NVMe 2.0: Section 5.16.1.13: Figures 220 and 221
Type	Collection(StorageControlle r.v1_0_0.ANACharacteristi cs)	Log Page
Descript ion	This property contains the combination of ANA type and volume information.	This log consists of a header describing the log and descriptors containing the asymmetric namespace access information for ANA Groups
LongDesc ription	This property shall contain the combination of ANA type and volume information.	This log consists of a header describing the log and descriptors containing the asymmetric namespace access information for ANA Groups that contain namespaces that are attached to the controller processing the command. ANA Group Descriptors shall be returned in ascending ANA Group Identifier order
Mandator y	Optional	Mandatory if ANA Reporting is supported
Notes		

6.4.3.2.34 NVMeControllerProperties.ANACharacteristics.AccessState The mapping for `NVMeControllerProperties.ANACharacteristics.AccessState` is summarized in Table 133.

Table 133: NNVMeControllerProperties.ANACharacteristics.AccessState mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeControllerProperties.ANACharacteristics.AccessStat e	NVM Spec Property / Field:** Asymmetric Namespace Access Log (Log ID 0ch) / ANA Group Descriptors 0 - n-1: Byte 16 Asymmetric Namespace Access State (Bits 00 - 03) **NVM Spec: Section:Figure** NVMe 2.0: Section 5.16.1.13: Figures 220 and 221
Type	StorageController.v1_0_0. ANAAccessState	Reported ANA Access state
Descript ion	Reported ANA Access state.	This field indicates the Asymmetric Namespace Access state for all namespaces in this ANA Group when accessed through this controller.
LongDesc ription	This property shall contain the reported ANA Access State.	Within the Asymmetric Namespace Access Log (Log ID 0ch), for each ANA Group Descriptor the Asymmetric Namespace Access state for all namespaces in this ANA Group when accessed through this controller.
Mandator y	Optional	Mandatory if ANA Reporting is supported

	Redfish/Swordfish	NVMe / NVMe-oF
Notes	Available values: Optimized / NonOptimized / Inacessible / PersistentLoss	Available values: 01h ANA Optimized state02h ANA Non-Optimized state03h ANA Inaccessible state04h ANA Persistent Loss state0Fh ANA Change state

6.4.3.2.35 NVMeControllerProperties.ANACharacteristics.Volume The mapping for `NVMeControllerProperties.ANACharacteristics.Volume` is summarized in Table 134.

Table 134: NNVMeControllerProperties.ANACharacteristics.Volume mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeControllerProperties.AN ACharacteristics.Volume	NVM Spec Property / Field:** Asymmetric Namespace Access Log (Log ID 0ch) / ANA Group Descriptors / Namespace Identifier 0 - n-1 **NVM Spec: Section:Figure** NVMe 2.0: Section 5.16.1.13: Figures 220, 221, and 222
Type	Volume.Volume	ANA Group Descriptor
Descript ion	The specified volume.	The Namespace Identifier
LongDesc ription	This property shall contain a link to the specified volume.	The Namespace Identifier
Mandator y	Optional	Mandatory if ANA Reporting is supported
Notes	This field contains the pointer to the namespace for which the access state applies.	The namespace id should be 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.

6.4.3.2.36 NVMeControllerProperties.NVMeSMARTCriticalWarnings.PRMUnreliable

The mapping for NVMeControllerProperties.NVMeSMARTCriticalWarnings.PMRUnreliable is summarized in Table 135.

Table 135: NNVMeControllerProperties.NVMeSMARTCriticalWarnings.PRMunreliable mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeControllerProperties.NVMeSMARTCriticalWarnings.PMRUnreliable	**NVM Spec Property / Field:** Property Get / Persistent Memory Region Status (PMRSTS) Offset E08h: Bits 09 : 11 - Health Status (HSTS) **NVM Spec: Section:Figure** NVMe 2.0: Section 3.1.3.24: Figure 65: Bits 09 : 11 (Health Status / HSTS), value 011b (Unreliable)
Type	Boolean	Single bit (bool)
Descript ion	The Persistent Memory Region has become unreliable.	Persistent Memory Region has become read-only or unreliable
LongDesc ription	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.	This field shall indicated that the Persistent Memory Region has become read-only or unreliable

	Redfish/Swordfish	NVMe / NVMe-oF
Mandator y	Optional Recommended for NVM Drives; required for more complex devices with NVMe front ends, such as opaque arrays.	Optional
Notes		

6.4.3.2.37 NVMeControllerProperties.NVMeSMARTCriticalWarnings.PowerBackupFailed

The mapping for `NVMeControllerProperties.NVMeSMARTCriticalWarnings.PowerBackupFail`
is summarized in Table 136.

Table 136: NVMeControllerProperties.NVMeSMARTCriticalWarnings.

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeControllerProperties.NV MeSMARTCriticalWarnings.Pow erBackupFailed	**NVM Spec Property / Field:** SMART / Health Information Log Page (Log ID 02h) / Critical Warning / Volatile memory backup device has failed **NVM Spec: Section:Figure** NVMe 2.0: Section 5.16.1.3: Figure 207: Bytes 00, Bits 04
Type	Boolean	Single bit (bool)
Descript ion	Indicates that the volatile memory backup device has failed.	Indicates that the volatile memory backup device has failed.
LongDesc ription	This property shall indicate that the volatile memory backup device has failed.	This property shall indicate that the volatile memory backup device has failed.
Mandator y	Optional Recommended for NVM Drives. Required for more complex devices with NVMe front ends, such as opaque arrays.	Optional
Notes		

6.4.3.2.38 NVMeControllerProperties.NVMeSMARTCriticalWarnings.MediaInReadOnly

The mapping for `NVMeControllerProperties.NVMeSMARTCriticalWarnings.MediaInReadOnly` is summarized in Table 137.

Table 137: NVMeControllerProperties.NVMeSMARTCriticalWarnings.MediaInReadOnly mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeControllerProperties.NVMeSMARTCriticalWarnings.MediaInReadOnly	**NVM Spec Property / Field:** SMART / Health Information Log Page (Log ID 02h) / Critical Warning / All of the media has been placed in read only mode **NVM Spec: Section:Figure** NVMe 2.0: Section 5.16.1.3: Figure 207: Bytes 00, Bits 03
Type	Boolean	Single bit (bool)
Description	Indicates the media has been placed in read only mode.	Indicates all of 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.	This property indicates all of the media has been placed in read only mode. The controller shall not set this bit to "1" if the read-only condition on the media is a result of a change in the write protection state of a namespace
Mandatory	Mandatory	Optional
Notes		

6.4.3.2.39 NVMeControllerProperties.NVMeSMARTCriticalWarnings.OverallSystemDegraded

The mapping for `NVMeControllerProperties.NVMeSMARTCriticalWarnings.OverallSubsyste`
is summarized in Table 138.

Table 138: OverallSystemDegraded mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeControllerProperties.NVMeSMARTCriticalWarnings.OverallSubsystemDegraded	**NVM Spec Property / Field:** SMART / Health Information Log Page (Log ID 02h) / Critical Warning / NVM subsystem reliability has been degraded due to significant media related errors or any internal error that degrades NVM subsystem reliability. **NVM Spec: Section:Figure** NVMe 2.0: Section 5.16.1.3: Figure 207: Bytes 00, Bits 02
Type	Boolean	Single bit (bool)
Descript ion	Indicates that the NVM subsystem reliability has been compromised.	Indicates that NVM subsystem reliability has been degraded.
LongDesc ription	This property shall indicate that the NVM subsystem reliability has been compromised.	NVM subsystem reliability has been degraded due to significant media related errors or any internal error that degrades NVM subsystem reliability.
Mandator y Notes	Mandatory	Optional

6.4.3.2.40 NVMeControllerProperties.NVMeSMARTCriticalWarnings.SpareCapacityWornOut

The mapping for `NVMeControllerProperties.NVMeSMARTCriticalWarnings.SpareCapacityWo`
is summarized in Table 139.

Table 139:

NVMeControllerProperties.NVMeSMARTCriticalWarnings.SpareCapacityWornOut

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeControllerProperties.NVMeSMARTCriticalWarnings.SpareCapacityWornOut	**NVM Spec Property / Field:** SMART / Health Information Log Page (Log ID 02h) / Critical Warning / available spare capacity has fallen below the threshold. **NVM Spec: Section:Figure** NVMe 2.0: Section 5.16.1.3: Figure 207: Bytes 00, Bits 00
Type	Boolean	Single bit (bool)
Descript ion	Indicates that the available spare capacity has fallen below the threshold.	Indicates that the available spare capacity has fallen below the threshold.
LongDesc ription	This property shall indicate that the available spare capacity has fallen below the threshold.	This property shall indicate that the available spare capacity has fallen below the threshold.
Mandator y	Optional Mandatory for NVMe drives, as well as for more complex devices with NVMe front ends, such as opaque arrays.	Optional
Notes		

6.4.3.2.41 PCIeInterface.PCIeType   The mapping for `PCIeInterface.PCIeType` is summarized in Table 140.

Table 140: PCIeInterface.PCIeType mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	PCIeInterface.PCIeType	N/A
Type	enum (PCIeDevice.PCIeType)	N/A
Descript ion	The version of the PCIe specification in use by this device.	N/A
LongDesc ription	This property shall contain the negotiated PCIe interface version in use by this device.	N/A
Mandator y	Optional Mandatory 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.	
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.MaxPCIeType   The mapping for `PCIeInterface.MaxPCIeType` is summarized in Table 141.

Table 141: PCIeInterface.MaxPCIeType mapping

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

6.4.3.2.43 PCIeInterface.LanesInUse   The mapping for `PCIeInterface.LanesInUse` is summarized in Table 142.

Table 142: PCIeInterface.LanesInUse mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	PCIeInterface.LanesInUse	N/A
Type	int64	N/A
Descript ion	The number of PCIe lanes in use by this device.	N/A
LongDesc ription	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
Mandator y	Optional Mandatory for PCIe attach NVMe Drives; do not implement for ethernet-attach drives.	
Notes		The functionality comes from the underlying implementation and does not originate in the NVMe specs

6.4.3.2.44 PCIeInterface.MaxLanes   The mapping for `PCIeInterface.MaxLanes` is summarized in Table 143.

Table 143: PCIeInterface.MaxLanes mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	PCIeInterface.MaxLanes	N/A
Type	int64	N/A
Descript ion	The number of PCIe lanes supported by this device.	N/A
LongDesc ription	This property shall contain the maximum number of PCIe lanes supported by this device.	N/A
Mandator y	Optional Mandatory for PCIe attach NVMe Drives; do not implement for ethernet-attach drives.	
Notes		The functionality comes from the underlying implementation and does not originate in the NVMe specs

6.4.3.2.45 Ports   The mapping for `Ports` is summarized in Table 144.

Table 144: Ports mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Ports	N/A
Type	PortCollection.PortCollecti on	N/A
Descript ion	The link to the collection of ports that exist on the storage controller.	N/A
LongDesc ription	This property shall contain a link to a resource collection of type PortCollection.	N/A
Mandator y	Do Not Implement for NVMe drives Optional for more complex devices with NVMe front ends, such as opaque arrays.	
Notes		

6.4.3.2.46 SKU   The mapping for SKU is summarized in Table 145.

Table 145: SKU mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	SKU	N/A
Type	Edm.String	N/A
Descript ion	The SKU for this storage controller.	N/A
LongDesc ription	This property shall contain the stock-keeping unit number for this storage storage controller.	N/A
Mandator y	Do Not Implement	
Notes		

6.4.3.2.47 SpeedGbps   The mapping for SpeedGbps is summarized in Table 146.

Table 146: SpeedGbps mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	SpeedGbps	N/A
Type	Decimal	N/A
Descript ion	The maximum speed of the storage controller's device interface.	N/A
LongDesc ription	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
Mandator y	Do Not Implement	
Notes		

6.4.3.2.48 Status.State   The mapping for `Status.State` is summarized in Table 147.

Table 147: Status.State mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Status.State	**NVM Spec Property / Field:** Property Get / CSTS – Controller Status (Offset 1ch) **NVM Spec: Section:Figure** NVMe 2.0: Section 3.1.3.6: Figure 47
Type	Resource.State (enum)	32-bit value
Descript ion	The known state of the resource, such as, enabled.	
LongDesc ription	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.	
Mandator y	Mandatory	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), Processing Paused (CSTS.PP) maps to Deferring. If both Ready and Shutdown are indicated, then the system should indicate StandbyOffline. If both Ready and Processing Paused are indicated, then the system should indicate Deferring.

6.4.3.2.49 Status.Health   The mapping for `Status.Health` is summarized in Table 148.

Table 148: Status.Health mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Status.Health	**NVM Spec Property / Field:** Property Get command / Controller Status (CSTS): - Offset 1Ch **NVM Spec: Section:Figure** NVMe 2.0: Section 3.1.3.6: Figure 47**NVM Spec Property / Field:** Get Log Page command / SMART/Health Information Log (Log Identifier 02h)**NVM Spec: Section:Figure** NVMe 2.0: Section 5.16.1.3: Figure 207
Type	Resource.Health	32-bit value
Descript ion	The health state of this resource in the absence of its dependent resources.	This field indicates critical warnings for the state of the controller from both the SMART & general health information log and the Controller Status Property
LongDesc ription	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.	This field indicates critical warnings for the state of the controller from the SMART and general health information . The information provided is over the life of the controller and is retained across power cycles.
Mandator y	Mandatory	Optional

	Redfish/Swordfish	NVMe / NVMe-oF
Notes	Possible Values: OK / Warning / Critical	This comes from CSTS Controller Status. Redfish/Swordfish "OK" corresponds to NVMe/NVMe-oF "Ready" (RDY) bit set to "1" and no warnings in the CSTS or SMART/Health information log pageRedfish/Swordfi sh "Warning" corresponds to "Volatile Memory Backup Device has failed", "NVM subsystem reliability has been degraded", "Temperature greater than or equal to an over temperature threshold", "temperature less than or equal to an under temperature threshold", or "available spare capacity has fallen below the threshold"Redfish/S wordfish "Critical" corresponds to NVMe/NVMe-oF "Controller Fatal Status" (CSTS.CFS) bit set to "1" which indicates that a serious error condition has occurred.

6.4.3.2.50 SupportedControllerProtocols The mapping for `SupportedCon‐` `trollerProtocols` is summarized in Table 149.

Table 149: SupportedControllerProtocols mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	SupportedControllerProtocol s	N/A
Type	Collection(Protocol.Protoco l)	N/A
Descript ion	The supported set of protocols for communicating to this storage controller.	N/A
LongDesc ription	This property shall contain the supported set of protocols for communicating to this storage controller.	N/A
Mandator y	Optional	
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.	

6.4.3.2.51 SupportedDeviceProtocols   The mapping for `SupportedDevicePro-`
`tocols` is summarized in Table 150.

Table 150: SupportedDeviceProtocols mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	SupportedDeviceProtocols	N/A
Type	Collection(Protocol.Protoco l)	N/A
Descript ion	The protocols that the storage controller can use to communicate with attached devices.	N/A
LongDesc ription	This property shall contain the set of protocols this storage controller can use to communicate with attached devices.	N/A
Mandator y Notes	Do Not Implement	

6.5 Namespace

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

6.5.1 Mockup

```
{
  "@odata.type": "#Volume.v1_9_0.Volume",
  "Id": "1",
  "Name": "Namespace 1",
  "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.",
  "DisplayName": "Not set",

  "Status": {
    "State": "Enabled",
    "Health": "OK"
  },
  "Identifiers": [{
    "DurableNameFormat": "NQN",
    "DurableName": "nqn.2014-08.org.nvmexpress:uuid:6c5fe566-
    ↪  10e6-4fb6-aad4-8b4159029384"
  }],
  "RemainingCapacityPercent": 100,
  "BlockSizeBytes": 4096,
  "Capacity": {
    "Data": {
      "ConsumedBytes": 0,
      "AllocatedBytes": 10737418240
      "ProvisionedBytes": 10737418240
    }
  },
```

```
  "RAIDType": "None",
  "NVMeNamespaceProperties": {
    "IsShareable": false,
    "NamespaceId": "0x22F",
    "NamespaceFeatures": {
      "SupportsThinProvisioning": false,
      "SupportsAtomicTransactionSize": false,
      "SupportsDeallocatedOrUnwrittenLBError": false,
      "SupportsNGUIDReuse": false,
      "SupportsIOPerformanceHints": false
    },
    "LBAFormat": {
        "LBAFormatType": "LBAFormat0",
        "RelativePerformance": "Best",
        "LBADataSizeBytes": 4096,
        "LBAMetadataSizeBytes": 0
    },
    "MetadataTransferredAtEndOfDataLBA": false,
    "NVMeVersion": "2.0"
  },
  "Links": {
    "Drives": [{
      "@odata.id": "/red-
      ↪   fish/v1/Chassis/SimplestNVMeSSD/Drives/SimplestNVMeSSD"
    }],
    "Controllers": [{
      "@odata.id": "/redfish/v1/Systems/Sys-
      ↪   1/Storage/SimplestNVMeSSD/Controllers/NVMeIOController"
    }]
  },
  "@odata.id": "/redfish/v1/Systems/Sys-
  ↪   1/Storage/SimplestNVMeSSD/Volumes/SimpleNamespace",
  "@Redfish.Copyright": "Copyright 2015-2024 SNIA. All
  ↪   rights reserved."
}
```

6.5.2 Property Mapping

6.5.2.1 ALUA   The mapping for ALUA is summarized in Table 151.

Table 151: BlockSizeBytes mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	ALUA	N/A
Type	Int64	N/A
Description	ALUA properties for this volume.	N/A
LongDescription	This shall identify the ALUA properties for this volume.	
Mandatory	Optional	N/A
Notes		

6.5.2.2 ALUA.ANAGroupID   The mapping for ALUA.ANAGroupID is summarized in Table 152.

Table 152: ALUA.ANAGroupID mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Prope rty	ANAGroupID	Retrieved in the Identify CommandANA Group ID retrieved in the I/O Command Set Independent Identify Namespace data structure (CNS 08h) or in the Identify Namespace data structure for the specified NSID or the NVM Command Set Identify Namespace Data Structure (CNS 00h)NVMe Base Specification: Section 5.17.2.8: Figure 280: Bytes 07:04 - ANA Group Identifier (ANAGRPID) NVMe NVM Command Set Specification: Section 4.1.5.1: Figure 97: Bytes 92:95 ANA Group Identifier (ANAGRPID)
Type	Edm.Decimal	DWORD (4 bytes)
Descr iptio n	The ANA group id for this volume.	The ANA Group Identifier associated with all namespaces in an ANA Group described by an ANA Group Descriptor. All namespaces in an ANA Group described by an ANA Group Descriptor shall have the same ANA Group ID (ANAGroupID).

	Redfish/Swordfish	NVMe / NVMe-oF
LongD escri ption	This shall contain the ANA group id for this volume. This corresponds to the value in the ANAGroupId field in volume.	The ANA Group Identifier associated with all namespaces in an ANA Group described by an ANA Group Descriptor. When retrieved via the Identify Command (Command dword 11) For NSID other than FFFFFFFFh, this field indicates the ANA Group Identifier of the ANA group of which the namespace is a member. Each namespace that is attached to a controller that supports Asymmetric Namespace Access Reporting (refer to the CMIC field) shall report a valid ANAGRPID.
Manda tory	Optional	Optional
Notes		The ANA Group Identifier (ANAGRPID) for each ANA Group shall be unique within an NVM subsystem. If the controller does not support Asymmetric Namespace Access Reporting, then this field shall be cleared to 0h.

6.5.2.3 BlockSizeBytes   The mapping for `BlockSizeBytes` is summarized in Table 153.

Table 153: BlockSizeBytes mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	BlockSizeBytes	**NVM Spec Property / Field:** Formatted LBA Size (FLBAS) retrieved in some Identify Namespace data structures for the specified NSID or the common namespace capabilities for the NVM Command Set (CNS 00h).Refer to the applicable NVMe I/O Command Set specification for details. For NVMe I/O Command Sets that don't define this field, it is considered reserved **NVM Spec: Section:Figure** NVMe NVM Command Set Specification 1.0b: Section 4.1.5.1: Figure 97: Bytes 26 - Formatted LBA Size (FLBAS) and uses the Number LBA Formats (NBLAF) field defined in Section 4.1.5.1: Figure 97: Bytes 25 (Number of LBA Formats)NVM Express NVM Zoned Namespace Command Set Specification 1.1b: Section A.5: Figure 53 (Size and Capacity Fields): Bytes 26.
Type	Int64	Int 64

	Redfish/Swordfish	NVMe / NVMe-oF
Descript ion	The size, in bytes, of the smallest addressable unit, or block.	The LBA data size & metadata size combination that the namespace has been formatted with.
LongDesc ription	This property shall contain size of the smallest addressable unit of the associated drive or device.	This field indicates the LBA data size & metadata size combination that the namespace has been formatted with.
Mandator y	Mandatory	Only Mandatory / applicable for the NVM Command Set and I/O Command Sets based on the NVM Command Set. Prohibited for all other I/O Command Sets.

Redfish/Swordfish	NVMe / NVMe-oF
Notes	Refer to the applicable NVMe I/O Command Set specification for details. For NVMe I/O Command Sets that don't define this field, it is considered reserved. For NVM Express Command Sets that Support Blocks: Bits 6:5 indicate the most significant 2 bits of the Format Index of the supported LBA Format indicated in this data structure that was used to format the namespace. If the NLBAF field is less than or equal to 16, then the host should ignore these bits.Bit 4 indicates whether or not the metadata is transferred at the end of the data LBA, creating an extended data LBA. (note: Bit 4 is not applicable when there is no metadata).Bits 3:0 indicate the least significant 4 bits of the Format Index of the supported LBA Format indicated in this data structure that was used to format the namespace.

6.5.2.4 Capacity.Data.ConsumedBytes   The mapping for `Capacity.Data.ConsumedBytes` is summarized in Table 154.

Table 154: Capacity.Data.ConsumedBytes mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Capacity.Data.ConsumedBytes	**NVM Spec Property / Field:** Namespace Utilization (NUSE) retrieved in some Identify Namespace data structures for the specified NSID or the common namespace capabilities for the NVM Command Set (CNS 00h).Refer to the applicable NVMe I/O Command Set specification for details. For NVMe I/O Command Sets that don't define this field, it is considered reserved**NVM Spec: Section:Figure** NVMe NVM Command Set Specification 1.0b: Section 4.1.5.1: Figure 97: Bytes 16 - 23 - Namespace Utilization (NUSE).NVM Express Key Value Command Set Specification 1.0b: Section 4.1.5.1 - I/O Command Set specific Identify Namespace data structure (CNS 05h, CSI 01h): Figure 36: Bytes 16 -23 (Namespace Utilization)
Type	Int64	Int 64
Descript ion	The number of bytes consumed in this data store for this data type.	The current number of logical blocks allocated in the namespace.

	Redfish/Swordfish	NVMe / NVMe-oF
LongDesc ription	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. For KV Command set: This field indicates the total size of the namespace in bytes. This is the space to store KV keys and KV values. This field is undefined prior to the namespace being formatted.
Mandator y	Mandatory	Only Mandatory / applicable for the NVM Command Set and I/O Command Sets based on the NVM Command Set. Prohibited for all other I/O Command Sets.
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.	Refer to the applicable NVMe I/O Command Set specification for details. For NVMe I/O Command Sets that don't define this field, it is considered reserved

6.5.2.5 Capacity.Data.ProvisionedBytes   The mapping for `Capacity.Data.ProvisionedBytes` is summarized in Table 155.

Table 155: Capacity.Data.ProvisionedBytes mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Capacity.Data.ProvisionedBy tes	**NVM Spec Property / Field:** Namespace Size (NSZE) retrieved in some Identify Namespace data structures for the specified NSID or the common namespace capabilities for the NVM Command Set (CNS 00h).Refer to the applicable NVMe I/O Command Set specification for details. For NVMe I/O Command Sets that don't define this field, it is considered reserved**NVM Spec: Section:Figure** NVMe NVM Command Set Specification 1.0b: Section 4.1.5.1: Figure 97: Bytes 00 - 07 - Namespace Size (NSZE).NVM Express NVM Zoned Namespace Command Set Specification 1.1b: Section A.5: Figure 53: Bytes 0:7NVM Express Key Value Command Set Specification 1.0b: Section 4.1.5.1: Figure 36: Bytes 00:07
Type	Int64	Int 64
Descript ion	The maximum number of bytes that can be allocated in this data store for this data type.	The total size of the NVM allocated to this namespace.

	Redfish/Swordfish	NVMe / NVMe-oF
LongDesc ription	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 blocks. A namespace of size n consists of LBA 0 through (n - 1). The number of logical blocks is based on the formatted LBA size. For ZNS: (NSZE-1) indicates the highest possible LBA in the zoned namespace.For KV: This is the space to store KV keys and KV values. This field is undefined prior to the namespace being formatted.
Mandator y	Optional	Only Mandatory / applicable for the NVM Command Set and I/O Command Sets based on the NVM Command Set. Prohibited for all other I/O Command Sets.
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.	Refer to the applicable NVMe I/O Command Set specification for details. For NVMe I/O Command Sets that don't define this field, it is considered reserved

6.5.2.6 Capacity.Data.AllocatedBytes   The mapping for `Capacity.Data.AllocatedBytes` is summarized in Table 156.

Table 156: Capacity.Data.AllocatedBytes mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Capacity.Data.AllocatedByte s	**NVM Spec Property / Field:** Namespace Capacity (NCAP) retrieved in some Identify Namespace data structures for the specified NSID or the common namespace capabilities for the NVM Command Set (CNS 00h).Refer to the applicable NVMe I/O Command Set specification for details. For NVMe I/O Command Sets that don't define this field, it is considered reserved**NVM Spec: Section:Figure** NVMe NVM Command Set Specification 1.0b: Section 4.1.5.1: Figure 97: Bytes 08 - 15 - Namespace Capacity (NCAP).NVM Express NVM Zoned Namespace Command Set Specification 1.1b: Section A.5: Figure 53 (Size and Capacity Fields): Bytes 08 - 15 (Namespace Capacity).
Type	Int64	Int 64
Descript ion	The number of bytes currently allocated by the storage system in this data store for this data type.	The maximum number of logical blocks that may be allocated in the namespace at any point in time.

	Redfish/Swordfish	NVMe / NVMe-oF
LongDesc ription	The value shall be the number of bytes currently allocated by the storage system in this data store for this data type.	For NVM Command Set: The maximum number of logical blocks that may be allocated in the namespace at any point in timeFor Zoned Namespace Command Set: The maximum number of allocatable logical blocks in the zoned namespace.
Mandator y	Optional	Only Mandatory / applicable for the NVM Command Set and I/O Command Sets based on the NVM Command Set. Prohibited for all other I/O Command Sets.
Notes	Reporting capacity in bytes is the Redfish and Swordfish standard mechanism. Not required when creating a namespace; service implementations will provide a suitable value. For non-thin-provisioned volumes, this will typically be equal to `ProvisionedBytes`.	Refer to the applicable NVMe I/O Command Set specification for details. For NVMe I/O Command Sets that don't define this field, it is considered reserved

6.5.2.7 Capacity.Metadata   The mapping for `Capacity.Metadata` is summarized in Table 157.

Table 157: Capacity.Metadata.AllocatedBytes mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Capacity.Metadata	N/A
Type	Int64	N/A

	Redfish/Swordfish	NVMe / NVMe-oF
Descript ion	The number of bytes currently allocated by the storage system in this data store for this data type.	N/A
LongDesc ription	The value shall be the number of bytes currently allocated by the storage system in this data store for this data type.	N/A
Mandator y	Optional	
Notes	Not required for NVMe Drives; optional to implement for more complex devices.	Do not return metadata information for NVMe devices. This is included in the overall reported capacity information.

6.5.2.8 CapacitySources   The mapping for `CapacitySources` is summarized in Table [158].

Table 158: CapacitySources mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	CapacitySources	See notes
Type	Collection(Capacity.Capacit ySource)	See notes
Descript ion	An array of space allocations to this volume.	See notes
LongDesc ription	Fully or partially consumed storage from a source resource. Each entry provides capacity allocation information from a named source resource.	See notes.

	Redfish/Swordfish	NVMe / NVMe-oF
Mandator y	No	No
Notes	Contains the information about the providing capacity (e.g, EnduranceGroup) for this namespace. Each entry in the collection is mapped such that the CapacitySource instance contains the overall capacity (in bytes) and the types and pointers to the underlying capacity sources.	For each entry in the CapacitySource collection, map the information from the NVMe specification: **NVM Spec Property / Field:** Supported Capacity Configuration List (Log Identifier 11h)**NVM Spec: Section:Figure** NVMe Base Specification 2.0: Section 5.16.1.17: Figures 250 and 251.

6.5.2.9 Description   The mapping for `Description` is summarized in Table 159.

Table 159: Description mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Description	N/A
Type	String	N/A
Descript ion	The description of this resource.	N/A
LongDesc ription	This object represents the description of this resource. The resource values shall comply with the Redfish Specification-described requirements.	N/A
Mandator y	Mandatory	

	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."

6.5.2.10 DisplayName   The mapping for `DisplayName` is summarized in Table 160.

Table 160: DisplayName mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	DisplayName	**NVM Spec Property / Field:** Namespace Admin Label (1Fh) **NVM Spec: Section:Figure** NVMe 2.0d: Section 5.37.1.26, Figure 360
Type	String	UTF-8
Descript ion	A user-configurable string to name the volume.	The Namespace Admin Label feature provides the ability to set and get the Namespace Admin Label for a namespace.

	Redfish/Swordfish	NVMe / NVMe-oF
LongDesc ription	This property shall contain a user-configurable string to name the volume.	This field contains the Namespace Admin Label for the namespace as a null-terminated UTF 8 string. The default value of this Feature is all nulls (i.e., all bytes cleared to 0h). Sanitize operations (refer to section 8.30) affect the values of this Feature; any successful sanitize operation shall modify this Feature by resetting both the saved value and the current value to the default value.
Mandator y	Optional Recommended for NVMe Drives, as well as for more complex devices with NVMe front ends, such as opaque arrays.	Optional
Notes	This contains an (end) user settable "friendly" name for the namespace. In previous versions of the mapping guide, this may have mapped to a proprietary value, but switched to mapping to the Namespace Admin Label in v1.2.7.	

6.5.2.11 Identifiers   The mapping for `Identifiers` is summarized in Table 161.

Table 161: Identifiers mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Identifiers	**NVM Spec Property / Field:** Identify / Namespace Identification Descriptor list (CNS 03h)**NVM Spec: Section:Figure** NVMe 2.0: Section 5.17.2.3, Figure 277
Type	Collection(Resource.Identif ier)	A variable length Namespace Identification Descriptor structures
Descript ion	The Durable names for the storage controller.	A list of Namespace Identification Descriptor structures containing Namespace Type (i.e., IEEE Extended Unique Identifier, Namespace Globally Unique Identifier, Namespace UUID, Command Set Identifier), Namespace Identifier Length (NIDL), and Namespace ID (NID).

	Redfish/Swordfish	NVMe / NVMe-oF
LongDesc ription	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.
Mandator y	Optional	Yes
Notes	This is an array of unique identifiers for the NVM Subsystem including Namespace Type and Namespace ID.	

6.5.2.12 Identifiers.DurableName   The mapping for `Identifiers.DurableName` is summarized in Table 162.

Table 162: Identifiers.DurableName mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Identifiers.DurableName	**NVM Spec Property / Field:** Identify / Namespace Identification Descriptor list (CNS 03h) / Namespace Identifier (NID)**NVM Spec: Section:Figure** NVMe 2.0: Section 5.17.2.3, Figure 277: Bytes 04 : (NIDL + 3)
Type	Variable - see notes	Variable: See notes
Descript ion	The Durable names for the storage controller.	This field contains a value that is globally unique and assigned to the namespace when the namespace is created.
LongDesc ription	This property shall contain a list of all known durable names for the Namespace.	This field contains a value that is globally unique and assigned to the namespace when the namespace is created. This field remains fixed throughout the life of the namespace and is preserved across namespace and controller operations (e.g., Controller Level Reset, namespace format, etc.). The type of the value is specified by the Namespace Identifier Type (NIDT) field, and the size is specified by the Namespace Identifier Length (NIDL) field.
Mandator y	Optional	Yes

	Redfish/Swordfish	NVMe / NVMe-oF
Notes	This is an array of unique identifiers for the Namespace. Type and length of descriptor are in the corresponding Identifiers.DurableNameForm at property.	The type and length of the descriptor are in the corresponding Identifiers.DurableNameForm at property

6.5.2.13 Identifiers.DurableNameFormat   The mapping for `Identifiers.DurableNameFormat` is summarized in Table 163.

Table 163: Identifiers.DurableNameFormat mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Identifiers.DurableNameForm at	**NVM Spec Property / Field:** Identify / Namespace Identification Descriptor list (CNS 03h) / Namespace Identifier Type (NIDT)**NVM Spec: Section:Figure** NVMe 2.0: Section 5.17.2.3, Figure 277: Bytes 00
Type	Int64	Int64
Descript ion	The Durable names for the storage controller.	The Namespace Identifier data type
LongDesc ription	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.
Mandator y	Optional	Yes

	Redfish/Swordfish	NVMe / NVMe-oF
Notes	This is an array of types for the unique identifiers for the NVM Subsystem. Values may be "EUI64", "NGUID", or "UUID".	Allowed values:1h = an 8-byte IEEE Extended Unique Identifier2h = a 10-byte Namespace Globally Unique Identifier.3h = an 8-byte Namespace UUID.4h = the Command Set Identifier (the command set that operates on this namespace).

6.5.2.14 InitializeMethod   The mapping for `InitializeMethod` is summarized in Table 164.

Table 164: InitializeMethod mapping

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

	Redfish/Swordfish	NVMe / NVMe-oF
LongDesc ription	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.	N/A
Mandator y	Optional 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. This maps to functionality corresponding to traditional drive capabilities outside the scope of NVMe.

6.5.2.15 Links.Controllers   The mapping for `Links.Controllers` is summarized in Table 165.

Table 165: Links.Controllers

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Links.Controllers	N/A
Type	Collection(StorageControlle r.StorageController)	N/A
Descript ion	An array of the Controllers associated with this volume.	N/A

	Redfish/Swordfish	NVMe / NVMe-oF
LongDesc ription	This parameter shall contain an array of the controllers (of type StorageController) associated with this volume. When the volume is of type NVMe, these may be both the physical and logical controller representations.	N/A
Mandator y	Optional Mandatory for NVMe Drives and well as for more complex devices with an NVMe front end, such as opaque arrays.	
Notes	This array shall contain links to the controllers for this namespace.	These are the Redfish constructs that correspond to the Admin and IO Controllers associated with this namespace.

6.5.2.16 Links.Drives   The mapping for `Links.Drives` is summarized in Table 166.

Table 166: Links.Drives mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Links.Drives	N/A
Type	Collection(Drive.Drive)	N/A
Descript ion	An array of the drives to be used by the volume	N/A
LongDesc ription	This parameter shall contain an array of the drives to be used by the volume.	N/A

	Redfish/Swordfish	NVMe / NVMe-oF
Mandator y	Optional Mandatory for NVMe Drives. 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.	
Notes	This array shall contain links to the drive object for this namespace.	Redfish construct. Not in the NVMe spec.

6.5.2.17 LogicalUnitNumber The mapping for `LogicalUnitNumber` is summarized in Table 167.

Table 167: LogicalUnitNumber mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	LogicalUnitNumber	N/A
Type	Int64	N/A
Descript ion	Indicates the host-visible LogicalUnitNumber assigned to this Volume.	N/A
LongDesc ription	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
Mandator y	Optional	

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

6.5.2.18 MaxBlockSizeBytes   The mapping for `MaxBlockSizeBytes` is summarized in Table 168.

Table 168: MaxBlockSizeBytes mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	MaxBlockSizeBytes	**NVM Spec Property / Field:** Formatted LBA Size (FLBAS) retrieved in some Identify Namespace data structures for the specified NSID or the common namespace capabilities for the NVM Command Set (CNS 00h).Refer to the applicable NVMe I/O Command Set specification for details. For NVMe I/O Command Sets that don't define this field, it is considered reserved**NVM Spec: Section:Figure** NVMe NVM Command Set Specification 1.0b: Section 4.1.5.1: Figure 97: Bytes 26 - Formatted LBA Size (FLBAS) and uses the Number LBA Formats (NBLAF) field defined in Section 4.1.5.1: Figure 97: Bytes 25 (Number of LBA Formats)NVM Express NVM Zoned Namespace Command Set Specification 1.1b: Section A.5: Figure 53 (Size and Capacity Fields): Bytes 26.
Type	Int64	
Descript ion	The size, in bytes, of the smallest addressable unit, or block.	The LBA data size & metadata size combination that the namespace has been formatted with.

	Redfish/Swordfish	NVMe / NVMe-oF
LongDesc ription	This property shall contain size of the smallest addressable unit of the associated drive or device.	This field indicates the LBA data size & metadata size combination that the namespace has been formatted with.
Mandator y	Optional Recommended to not implement; however, if implemented this should have the same value as BlockSizeBytes.	Only Mandatory / applicable for the NVM Command Set and I/O Command Sets based on the NVM Command Set. Prohibited for all other I/O Command Sets.

Redfish/Swordfish	NVMe / NVMe-oF
Notes	Refer to the applicable NVMe I/O Command Set specification for details. For NVMe I/O Command Sets that don't define this field, it is considered reserved. 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. For NVM Express Command Sets that Support Blocks: Bits 6:5 indicate the most significant 2 bits of the Format Index of the supported LBA Format indicated in this data structure that was used to format the namespace. If the NLBAF field is less than or equal to 16, then the host should ignore these bits.Bit 4 indicates whether or not the metadata is transferred at the end of the data LBA, creating an extended data LBA. (note: Bit 4 is not applicable when there is no metadata).Bits 3:0 indicate the least significant 4 bits of the Format Index of the supported LBA Format indicated in this data structure that was used to format the namespace.

6.5.2.19 Name   The mapping for Name is summarized in Table 169.

Table 169: Name mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Name	N/A
Type	String	N/A
Descript ion	The name of the resource or array member.	N/A
LongDesc ription	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.	N/A
Mandator y	Mandatory	
Notes	In Redfish, Name is a read-only field.	

6.5.2.20 NVMeNamespaceProperties.NamespaceId   The mapping for NVMe-
`NamespaceProperties.NamespaceId` is summarized in Table 170.

Table 170: NVMeNamespaceProperties.NamespaceId mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeNamespaceProperties.Nam espaceId	**NVM Spec Property / Field:** Identify / Namespace Identification Descriptor list (CNS 03h) / Namespace Identifier (NID)**NVM Spec: Section:Figure** NVMe 2.0: Section 5.17.2.3, Figure 277: Bytes 04 : (NIDL + 3)
Type	String	Variable: See notes
Descript ion	The NVMe Namespace Identifier for this namespace.	This field contains a value that is globally unique and assigned to the namespace when the namespace is created.
LongDesc ription	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.	This field contains a value that is globally unique and assigned to the namespace when the namespace is created. This field remains fixed throughout the life of the namespace and is preserved across namespace and controller operations (e.g., Controller Level Reset, namespace format, etc.). The type of the value is specified by the Namespace Identifier Type (NIDT) field, and the size is specified by the Namespace Identifier Length (NIDL) field.
Mandator y	Mandatory	Yes

	Redfish/Swordfish	NVMe / NVMe-oF
Notes		The type and length of the descriptor are in the corresponding Identifiers.DurableNameForm at property

6.5.2.21 NVMeNamespaceProperties.IsBootCapable The mapping for NVMe-NamespaceProperties.IsBootCapable is summarized in Table 171.

Table 171: NVMeNamespaceProperties.IsBootCapable mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Prope rty	NVMeNamespaceProperties.IsBoo tCapable	**NVM Spec Property / Field:** Property Get Command / Controller Capabilities - CAP (Offset 00h) **NVM Spec: Section:Figure** NVMe 2.0: Section 3.1.3.1 (Controller Capabilities), Figure 36: Bits 45 - Boot Partition Support (BPS)
Type	Boolean	Single bit (bool)
Descr iptio n	This property indicates whether or not the Volume contains a boot image and is capable of booting.	This bit indicates whether the controller supports Boot Partitions

	Redfish/Swordfish	NVMe / NVMe-oF
LongDescription	This property shall indicate whether or not the Volume contains a boot image and is capable of booting. This property may be settable by an admin or client with visibility into the contents of the volume. This property should only be set to true when VolumeUsage is either not specified, or when VolumeUsage is set to Data or SystemData.	This bit indicates whether the controller supports Boot Partitions. If this bit is set to "1", the controller supports Boot Partitions. If this bit is cleared to "0", the controller does not support Boot Partitions
Mandatory	Optional	No
Notes		

6.5.2.22 NVMeNamespaceProperties.IsShareable   The mapping for NVMeNamespaceProperties.IsShareable is summarized in Table 172.

Table 172: NVMeNamespaceProperties.IsShareable mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeNamespaceProperties.IsShareable	**NVM Spec Property / Field:** Identify command / I/O Command Set Independent Identify Namespace data structure (CNS 08h) / Namespace Multi-path I/O and Namespace Sharing Capabilities (NMIC) **NVM Spec: Section:Figure** NVMe 2.0: Section 5.17.2.8 (I/O Command Set Independent Identify Namespace Data Structure), Figure 280: Byte 01 - Namespace Multi-path I/O and Namespace Sharing Capabilities (NMIC)
Type	Boolean	Single bit (bool)
Descript ion	Indicates the namespace is shareable.	Indicates if the namespace is a private namespace and is able to be attached to only one controller at a time or if the namespace may be attached to two or more controllers in the NVM subsystem concurrently (i.e., may be a shared namespace)

	Redfish/Swordfish	NVMe / NVMe-oF
LongDesc ription	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.
Mandatory Notes	Optional	No

6.5.2.23 NVMeNamespaceProperties.NamespaceFeatures.SupportsThinProvisioning

The mapping for NVMeNamespaceProperties.NamespaceFeatures.SupportsThinProvisioning is summarized in Table 6.5.2.23.

Table 173: NVMeNamespaceProperties.NamespaceFeatures.

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeNamespaceProperties.NamespaceFeatures.SupportsThin Provisioning	**NVM Spec Property / Field:** Namespace Features (NSFEAT) retrieved in some Identify Namespace data structures for the specified NSID or the common namespace capabilities for the NVM Command Set (CNS 00h).Refer to the applicable NVMe I/O Command Set specification for details. For NVMe I/O Command Sets that don't define this field, it is considered reserved**NVM Spec: Section:Figure** NVMe NVM Command Set Specification 1.0b: Section 4.1.5.1: Figure 97: Bytes 24 - Namespace Features (NSFEAT), Bit 0 (ThinP).
Type	Boolean	Single bit (bool)
Descript ion	This property indicates whether or not the NVMe Namespace supports thin provisioning.	Indicates that the namespace supports thin provisioning
LongDesc ription	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.
Mandator y	Mandatory	Yes

	Redfish/Swordfish	NVMe / NVMe-oF
Notes		Only Mandatory / applicable for the NVM Command Set and I/O Command Sets based on the NVM Command Set. Prohibited for all other I/O Command Sets.

SupportsThinProvisioning mapping

6.5.2.24 NVMeNamespaceProperties.NamespaceFeatures.SupportsDeallocatedOrUnwrittenLBError

The mapping for `NVMeNamespaceProperties.NamespaceFeatures.SupportsDeallocatedOrUnw` is summarized in Table 6.5.2.24.

Table 174: NVMeNamespaceProperties.NamespaceFeatures.

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeNamespaceProperties.NamespaceFeatures.SupportsDeallocatedOrUnwrittenLBError	**NVM Spec Property / Field:** Namespace Features (NSFEAT) retrieved in some Identify Namespace data structures for the specified NSID or the common namespace capabilities for the NVM Command Set (CNS 00h).Refer to the applicable NVMe I/O Command Set specification for details. For NVMe I/O Command Sets that don't define this field, it is considered reserved**NVM Spec: Section:Figure** NVMe NVM Command Set Specification 1.0b: Section 4.1.5.1: Figure 97: Bytes 24 - Namespace Features (NSFEAT), Bit 2 (DAE).
Type	Boolean	Single bit (bool)
Descript ion	This property indicates that the controller supports deallocated or unwritten logical block error for this namespace.	Indicates if the controller supports the Deallocated or Unwritten Logical Block error for this namespace.
LongDescription	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.

	Redfish/Swordfish	NVMe / NVMe-oF
Mandator y	Mandatory	Yes
Notes		Only Mandatory / applicable for the NVM Command Set and I/O Command Sets based on the NVM Command Set. Prohibited for all other I/O Command Sets.

SupportsDeallocatedOrUnwrittenLBError mapping

6.5.2.25 NVMeNamespaceProperties.NamespaceFeatures.SupportsNGUIDReuse

The mapping for NVMeNamespaceProperties.NamespaceFeatures.SupportsNGUIDReuse is summarized in Table 175.

Table 175: NVMeNamespaceProperties.NamespaceFeatures.SupportsNGUIDReuse
mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeNamespaceProperties.NamespaceFeatures.SupportsNGUIDReuse	**NVM Spec Property / Field:** Namespace Features (NSFEAT) retrieved in some Identify Namespace data structures for the specified NSID or the common namespace capabilities for the NVM Command Set (CNS 00h).Refer to the applicable NVMe I/O Command Set specification for details. For NVMe I/O Command Sets that don't define this field, it is considered reserved**NVM Spec: Section:Figure** NVMe NVM Command Set Specification 1.0b: Section 4.1.5.1: Figure 97: Bytes 24 - Namespace Features (NSFEAT), Bit 3 (UIDRESUSE). NVMe KV Command Set Specification 1.0b: Section 4.1.5.1: Figure 36: Bytes 24 - Namespace Feaetures (NSFEAT), Bit 3 (UIDRESUSE).
Type	Boolean	Single bit (bool)
Descript ion	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
LongDesc ription	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 new namespace created after this namespace is deleted. This bit shall be cleared to "0" if both NGUID and EUI64 fields are cleared to 0h.
Mandator y	Mandatory	Yes
Notes		Only Mandatory / applicable for the NVM Command Set and I/O Command Sets based on the NVM Command Set, and the KV ommand Set. Prohibited for all other I/O Command Sets.

6.5.2.26 NVMeNamespaceProperties.NamespaceFeatures.SupportsAtomicTransactionSize

The mapping for NVMeNamespaceProperties.NamespaceFeatures.SupportsAtomicTransactic is summarized in Table 6.5.2.26.

Table 176:

NVMeNamespaceProperties.NamespaceFeatures.SupportsAtomicTransactionSize

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeNamespaceProperties.Nam espaceFeatures.SupportsAtom icTransactionSize	**NVM Spec Property / Field:** Namespace Features (NSFEAT) retrieved in some Identify Namespace data structures for the specified NSID or the common namespace capabilities for the NVM Command Set (CNS 00h).Refer to the applicable NVMe I/O Command Set specification for details. For NVMe I/O Command Sets that don't define this field, it is considered reserved**NVM Spec: Section:Figure** NVMe NVM Command Set Specification 1.0b: Section 4.1.5.1: Figure 97: Bytes 24 - Namespace Features (NSFEAT), Bit 4 (OPTPERF).
Type	Boolean	Single bit (bool)
Descript ion	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
LongDesc ription	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.
Mandator y	Mandatory	Yes
Notes		Only Mandatory / applicable for the NVM Command Set and I/O Command Sets based on the NVM Command Set. Prohibited for all other I/O Command Sets.

SupportsAtomicTransactionSize mapping

6.5.2.27 NVMeNamespaceProperties.NamespaceFeatures.SupportsIOPerformanceHints

The mapping for NVMeNamespaceProperties.NamespaceFeatures.SupportsIOPerformanceHin is summarized in Table 6.5.2.27.

Table 177:

NVMeNamespaceProperties.NamespaceFeatures.SupportsIOPerformanceHints

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeNamespaceProperties.NamespaceFeatures.SupportsIOPerformanceHints	**NVM Spec Property / Field:** Namespace Features (NSFEAT) retrieved in some Identify Namespace data structures for the specified NSID or the common namespace capabilities for the NVM Command Set (CNS 00h).Refer to the applicable NVMe I/O Command Set specification for details. For NVMe I/O Command Sets that don't define this field, it is considered reserved**NVM Spec: Section:Figure** NVMe NVM Command Set Specification 1.0b: Section 4.1.5.1: Figure 97: Bytes 24 - Namespace Features (NSFEAT), Bit 1 (NSABP).
Type	Boolean	Single bit (bool)
Descript ion	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
LongDesc ription	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
Mandator y	Mandatory	Yes
Notes		Only Mandatory / applicable for the NVM Command Set and I/O Command Sets based on the NVM Command Set. Prohibited for all other I/O Command Sets.

SupportsIOPerformanceHints mapping

6.5.2.28 NVMeNamespaceProperties.LBAFormat.LBAFormatType   The   mapping for NVMeNamespaceProperties.LBAFormat.LBAFormatType is summarized in Table 178.

Table 178: 176: NVMeNamespaceProperties.LBAFormat.LBAFormatType mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Prope rty	NVMeNamespaceProperties.LBAFo rmat.LBAFormatType	**NVM Spec Property / Field:** Retrieved in the Identify Namespace data structure for the specified NSID or the common namespace capabilities for the NVM Command Set (CNS 00h). Refer to the applicable NVMe I/O Command Set specification for details. For NVMe I/O Command Sets that don't define this field, it is considered reserved**NVM Spec: Section:Figure** NVMe NVM Command Set Specification 1.0b: Section 4.1.5.1: Figure 97 (Identify Namespace Data Structure, NVM Command Set, NVM Command Set Specific): Byte 26 - Formatted LBA Size (FLBAS).
Type	String	2 bits
Descr iptio n	indicates the LBA data size & metadata size combination that the namespace has been formatted with.	Indicates the LBA data size & metadata size combination that the namespace has been formatted with.

	Redfish/Swordfish	NVMe / NVMe-oF
LongD escri ption	This field indicates the LBA data size & metadata size combination that the namespace has been formatted with.	This field indicates the LBA data size & metadata size combination that the namespace has been formatted with.Bits 7 is reserved.Bits 6:5 indicate the most significant 2 bits of the Format Index of the supported LBA Format indicated in this data structure that was used to format the namespace. If the NLBAF field is less than or equal to 16, then the host should ignore these bits.Bit 4 if set to "1" indicates that the 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. Bit 4 is not applicable when there is no metadata.Bits 3:0 indicate the least significant 4 bits of the Format Index of the supported LBA Format indicated in this data structure that was used to format the namespace
Manda tory	Mandatory	Yes
Notes		Only Mandatory / applicable for the NVM Command Set and I/O Command Sets based on the NVM Command Set. Prohibited for all other I/O Command Sets.

6.5.2.29 NVMeNamespaceProperties.LBAFormat.RelativePerformance The mapping for NVMeNamespaceProperties.LBAFormat.RelativePerformance is summarized in Table 179.

Table 179: 177: NVMeNamespaceProperties.LBAFormat.RelativePerformance mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Prope rty	NVMeNamespaceProperties.LBAFo rmat.RelativePerformance	**NVM Spec Property / Field:** Retrieved in the Identify Namespace data structure for the specified NSID or the common namespace capabilities for the NVM Command Set (CNS 00h). This property may be found by retrieving the LBA Format data structure which starts at the LBAFormat offset for which this Namespace was formatted.Refer to the applicable NVMe I/O Command Set specification for details. For NVMe I/O Command Sets that don't define this field, it is considered reserved**NVM Spec: Section:Figure** NVMe NVM Command Set Specification 1.0b: Section 4.1.5.1: Figure 98 (LBA Format Data Structure, NVM Command Set Specific): Bits 24:25 - Relative Performance (RP).
Type	String	2 bits
Descr iptio n	The relative performance of the LBA format indicated relative to other LBA formats supported by the controller.	The relative performance of the LBA format indicated relative to other LBA formats supported by the controller.

	Redfish/Swordfish	NVMe / NVMe-oF
LongD escri ption	This field indicates the relative performance of the LBA format indicated relative to other LBA formats supported by the controller ("Best", "Better", "Good", "Degraded"). Depending on the size of the LBA and associated metadata, there may be performance implications. The performance analysis is based on better performance on a queue depth 32 with 4 KiB read workload.	This field indicates the relative performance of the LBA format indicated relative to other LBA formats supported by the controller (see value/definition table below). Depending on the size of the LBA and associated metadata, there may be performance implications. The performance analysis is based on better performance on a queue depth 32 with 4 KiB read workload.Value Definition00b Best performance01b Better performance10b Good performance11b Degraded performance
Manda tory	Mandatory	Yes
Notes		Only Mandatory / applicable for the NVM Command Set and I/O Command Sets based on the NVM Command Set. Prohibited for all other I/O Command Sets.

6.5.2.30 NVMeNamespaceProperties.LBAFormat.LBADataSizeBytes The map-
ping for `NVMeNamespaceProperties.LBAFormat.LBADataSizeBytes` is
summarized in Table 180.

Table 180: 178: NVMeNamespaceProperties.LBAFormat.LBADataSizeBytes mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Prope rty	NVMeNamespaceProperties.L BAFormat.LBADatasizeBytes	**NVM Spec Property / Field:** Retrieved in the Identify Namespace data structure for the specified NSID or the common namespace capabilities for the NVM Command Set (CNS 00h). This property may be found by retrieving the LBA Format data structure which starts at the LBAFormat offset for which this Namespace was formatted.Refer to the applicable NVMe I/O Command Set specification for details. For NVMe I/O Command Sets that don't define this field, it is considered reserved**NVM Spec: Section:Figure** NVMe NVM Command Set Specification 1.0b: Section 4.1.5.1: Figure 98 (LBA Format Data Structure, NVM Command Set Specific): Bits 16:23 - LBA Data Size (LBADS).
Type	8 bits	8 bits
Descr iptio n	The LBA data size supported.	The LBA data size supported, reported in terms of a power of two (2^n).

	Redfish/Swordfish	NVMe / NVMe-oF
LongD escri ption	This field indicates the LBA data size supported.	This field indicates the LBA data size supported. The value is reported in terms of a power of two (2^n). A value smaller than 9 (i.e., 512 bytes) is not supported. If the value reported is 0h, then the LBA format is not supported / used or is not currently available.
Manda tory	Mandatory	Yes
Notes		Only Mandatory / applicable for the NVM Command Set and I/O Command Sets based on the NVM Command Set. Prohibited for all other I/O Command Sets.

6.5.2.31 NVMeNamespaceProperties.LBAFormat.LBAMetaDataSizeBytes   The mapping for `NVMeNamespaceProperties.LBAFormat.LBAMetaDataSizeBytes` is summarized in Table 181.

Table 181: 179: NVMeNamespaceProperties.LBAFormat.LBAMetaDataSizeBytes mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Prope rty	NVMeNamespaceProperties.LBA Format.LBAMetaDatasizeBytes	**NVM Spec Property / Field:** Retrieved in the Identify Namespace data structure for the specified NSID or the common namespace capabilities for the NVM Command Set (CNS 00h). This property may be found by retrieving the LBA Format data structure which starts at the LBAFormat offset for which this Namespace was formatted.Refer to the applicable NVMe I/O Command Set specification for details. For NVMe I/O Command Sets that don't define this field, it is considered reserved**NVM Spec: Section:Figure** NVMe NVM Command Set Specification 1.0b: Section 4.1.5.1: Figure 98 (LBA Format Data Structure, NVM Command Set Specific): Bits 00:15 - MetaData Size (MS).
Type	16 bits	16 bits
Descr iptio n	The LBA metadata size supported.	The number of metadata bytes provided per LBA based on the LBA Data Size indicated

	Redfish/Swordfish	NVMe / NVMe-oF
LongD escri ption	This field indicates the LBA metadata size supported.	This field indicates the number of metadata bytes provided per LBA based on the LBA Data Size indicated. If there is no metadata supported, then this field shall be cleared to 0h.If metadata is supported, then the namespace may support the metadata being transferred as part of an extended data LBA or as part of a separate contiguous buffer.If end-to-end data protection is enabled, then the first eight bytes or last eight bytes of the metadata is the protection information (refer to the DPS field in the Identify Namespace data structure).
Manda tory	Mandatory	Yes
Notes		Only Mandatory / applicable for the NVM Command Set and I/O Command Sets based on the NVM Command Set. Prohibited for all other I/O Command Sets.

6.5.2.32

6.5.2.33 **NVMeNamespaceProperties.MetadataTransferredAtEndOfDataLBA**   The
mapping for `NVMeNamespaceProperties.MetadataTransferredAtEndOfDataLBA`
is summarized in Table 182.

Table 182: NVMeNamespaceProperties.MetadataTransferredAtEndOfDataLBA mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeNamespaceProperties.Met adataTransferredAtEndOfData LBA	**NVM Spec Property / Field:** Metadata Capabilities (MC) retrieved in some Identify Namespace data structures for the specified NSID or the common namespace capabilities for the NVM Command Set (CNS 00h).Refer to the applicable NVMe I/O Command Set specification for details. For NVMe I/O Command Sets that don't define this field, it is considered reserved**NVM Spec: Section:Figure** NVMe NVM Command Set Specification 1.0b: Section 4.1.5.1: Figure 97: Bytes 27 - Metadata Capabilities (MC).
Type	Boolean	2 Bits
Descript ion	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
LongDescription	This property shall indicate whether or not the metadata is transferred at the end of the LBA creating an extended data LBA.	Bit 1 if set to "1" indicates the namespace supports the metadata being transferred as part of a separate buffer that is specified in the Metadata Pointer. Bit 1 if cleared to "0" indicates that the namespace does not support the metadata being transferred as part of a separate buffer. Bit 0 if set to "1" indicates that the namespace supports the metadata being transferred as part of an extended data LBA. Bit 0 if cleared to "0" indicates that the namespace does not support the metadata being transferred as part of an extended data LBA.
Mandatory	Mandatory	Yes
Notes		Only Mandatory / applicable for the NVM Command Set and I/O Command Sets based on the NVM Command Set. Prohibited for all other I/O Command Sets.

6.5.2.34 NVMeNamespaceProperties.NVMeVersion   The mapping for NVMeNamespaceProperties.NVMeVersion is summarized in Table 183.

Table 183: NVMeNamespaceProperties.NVMeVersion mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeNamespaceProperties.NVMeVersion	**NVM Spec Property / Field:** Property Get Command / Version (VS) - Offset 8h **NVM Spec: Section:Figure** NVMe Base Specification 2.0: Section 3.1.3.2
Type	String	Int64
Description	The version of the NVMe Base Specification supported.	This property indicates the major, minor, and tertiary version of the NVM Express base specification that the controller implementation supports
LongDescription		This property 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.
Mandatory	Mandatory	Yes
Notes		

6.5.2.35 OptimumIOSizeBytes The mapping for `OptimumIOSizeBytes` is summarized in Table 184.

Table 184: OptimumIOSizeBytes mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	OptimumIOSizeBytes	**NVM Spec Property / Field:** Namespace Optimal Write Size (NOWS) retrieved in some Identify Namespace data structures for the specified NSID or the common namespace capabilities for the NVM Command Set (CNS 00h).Refer to the applicable NVMe I/O Command Set specification for details. For NVMe I/O Command Sets that don't define this field, it is considered reserved**NVM Spec: Section:Figure** NVMe NVM Command Set Specification 1.0b: Section 4.1.5.1: Figure 97: Bytes 72 - 73 - Namespace Optimal Write Size (NOWS)
Type	Int64	2 Bytes
Descript ion	The size in bytes of this Volume's optimum IO size.	This field indicates the size in logical blocks for optimal write performance for this namespace.

	Redfish/Swordfish	NVMe / NVMe-oF
LongDesc ription	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. If the OPTPERF bit is cleared to "0", then this field is reserved. The size indicated should be less than or equal to Maximum Data Transfer Size (MDTS) that is specified in units of minimum memory page size. The value of this field may change if the namespace is reformatted. The value of this field should be a multiple of Namespace Preferred Write Granularity (NPWG).
Mandator y	Optional Recommended for implementations with Endurance Groups and NVM Sets.	Optional

Redfish/Swordfish	NVMe / NVMe-oF
Notes	Only Mandatory / applicable for the NVM Command Set and I/O Command Sets based on the NVM Command Set. Prohibited for all other I/O Command Sets. Convert from blocks to bytes.If the namespace is associated with an NVM set, NOWS defined for this namespace shall be set to the Optimal Write Size field setting defined in NVM Set Attributes Entry (refer to the Namespace Identification Descriptor in the NVMe Base Specification) for the NVM Set with which this namespace is associated. If NOWS is not supported, the Optimal Write Size field in NVM Sets Attributes Entry (refer to the Namespace Identification Descriptor in the NVMe Base Specification) for the NVM Set with which this namespace is associated should be used by the host for I/O optimization. Refer to the NVMe NVM Command Set Specification, section 5.8.2 for how this field is utilized to improve performance and endurance

6.5.2.36 ProvisioningPolicy   The mapping for `ProvisioningPolicy` is summarized in Table 185.

Table 185: OptimumIOSizeBytes mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	ProvisioningPolicy	**NVM Spec Property / Field:** Namespace Features (NSFEAT) retrieved in some Identify Namespace data structures for the specified NSID or the common namespace capabilities for the NVM Command Set (CNS 00h).Refer to the applicable NVMe I/O Command Set specification for details. For NVMe I/O Command Sets that don't define this field, it is considered reserved**NVM Spec: Section:Figure** NVMe NVM Command Set Specification 1.0b: Section 4.1.5.1: Figure 97: Bytes 24 - Namespace Features (NSFEAT), Bit 0 (ThinP).
Type	enum (DataStorageLoSCapabilities .ProvisioningPolicy)	Single Bit (BOOL)
Descript ion	This property specifies the volume's storage allocation, or provisioning policy.	Indicates that the namespace supports thin provisioning

	Redfish/Swordfish	NVMe / NVMe-oF
LongDescription	This property shall specify the volume's supported storage allocation policy.	if set to "1" indicates that the namespace supports thin provisioning. If cleared to "0" indicates that thin provisioning is not supported
Mandatory	Optional Recommended for implementations that support thin provisioning.	Mandatory
Notes	possible values: Fixed / Thin	Only Mandatory / applicable for the NVM Command Set and I/O Command Sets based on the NVM Command Set. Prohibited for all other I/O Command Sets.

6.5.2.37 Status.State   The mapping for `Status.state` is summarized in Table 186.

Table 186: Status.State mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Status.State	**NVM Spec Property / Field:** Identify Command / I/O Command Set Independent Identify Namespace data structure (CNS 08h) / Namespace Status(NSTAT) **NVM Spec: Section:Figure** NVMe 2.0: Section 5.17.2.8: Figure 280 - Byte14: Namespace Status (NSTAT): - Bit 0: Namespace Ready (NRDY)
Type	Resource.State (enum)	Boolean

	Redfish/Swordfish	NVMe / NVMe-oF
Descript ion	The known state of the resource, such as, enabled.	Indicates if the Namespace is "ready"
LongDesc ription	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.	This field indicates the status of the namespace with the specified NSID. A value of "1" indicates that the namespace is ready. A value of "0" indicates that the namespace is not ready.
Mandator y	Mandatory	Yes
Notes	Possible values: Enabled / Disabled / StandbyOffline / StandbySare / InTest / Starting / Absent / UnavailableOffline / Deferring / Quiesced / Updating / Qualified	Namespace Ready (NRDY value = 1) maps to "Enabled". Namespace Not Ready (NRDY value = 0) maps to "Disabled".

6.5.2.38 Status.Health   The mapping for `Status.Health` is summarized in Table 187.

Table 187: Status.Health mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Status.Health	Property Get / Controller Status (Offset 18h) / Controller Fatal Status (CSTS.CFS) - Bit 01 **NVM Spec: Section:Figure** NVMe 2.0: 3.1.3.6: Figure 47
Type	Resource.Health	Bit
Descript ion	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.
Mandatory	Mandatory	Yes
Notes	Possible Values: OK / Warning / Critical	Reference Figure 222 (NVM Subsystem Hardware Error Event Codes) bit 09h of the NVMe Base Specification.

6.5.2.39 Status.HealthRollup The mapping for `Status.HealthRollup` is summarized in Table 188.

Table 188: Status.HealthRollup mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Status.HealthRollup	N/A
Type	Resource.Health	N/A
Descript ion	The overall health state from the view of this resource.	N/A
LongDesc ription	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
Mandator y	Optional	
Notes	Do not use for Namespace. There are no dependent resources.	Do not implement.

6.5.2.40 StorageGroups   The mapping for `StorageGroups` is summarized in Table 189.

Table 189: StorageGroups mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	StorageGroups	N/A
Type	StorageGroupCollection.Stor ageGroupCollection	N/A
Descript ion	An array of references to Storage Groups that includes this volume.	N/A
LongDesc ription	The value of this property shall contain references to all storage groups that include this volume.	N/A
Mandator y	Do Not Implement	
Notes		

6.5.2.41 WriteCachePolicy   The mapping for `WriteCachePolicy` is summarized in Table 190.

Table 190: WriteCachePolicy mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	WriteCachePolicy	N/A
Type	StorageGroupCollection.Stor ageGroupCollection	N/A
Descript ion	Indicates the write cache policy setting for the Volume	N/A
LongDesc ription	This property shall contain a boolean indicator of the write cache policy for the Volume.	N/A
Mandator y	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.

```
{{
    "@odata.type": "#StoragePool.v1_7_1.StoragePool",
    "Id": "1",
    "Name": "Endurance Group 1",
    "Description": "Single Endurance Group",
    "NVMeProperties": {
        "NVMePoolType": "EnduranceGroup"
    },
    "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.type": "#Capacity.v1_2_0.CapacitySource",
            "Name": "Source1",
            "Id": "Source1",
            "ProvidedCapacity": {
                "Data": {
                    "ConsumedBytes": 10995116277760,
                    "AllocatedBytes": 10995116277760
                }
            },
            "ProvidingDrives": {
                "@odata.id": "/red-
                ↪   fish/v1/Storage/FabricAttachArray/StoragePools/
                ↪   Endurance-
                ↪   Group1/CapacitySources/Source1/ProvidingDrives"
            },
            "ProvidingPools": {
                "@odata.id": "/red-
                ↪   fish/v1/Storage/FabricAttachArray/StoragePools/
                ↪   Endurance-
                ↪   Group1/CapacitySources/Source1/ProvidingPools"
            },
            "@odata.id": "/red-
            ↪   fish/v1/Storage/FabricAttachArray/StoragePools/
            ↪   EnduranceGroup1/CapacitySources/Source1"
        }
    ],
    "@odata.id": "/red-
    ↪   fish/v1/Storage/FabricAttachArray/StoragePools/EnduranceGroup1",
    "@Redfish.Copyright": "Copyright 2015-2024 SNIA. All
    ↪   rights reserved."
}
```

6.6.2 Property Mapping

6.6.2.1 AllocatedPools   The mapping for `AllocatedPools` is summarized in Table 191.

Table 191: AllocatedPools mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	AllocatedPools	**NVM Spec: Property / Field:** NVMe 2.0: Identify Command / NVM Set List (CNS 04h) **NVM Spec: Section:Figure** NVMe 2.0: 5.17.2.4: Figure 278
Type	StoragePoolCollection.StoragePoolCollection	NVM Set List
Descript ion	A reference to the collection of storage pools allocated from this storage pool.	A list of NVM sets
LongDesc ription	The value of this property shall contain a reference to the collection of storage pools allocated from this storage pool.	The data structure is an ordered list of up to 31 NVM Set Attribute Entry data structures, sorted 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 and are accessible by the controller processing the command
Mandator y	Mandatory	I/O Controller: Optional Admin Controller: Optional Discovery Controller: Prohibited

	Redfish/Swordfish	NVMe / NVMe-oF
Notes	Contains a pointer to the NVM Set allocated from this Endurance Group.	

6.6.2.2 Capacity.Data.AllocatedBytes The mapping for `Capacity.Data.AllocatedBytes` is summarized in Table 192.

Table 192: Capacity.Data.AllocatedBytes mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Capacity.Data.AllocatedByte s	**NVM Spec: Property / Field:** NVMe 2.0: Endurance Group Information Log (Log Identifier 09h) Total Endurance Group Capacity (TEGCAP) **NVM Spec: Section:Figure** NVMe 2.0: 5.116.1.10: Figure 218 Get Log Page – Endurance Group Information Log (Log Identifier 09h) : Total Endurance Group Capacity (TEGCAP) (bytes 175:160)
Type	Int64	16-Bytes
Descript ion	The number of bytes currently allocated by the storage system in this data store for this data type.	This field indicates the total NVM capacity in this Endurance Group. The value is in bytes.
LongDesc ription	The value shall be the number of bytes currently allocated by the storage system in this data store for this data type.	This field indicates the total NVM capacity in this Endurance Group. The value is in bytes. If this field is cleared to 0h, the NVM subsystem does not report the total NVM capacity in this Endurance Group.
Mandator y	Optional Mandatory for NVMe Drives.	I/O Controller: Optional Admin Controller: Optional Discovery Controller: Prohibited

Redfish/Swordfish	NVMe / NVMe-oF
Notes	

6.6.2.3 Capacity.Data.ConsumedBytes   The mapping for `Capacity.Data.ConsumedBytes` is summarized in Table 193.

Table 193: Capacity.Data.ConsumedBytes mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Capacity.Data.ConsumedBytes	**NVM Spec: Property / Field:** NVMe 2.0: Endurance Group Information Log (Log Identifier 09h) Total Endurance Group Capacity (TEGCAP) - Unallocated Endurance Group Capacity (UEGCAP) **NVM Spec: Section:Figure** NVMe 2.0: 5.116.1.10: Figure 218 Get Log Page – Endurance Group Information Log (Log Identifier 09h) Subtract Unallocated Endurance Group Capacity (UEGCAP) (bytes 191:176) from Total Endurance Group Capacity (TEGCAP) (bytes 175:160)
Type	Int64	16-Bytes
Descript ion	The number of bytes consumed in this data store for this data type.	Total Endurance Group Capacity (TEGCAP) - Unallocated Endurance Group Capacity (UEGCAP)
LongDesc ription	The value shall be the number of logical bytes currently consumed in this data store for this data type.	Total Endurance Group Capacity (TEGCAP) - Unallocated Endurance Group Capacity (UEGCAP)
Mandator y	Optional Mandatory for NVMe Drives.	I/O Controller: Optional Admin Controller: Optional Discovery Controller: Prohibited

	Redfish/Swordfish	NVMe / NVMe-oF
Notes		Total Endurance Group Capacity (TEGCAP): This field indicates the total NVM capacity in this Endurance Group. The value is in bytes. If this field is cleared to 0h, the NVM subsystem does not report the total NVM capacity in this Endurance Group. Unallocated Endurance Group Capacity (UEGCAP): This field indicates the unallocated NVM capacity in this Endurance Group. The value is in bytes. If this field is cleared to 0h, the NVM subsystem does not report the unallocated NVM capacity in this Endurance Group.

6.6.2.4 CapacitySources   The mapping for `CapacitySources` is summarized in Table 194.

Table 194: CapacitySources mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	CapacitySources	See notes
Type	Collection(Capacity.CapacitySource)	See Notes
Descript ion	An array of space allocations to this volume.	A list of NVM Endurance Groups accessible by the controller and the Endurance Group Information Log for each Endurance Group.
LongDesc ription	Fully or partially consumed storage from a source resource. Each entry provides capacity allocation information from a named source resource.	See notes.
Mandator y	Required for NVMe Drives.	Optional

	Redfish/Swordfish	NVMe / NVMe-oF
Notes	Contains the information about the providing capacity (e.g, NVMSet) for this endurance group. Each entry in the collection is mapped such that the CapacitySource instance contains the overall capacity (in bytes) and the types and pointers to the underlying capacity sources.	For each entry in the CapacitySource collection, map the information from the NVMe specification: **NVM Spec: Property / Field:**NVMe 2.0: Identify Command / NVM Endurance Group List (CNS 19h) and NVMe 2.0: Endurance Group Information log (Log ID 09h)**NVM Spec: Section:Figure**NVMe 2.0: 5.17.2.18: Figure 288 and NVMe 2.0: 5.16.1.10: Figure 217. Identify command to access the Endurance Group List data structure (refer to section 5.17.2.18) to determine the Endurance Groups that are accessible by the controller. To determine the capacity information for each Endurance Group, the host uses the Get Log Page command to access the Endurance Group Information log page (refer to section 5.16.1.10).

6.6.2.5 CapacitySources@odata.count   The mapping for `CapacitySources@odata.count` is summarized in Table 195.

Table 195: CapacitySources@odata.count mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	CapacitySources@odata.coun t	**NVM Spec: Property / Field:** NVMe 2.0: Identify Command / NVM Endurance Group List (CNS 19h) **NVM Spec: Section:Figure** NVMe 2.0: 5.17.2.4: Figure 278. Bytes 00 (Number of Identifiers) in the NVM Endurance Group List
Type	(odata property)int64	Int64
Descript ion	Count of the number of items in the CapacitySources array.	Number of Identifiers in the NVM Endurance Group List
LongDesc ription		This field contains the number of Endurance Group Identifiers in the list. There may be up to 2,047 identifiers in the list. If this field is cleared to 0h, then no Endurance Group Identifiers are in the list.
Mandator y	Mandatory	Optional
Notes		

6.6.2.6 Description   The mapping for `Description` is summarized in Table 196.

Table 196: Description mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Description	N/A
Type	String	N/A
Descript ion	The description of this resource.	N/A
LongDesc ription	This object represents the description of this resource. The resource values shall comply with the Redfish Specification-described requirements.	N/A
Mandator y	Optional	
Notes	In Redfish, Description is a read-only field.	

6.6.2.7 Links.OwningStorageResource   The mapping for `Links.OwningStorageResource` is summarized in Table 197.

Table 197: Links.OwningStorageResource mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Links.OwningStorageResource	N/A
Type	Storage.Storage	N/A
Descript ion	A pointer to the Storage resource that owns or contains this StoragePool.	N/A
LongDesc ription	This shall be a pointer to the Storage resource that owns or contains this StoragePool.	N/A
Mandator y	Mandatory	
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 198.

Table 198: Name mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Name	**NVM Spec: Property / Field:** NVMe 2.0: Identify Command / CNS value of 19h (Endurance Group List) **NVM Spec: Section:Figure** NVMe 2.0: 5.17.2.18 Identify Controller data structure / CNS value of 19h to retrieve the Endurance Group List: Figure 288. This list will contain a Number of Endurance Group Identifiers equal to the Number of Identifiers (N) field in Bytes 01:00.
Type	String	2-Bytes
Descript ion	The name of the resource or array member.	The list contains Endurance Group Identifiers of Endurance Groups that are accessible by the controller processing the command.
LongDesc ription	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 Endurance Group List is a list of up to 2,047 Endurance Group Identifiers in increasing order that are accessible by the controller processing the command.

	Redfish/Swordfish	NVMe / NVMe-oF
Mandator y	Mandatory	Optional (Mandatory for controllers that support Variable Capacity Management)
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   `NVMeProper-`
`ties.NVMePoolType` is summarized in Table 199.

Table 199: NVMeProperties.NVMePoolType

	Redfish/Swordfish	NVMe / NVMe-oF
Property	StoragePool.NVMePoolType	N/A
Type	Enum	N/A
Descript ion	Indicates whether the StoragePool is used as an EnduranceGroup or an NVMSet.	N/A
LongDesc ription	This property shall indicate whether the StoragePool is used as an EnduranceGroup or an NVMSet.	N/A
Mandator y	Mandatory	
Notes	Set as "EnduranceGroup"	

6.6.2.10 NVMeEnduranceGroupProperties.PredictedMediaLifeLeftPercent   The mapping for `NVMeEnduranceGroupProperties.PredictedMediaLifeLeftPercent` is summarized in Table 200.

Table 200: NVMeEnduranceGroupProperties.PredictedMediaLifeLeftPercent mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeEnduranceGroupPropertie s.PredictedMediaLifeLeftPer cent	**NVM Spec: Property / Field:** NVMe 2.0: Endurance Group Information Log (Log Identifier 09h) Inverse of the "Percentage Used" field (Byte 05) **NVM Spec: Section:Figure** NVMe 2.0: 5.16.1.10: Figure 217
Type	Decimal %	Decimal %
Descript ion	The percentage of reads and writes that are predicted to be available for the media.	The Inverse of the "Percentage Used" field which contains a vendor specific estimate of the percentage of life used for the Endurance Group based on the actual usage and the manufacturer's prediction of NVM life.

	Redfish/Swordfish	NVMe / NVMe-oF
LongDesc ription	This property shall contain an indicator of the percentage of life remaining in the drive's media.	The Inverse of the "Percentage Used" field which contains a vendor specific estimate of the percentage of life used for the Endurance Group based on the actual usage and the manufacturer's 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.
Mandator y	Mandatory	I/O Controller: Optional Admin Controller: Optional Discovery Controller: Prohibited
Notes		The value is allowed to exceed 100. Percentages greater than 254 shall be represented as 255. This value shall be updated once per power-on hour when the controller is not in a sleep state.

6.6.2.11 NVMeEnduranceGroupProperties.EndGrpLifetime.PercentUsed   The mapping for `NVMeEnduranceGroupProperties.EndGrpLifetime.PercentUsed` is summarized in Table 201.

Table 201: NVMeEnduranceGroupProperties.EndGrpLifetime.PercentUsed mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeEnduranceGroupProperties.EndGrpLifetime.PercentUse d	**NVM Spec: Property / Field:** NVMe 2.0: Endurance Group Information Log (Log Identifier 09h) "Percentage Used" field (Byte 05) **NVM Spec: Section:Figure** NVMe 2.0: 5.16.1.10: Figure 217
Type	Int64	Decimal % from 00% - 255%
Descript ion	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.	Contains a vendor specific estimate of the percentage of life used for the Endurance Group based on the actual usage and the manufacturer's prediction of NVM life.

	Redfish/Swordfish	NVMe / NVMe-oF
LongDesc ription	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.	Contains a vendor specific estimate of the percentage of life used for the Endurance Group based on the actual usage and the manufacturer's 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. The value is allowed to exceed 100. Percentages greater than 254 shall be represented as 255. This value shall be updated once per power-on hour when the controller is not in a sleep state.
Mandator y	Mandatory	I/O Controller: Optional Admin Controller: Optional Discovery Controller: Prohibited
Notes		The value is allowed to exceed 100. Percentages greater than 254 shall be represented as 255. This value shall be updated once per power-on hour when the controller is not in a sleep state. This value shall be scaled to a value between 0% - 100% Calculate resulting value as 100 - value reported (PercentageUsed).

6.6.2.12 NVMeEnduranceGroupProperties.EndGrpLifetime.EnduranceEstimate

The mapping for NVMeEnduranceGroupProperties.EndGrpLifetime.EnduranceEstimate is summarized in Table 202.

Table 202: NVMeEnduranceGroupProperties.EndGrpLifetime.EnduranceEstimate mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeEnduranceGroupProperties.EndGrpLifetime.EnduranceEstimate	**NVM Spec: Property / Field:** NVMe 2.0: Endurance Group Information Log (Log Identifier 09h) "Endurance Estimate" field (Bytes 32-47) **NVM Spec: Section:Figure** NVMe 2.0: 5.16.1.10: Figure 217
Type	Int64	16-Bytes
Descript ion	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.	This field is an estimate of the total number of data bytes that may be written to the Endurance Group over the lifetime of the Endurance Group.

	Redfish/Swordfish	NVMe / NVMe-oF
LongDesc ription	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.	This field is 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 amplification of 1 (i.e., no increase in the number of write operations performed by the device beyond the number of write operations requested by a host).
Mandator y	Mandatory	I/O Controller: Optional Admin Controller: Optional Discovery Controller: Prohibited
Notes		This value is reported in billions (i.e., a value of 1 corresponds to 1,000,000,000 bytes written) and is rounded up (e.g., one indicates the number of bytes written is from 1 to 1,000,000,000, three indicates the number of bytes written is from 2,000,000,001 to 3,000,000,000). A value of 0h indicates that the controller does not report an Endurance Estimate. This value shall be scaled to a value between 0% - 100% Calculate resulting value as 100 - value reported (PercentageUsed).

6.6.2.13 **NVMeEnduranceGroupProperties.EndGrpLifetime.DataUnitsRead**   The

mapping for `NVMeEnduranceGroupProperties.EndGrpLifetime.DataUnitsRead` is summarized in Table 203.

Table 203: NVMeEnduranceGroupProperties.EndGrpLifetime.DataUnitsRead mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeEnduranceGroupProperties.EndGrpLifetime.DataUnitsRead	**NVM Spec: Property / Field:** NVMe 2.0: Endurance Group Information Log (Log Identifier 09h) "Data Units Read" field (Bytes 48 - 63) **NVM Spec: Section:Figure** NVMe 2.0: 5.16.1.10: Figure 217
	Int64	16-Bytes
Description	The property contains the total number of data units read from this endurance group.	Contains the total number of data bytes that have been read from the Endurance Group.
LongDescription	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.	Contains the total number of data bytes that have been read from the Endurance Group. This value does not include controller reads due to internal operations such as garbage collection.
Mandatory	Mandatory	I/O Controller: Optional Admin Controller: Optional Discovery Controller: Prohibited

Redfish/Swordfish	NVMe / NVMe-oF
Notes	This value is reported in billions (i.e., a value of 1 corresponds to 1,000,000,000 bytes read) and is rounded up (e.g., one indicates the number of bytes read is from 1 to 1,000,000,000, three indicates the number of bytes read is from 2,000,000,001 to 3,000,000,000). A value of 0h indicates that the controller does not report the number of Data Units Read

6.6.2.14 NVMeEnduranceGroupProperties.EndGrpLifetime.DataUnitsWritten

The mapping for NVMeEnduranceGroupProperties.EndGrpLifetime.DataUnitsWritten is summarized in Table 204.

Table 204: NVMeEnduranceGroupProperties.EndGrpLifetime.DataUnitsWritten mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeEnduranceGroupPropertie s.EndGrpLifetime.DataUnitsW ritten	**NVM Spec: Property / Field:** NVMe 2.0: Endurance Group Information Log (Log Identifier 09h) "Data Units Written" field (Bytes 64 - 79) **NVM Spec: Section:Figure** NVMe 2.0: 5.16.1.10: Figure 217
Type	Int64	16-Bytes
Descript ion	The property contains the total number of data units written from this endurance group.	Contains the total number of data bytes that have been written to the Endurance Group.
LongDesc ription	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.	Contains the total number of data bytes that have been written to the Endurance Group. This value does not include controller writes due to internal operations such as garbage collection.
Mandator y	Mandatory	I/O Controller: Optional Admin Controller: Optional Discovery Controller: Prohibited

Redfish/Swordfish	NVMe / NVMe-oF
Notes	This value is reported in billions (i.e., a value of 1 corresponds to 1,000,000,000 bytes written) and is rounded up (e.g., one indicates the number of bytes written is from 1 to 1,000,000,000, three indicates the number of bytes written is from 2,000,000,001 to 3,000,000,000). A value of 0h indicates that the controller does not report the number of Data Units Written.

6.6.2.15 **NVMeEnduranceGroupProperties.EndGrpLifetime.MediaUnitsWritten**

The mapping for NVMeEnduranceGroupProperties.EndGrpLifetime.MediaUnitsWritten is summarized in Table 205.

Table 205: NVMeEnduranceGroupProperties.EndGrpLifetime.MediaUnitsWritten mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeEnduranceGroupPropertie s.EndGrpLifetime.MediaUnits Written	**NVM Spec: Property / Field:** NVMe 2.0: Endurance Group Information Log (Log Identifier 09h) "Media Units Written" field (Bytes 80 - 95) **NVM Spec: Section:Figure** NVMe 2.0: 5.16.1.10: Figure 217
Type	Int64	16-Bytes
Descript ion	The property contains the total number of data units written from this endurance group.	Contains the total number of data bytes that have been written to the Endurance Group including both host and controller writes (e.g., garbage collection).
LongDesc ription	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.	Contains the total number of data bytes that have been written to the Endurance Group including both host and controller writes (e.g., garbage collection).

	Redfish/Swordfish	NVMe / NVMe-oF
Mandator y	Mandatory	I/O Controller: Optional Admin Controller: Optional Discovery Controller: Prohibited
Notes		This value is reported in billions (i.e., a value of 1 corresponds to 1,000,000,000 bytes written) and is rounded up (e.g., one indicates the number of bytes written is from 1 to 1,000,000,000, three indicates the number of bytes written is from 2,000,000,001 to 3,000,000,000). A value of 0h indicates that controller does not report the number of Media Units Written.

6.6.2.16 NVMeEnduranceGroupProperties.EndGrpLifetime.HostReadCommandCount

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

Table 206:

NVMeEnduranceGroupProperties.EndGrpLifetime.HostReadCommandCount mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeEnduranceGroupPropertie s.EndGrpLifetime.HostReadCo mmandCount	**NVM Spec: Property / Field:** NVMe 2.0: Endurance Group Information Log (Log Identifier 09h) "Host Read Commands" field (Bytes 96 - 111) **NVM Spec: Section:Figure** NVMe 2.0: 5.16.1.10: Figure 217
Type	Int64	16-Bytes
Descript ion	This property contains the number of read commands completed by all controllers in the NVM subsystem for the Endurance Group.	Contains the number of User Data Read Access Commands completed by the controller
LongDesc ription	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.	Contains the number of User Data Read Access Commands completed by the controller
Mandator y	Mandatory	I/O Controller: Optional Admin Controller: Optional Discovery Controller: Prohibited

	Redfish/Swordfish	NVMe / NVMe-oF
Notes		Refer to the specific NVMe I/O Command Set specification for the list of User Data Read Access Commands that affect this field.

6.6.2.17 NVMeEnduranceGroupProperties.EndGrpLifetime.HostWriteCommandCount

The mapping for `NVMeEnduranceGroupProperties.EndGrpLifetime.HostWriteCommandCount` is summarized in Table 6.6.2.17.

Table 207:

NVMeEnduranceGroupProperties.EndGrpLifetime.HostWriteCommandCount mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeEnduranceGroupProperties.EndGrpLifetime.HostWriteCommandCount	**NVM Spec: Property / Field:** NVMe 2.0: Endurance Group Information Log (Log Identifier 09h) "Host Write Commands" field (Bytes 112 - 127) **NVM Spec: Section:Figure** NVMe 2.0: 5.16.1.10: Figure 217
Type	Int64	16-Bytes
Descript ion	This property contains the number of write commands completed by all controllers in the NVM subsystem for the Endurance Group.	Contains the number of User Data Out Commands completed by the controller.
LongDesc ription	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.	Contains the number of User Data Out Commands completed by the controller.
Mandator y	Mandatory	I/O Controller: Optional Admin Controller: Optional Discovery Controller: Prohibited

Redfish/Swordfish	NVMe / NVMe-oF
Notes	Refer to the specific I/O Command Set specification for the list of User Data Out Commands that affect this field.

Table: NVMeEnduranceGroupProperties.EndGrpLifetime.HostWriteCommandCount mapping

6.6.2.18 **NVMeEnduranceGroupProperties.EndGrpLifetime.MediaAndDataIntegrityErrorCount**

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

Table 208: NVMeEnduranceGroupProperties.EndGrpLifetime
MediaAndDataIntegrityErrorCount mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeEnduranceGroupProperties.EndGrpLifetime.MediaAndDataIntegrityErrorCount	**NVM Spec: Property / Field:** NVMe 2.0: Endurance Group Information Log (Log Identifier 09h) "Media and Data Integrity Errors" field (Bytes 128 - 143) **NVM Spec: Section:Figure** NVMe 2.0: 5.16.1.10: Figure 217
Type	Int64	16-Bytes
Descript ion	This property contains the number of occurences where the controller detected an unrecovered data integrity error for the Endurance Group.	Contains the number of occurrences where the controller detected an unrecovered data integrity error.
LongDesc ription	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.	Contains the number of occurrences where the controller detected an unrecovered data integrity error.
Mandator y	Mandatory	I/O Controller: Optional Admin Controller: Optional Discovery Controller: Prohibited

Redfish/Swordfish	NVMe / NVMe-oF
Notes	Errors such as uncorrectable ECC, CRC checksum failure, or LBA tag mismatch are included in this field. Errors introduced as a result of a Write Uncorrectable command (refer to the NVM Command Set specification) may or may not be included in this field.

6.6.2.19 NVMeEnduranceGroupProperties.EndGrpLifetime.ErrorInformationLogEntryCount

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

Table 209:

NVMeEnduranceGroupProperties.EndGrpLifetime.ErrorInformationLogEntryCount
mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeEnduranceGroupProperties.EndGrpLifetime.ErrorInformationLogEntryCount	**NVM Spec: Property / Field:** NVMe 2.0: Endurance Group Information Log (Log Identifier 09h) "Number of Error Information Log Entries" field (Bytes 144 - 159) **NVM Spec: Section:Figure** NVMe 2.0: 5.16.1.10: Figure 217
Type	Int64	16-Bytes
Description	This property contains the number of error information log entries over the life of the controller for the endurance group.	Contains the number of Error Information log entries over the life of the controller.
LongDescription	This property shall contain the number of error information log entries over the life of the controller for the endurance group.	Contains the number of Error Information log entries over the life of the controller.
Mandatory	Mandatory	I/O Controller: Optional Admin Controller: Optional Discovery Controller: Prohibited
Notes		

6.6.2.20 NVMeSetProperties.SetIdentifier The mapping for `NVMeSetProper-ties.SetIdentifier` is summarized in Table 210.

Table 210: NVMeSetProperties.SetIdentifier

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeSetProperties.SetIdenti fier	**NVM Spec: Property / Field:** NVMe 2.0: Identify Command / NVM Set List (CNS 04h) **NVM Spec: Section:Figure** NVMe 2.0: 5.17.2.4
Type	String	16-bits
Descript ion	A 16-bit hex value that contains the NVMe Set identifier.	This field indicates the identifier of the NVM Set in the NVM subsystem that is described by this entry.
LongDesc ription	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.	This field indicates the identifier of the NVM Set in the NVM subsystem that is described by this entry. Identify Command / CNS value of 04h to retrieve the NVM Set List. The NVM Set Identifier is retrieved in bytes 00:01 of each NVM Set Attributes entry in the NVM Set List
Mandator y Notes	Do Not Implement	Optional

6.6.2.21 NVMeSetProperties.OptimalWriteSizeBytes   The mapping for `NVMeSet-Properties.OptimalWriteSizeBytes` is summarized in Table 211.

Table 211: NVMeSetProperties.OptimalWriteSizeBytes mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeSetProperties.OptimalWriteSizeBytes	**NVM Spec: Property / Field:** NVMe 2.0: Identify Command / NVM Set List (CNS 04h) f/ Optimal Write Size **NVM Spec: Section:Figure** NVMe 2.0: 5.17.2.4 Identify Command / NVM Set List (CNS value of 04h to retrieve the NVM Set List): Figure 278. The Optimal Write Size is retrieved in bytes 12:15 of each NVM Set Attributes entry (Figure 279) in the NVM Set List
Type	Int64	Bytes
Descript ion	This property contains the Optimal Write Size in Bytes for this NVMe Set.	This field indicates the size in bytes for optimal write performance.
LongDesc ription	This property shall contain the Optimal Write Size in Bytes for this NVMe Set.	This field indicates the size in bytes for optimal write performance. A value of 0h indicates that no Optimal Write Size is specified. This field should be cleared to 0h when namespaces within an NVM Set have different User Data Formats that do not allow an Optimal Write Size to be specified.
Mandator y	Do Not Implement	Optional

Redfish/Swordfish	NVMe / NVMe-oF
Notes	

6.6.2.22 NVMeSetProperties.EnduranceGroupIdentifier   The mapping for NVMe-
`SetProperties.EnduranceGroupIdentifier` is summarized in Table 212.

Table 212: NVMeSetProperties.EnduranceGroupIdentifier mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeSetProperties.Endurance GroupIdentifier	N/A
Type	String	N/A
Descript ion	A 16-bit hex value that contains the endurance group identifier.	N/A
LongDesc ription	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.	N/A
Mandator y	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 213.

Table 213: NVMeSetProperties.Random4kReadTypicalNanoSeconds mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeSetProperties.Random4kReadTypicalNanoSeconds	N/A
Type	Int64	N/A
Descript ion	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.	N/A
LongDesc ription	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.	N/A
Mandator y	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 214.

Table 214: Status.Health mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Status.Health	**NVM Spec: Property / Field:** NVMe 2.0: Get Log Page – Endurance Group Information Log (Log Identifier 09h) / Critical Warning **NVM Spec: Section:Figure** NVMe 2.0: 5.16.1.10: Figure 217 / Critical Warning (bytes 00)
Type	Resource.Health	Bit value
Descript ion	The health state of this resource in the absence of its dependent resources.	This field indicates critical warnings for the state of the Endurance Group.
LongDesc ription	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.	This field indicates critical warnings for the state of the Endurance Group. Each bit corresponds to a critical warning type; multiple bits may be set to "1". If a bit is cleared to "0", then that critical warning does not apply. Critical warnings may result in an asynchronous event notification to the host. Bits in this field represent the current associated state and are not persistent.
Mandator y	Optional	I/O Controller: Optional Admin Controller: Optional Discovery Controller: Prohibited

	Redfish/Swordfish	NVMe / NVMe-oF
Notes	Possible Values: OK / Warning / Critical	If bit 3 is set to "1", then all namespaces in the Endurance Group have been placed in read only mode for reasons other than a change in the write protect state of the namespace. The controller shall not set this bit to "1" if the read-only condition on the Endurance Group is a result of a change in the write protection state of all namespaces in the Endurance Group. If bit 2 is set to "1", then the Endurance Group reliability has been degraded due to significant media related errors or any internal error that degrades NVM subsystem reliability. if bit 0 is set to "1", then the available spare capacity of the Endurance Group has fallen below the threshold.

6.6.2.25 Status.State   The mapping for `Status.State` is summarized in Table 215.

Table 215: Status.State mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Status.State	N/A
Type	Resource.State (enum)	N/A
Descript ion	The known state of the resource, such as, enabled.	N/A
LongDesc ription	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.	N/A
Mandator y	Do Not Implement	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.

```
{
  "@odata.type": "#StoragePool.v1_7_1.StoragePool",
  "Id": "1",
  "Name": "Set 1",
  "Description": "First Set",
  "NVMeProperties": {
    "NVMePoolType": "NVMSet"
  },
  "NVMeSetProperties": {
    "SetIdentifier": "0x1F",
    "EnduranceGroupIdentifier": "0x1",
    "Random4kReadTypicalNanoSeconds": 34534345348,
    "UnallocatedNVMNamespaceCapacityBytes": 5497558138880,
    "OptimalWriteSizeBytes": 512
  },
  "Capacity": {
    "Data": {
      "AllocatedBytes": 10995116277760,
      "ConsumedBytes": 5497558138880
    }
  },
  "CapacitySources": [{
    "@odata.type": "#Capacity.v1_2_0.CapacitySource",
    "Name": "Source1",
    "Id": "Source1",
    "Description": "Capacity Source for NVM Set",
    "ProvidedCapacity": {
```

```
      "Data": {
        "AllocatedBytes": 10737418240
      }
    },
    "ProvidingPools": {
      "@odata.id": "/red-
        ↪   fish/v1/Storage/FabricAttachArray/StoragePools/EnduranceGroup1/
        ↪   CapacitySources/Source1/ProvidingPools"
    },
    "@odata.id": "/red-
      ↪   fish/v1/Storage/FabricAttachArray/StoragePools/NVMeSet1/
      ↪   CapacitySources/Source1"
  }],
  "AllocatedVolumes": {
    "@odata.id": "/red-
      ↪   fish/v1/Storage/FabricAttachArray/StoragePools/NVMeSet1/AllocatedVolu
  },
  "@odata.id": "/red-
    ↪   fish/v1/Storage/FabricAttachArray/StoragePools/NVMeSet1",
  "@Redfish.Copyright": "Copyright 2015-2024 SNIA. All
    ↪   rights reserved."
}
```

6.7.2 Property Mapping

6.7.2.1 AllocatedVolumes   The mapping for `AllocatedVolumes` is summarized in Table 216.

Table 216: AllocatedVolumes mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	AllocatedVolumes	N/A
Type	VolumeCollection.VolumeCollection	N/A
Descript ion	A reference to the collection of volumes allocated from this storage pool.	N/A
LongDesc ription	The value of this property shall contain a reference to the collection of volumes allocated from this storage pool.	N/A
Mandator y	Mandatory	
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 `Capacity.Data.AllocatedBytes` is summarized in Table 217

Table 217: Capacity.Data.AllocatedBytes mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Capacity.Data.AllocatedByte s	**NVM Spec Property / Field:** Identify Command / NVM Set List (CNS 04h) / NVM Set Attributes Entry / Total NVM Set Capacity **NVM Spec: Section:Figure** NVMe 2.0: Section 5.15.2.4, Figure 278: NVM Set Attributes List / Figure 279: NVM Set Attributes Entry: Bytes 16-31 (Total NVM Set Capacity)
Type	Int64	16 Bytes
Descript ion	The number of bytes currently allocated by the storage system in this data store for this data type.	This field indicates the total NVM capacity in this NVM Set.
LongDesc ription	The value shall be the number of bytes currently allocated by the storage system in this data store for this data type.	This value shall be the total NVM capacity in this NVM Set. The value is in bytes.
Mandator y	Mandatory	Mandatory
Notes		

6.7.2.3 Capacity.Data.ConsumedBytes   The mapping for `Capacity.Data.ConsumedBytes` is summarized in Table 218.

Table 218: Capacity.Data.ConsumedBytes mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Capacity.Data.ConsumedBytes	**NVM Spec Property / Field:** Identify Command / NVM Set List (CNS 04h) / NVM Set Attributes Entry / Unallocated NVM Set Capacity **NVM Spec: Section:Figure** NVMe 2.0: Section 5.15.2.4, Figure 278: NVM Set Attributes List / Figure 279: NVM Set Attributes Entry: Bytes 32 - 47 (Unallocated NVM Set Capacity)
Type	Int64	16 Bytes
Descript ion	The maximum number of bytes that can be allocated in this data store for this data type.	This field indicates the unallocated NVM capacity in this NVM Set.
LongDesc ription	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 unallocated NVM capacity in this NVM Set. The value is in bytes.
Mandator y	Mandatory	Optional
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 [219].

Table 219: CapacitySources mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	CapacitySources	N/A
Type	Collection(Capacity.Capacit ySource)	N/A
Descript ion	An array of space allocations to this volume.	N/A
LongDesc ription	Fully or partially consumed storage from a source resource. Each entry provides capacity allocation information from a named source resource.	N/A
Mandator y	Optional Recommended to not implement for NVMe Drives.	DNI
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 220.

Table 220: CapacitySources@odata.count mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	CapacitySources@odata.count	N/A
Type	(odata property)int64	N/A
Description	Count of the number of items in the CapacitySources array.	N/A
LongDescription		N/A
Mandatory	Optional Do Not Implement for NVMe Drives.	DNI
Notes		

6.7.2.6 Description   The mapping for `Description` is summarized in Table 221.

Table 221: Description mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Description	N/A
Type	String	N/A
Descript ion	The description of this resource.	See note below.
LongDesc ription	This object represents the description of this resource. The resource values shall comply with the Redfish Specification-described requirements.	N/A
Mandator y	Optional	
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 222.

Table 222: Links.OwningStorageResource mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Links.OwningStorageResource	N/A
Type	Storage.Storage	N/A
Descript ion	A pointer to the Storage resource that owns or contains this StoragePool.	N/A
LongDesc ription	This shall be a pointer to the Storage resource that owns or contains this StoragePool.	N/A
Mandator y	Mandatory	
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 223

Table 223: Name mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Name	**NVM Spec Property / Field:** Identify Command / NVM Set List (CNS 04h) / NVM Set Attributes Entry / NVM Set Identifier **NVM Spec: Section:Figure** NVMe 2.0: Section 5.15.2.4, Figure 278: NVM Set Attributes List / Figure 279: NVM Set Attributes Entry: Bytes 00 - 01 (NVM Set Identifier)
Type	String	16-bit value
Descript ion	The name of the resource or array member.	This field specifies the NVM Set ID
LongDesc ription	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.	This field specifies the NVM Set ID. An NVM Set Identifier is a 16-bit value that specifies the NVM Set
Mandator y	Mandatory	
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 `NVMeProper-ties.NVMePoolType` is summarized in Table 224.

Table 224: NVMeProperties.NVMePoolType

	Redfish/Swordfish	NVMe / NVMe-oF
Property	StoragePool.NVMePoolType	N/A
Type	Enum	N/A
Descript ion	Indicates whether the StoragePool is used as an EnduranceGroup or an NVMSet.	N/A
LongDesc ription	This property shall indicate whether the StoragePool is used as an EnduranceGroup or an NVMSet.	N/A
Mandator y	Mandatory	
Notes	Set as "NVMSet"	

6.7.2.10 NVMeEnduranceGroupProperties.PredictedMediaLifeLeftPercent   The
mapping for `NVMeEnduranceGroupProperties.PredictedMediaLifeLeftPercent`
is summarized in Table 225.

Table 225: NVMeEnduranceGroupProperties.PredictedMediaLifeLeftPercent mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeEnduranceGroupPropertie s.PredictedMediaLifeLeftPer cent	N/A
Type	Decimal %	N/A
Descript ion	The percentage of reads and writes that are predicted to be available for the media.	N/A
LongDesc ription	This property shall contain an indicator of the percentage of life remaining in the drive's media.	N/A
Mandator y Notes	Do Not Implement	DNI

6.7.2.11 NVMeEnduranceGroupProperties.EndGrpLifetime.PercentUsed   The mapping for `NVMeEnduranceGroupProperties.EndGrpLifetime.PercentUsed` is summarized in Table 226.

Table 226: NVMeEnduranceGroupProperties.EndGrpLifetime.PercentUsed mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeEnduranceGroupPropertie s.EndGrpLifetime.PercentUse d	N/A
Type	Int64	N/A
Descript ion	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.	N/A
LongDesc ription	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.	N/A
Mandator y	Do Not Implement	DNI

Redfish/Swordfish	NVMe / NVMe-oF
Notes	

6.7.2.12 **NVMeEnduranceGroupProperties.EndGrpLifetime.EnduranceEstimate**

The mapping for NVMeEnduranceGroupProperties.EndGrpLifetime.EnduranceEstimate is summarized in Table 227.

Table 227: NVMeEnduranceGroupProperties.EndGrpLifetime.EnduranceEstimate mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeEnduranceGroupPropertie s.EndGrpLifetime.EnduranceE stimate	N/A
Type	Int64	N/A
Descript ion	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.	N/A
LongDesc ription	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.	N/A
Mandator y Notes	Do Not Implement	DNI

6.7.2.13 **NVMeEnduranceGroupProperties.EndGrpLifetime.DataUnitsRead** The
mapping for `NVMeEnduranceGroupProperties.EndGrpLifetime.DataUnitsRead`
is summarized in Table 228.

Table 228: NVMeEnduranceGroupProperties.EndGrpLifetime.DataUnitsRead mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeEnduranceGroupPropertie s.EndGrpLifetime.DataUnitsR ead	N/A
Type	Int64	N/A
Descript ion	The property contains the total number of data units read from this endurance group.	N/A
LongDesc ription	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.	N/A
Mandator y Notes	Do Not Implement	DNI

6.7.2.14 **NVMeEnduranceGroupProperties.EndGrpLifetime.DataUnitsWritten**   The mapping for `NVMeEnduranceGroupProperties.EndGrpLifetime.DataUnitsWritten` is summarized in Table 229.

Table 229: NVMeEnduranceGroupProperties.EndGrpLifetime.DataUnitsWritten mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeEnduranceGroupProperties.EndGrpLifetime.DataUnitsWritten	N/A
Type	Int64	N/A
Descript ion	The property contains the total number of data units written from this endurance group.	N/A
LongDesc ription	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.	N/A
Mandator y Notes	Do Not Implement	DNI

6.7.2.15 **NVMeEnduranceGroupProperties.EndGrpLifetime.MediaUnitsWritten**

The mapping for `NVMeEnduranceGroupProperties.EndGrpLifetime.MediaUnitsWritten` is summarized in Table 230.

Table 230: NVMeEnduranceGroupProperties.EndGrpLifetime.MediaUnitsWritten mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeEnduranceGroupPropertie s.EndGrpLifetime.MediaUnits Written	N/A
Type	Int64	N/A
Descript ion	The property contains the total number of data units written from this endurance group.	N/A
LongDesc ription	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.	N/A
Mandator y Notes	Do Not Implement	DNI

6.7.2.16 NVMeEnduranceGroupProperties.EndGrpLifetime.HostReadCommandCount

The mapping for NVMeEnduranceGroupProperties.EndGrpLifetime.HostReadCommandCount is summarized in Table 231.

Table 231:

NVMeEnduranceGroupProperties.EndGrpLifetime.HostReadCommandCount mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeEnduranceGroupPropertie s.EndGrpLifetime.HostReadCo mmandCount	N/A
Type	Int64	N/A
Descript ion	This property contains the number of read commands completed by all controllers in the NVM subsystem for the Endurance Group.	N/A
LongDesc ription	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.	N/A
Mandator y	Do Not Implement	DNI

6.7.2.17 **NVMeEnduranceGroupProperties.EndGrpLifetime.HostWriteCommandCount**

The mapping for NVMeEnduranceGroupProperties.EndGrpLifetime.HostWriteCommandCount is summarized in Table 232.

Table 232:

NVMeEnduranceGroupProperties.EndGrpLifetime.HostWriteCommandCount mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeEnduranceGroupPropertie s.EndGrpLifetime.HostWriteC ommandCount	N/A
Type	Int64	N/A
Descript ion	This property contains the number of write commands completed by all controllers in the NVM subsystem for the Endurance Group.	N/A
LongDesc ription	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.	N/A
Mandator y Notes	Do Not Implement	DNI

6.7.2.18 NVMeEnduranceGroupProperties.EndGrpLifetime.MediaAndDataIntegrityErrorCount

The mapping for NVMeEnduranceGroupProperties.EndGrpLifetime.MediaAndDataIntegrityE
is summarized in Table 233.

Table 233:

NVMeEnduranceGroupProperties.EndGrpLifetime.MediaAndDataIntegrityErrorCount
mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeEnduranceGroupPropertie s.EndGrpLifetime.MediaAndDa taIntegrityErrorCount	N/A
Type	Int64	N/A
Descript ion	This property contains the number of occurences where the controller detected an unrecovered data integrity error for the Endurance Group.	N/A
LongDesc ription	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.	N/A
Mandator y Notes	Do Not Implement	DNI

6.7.2.19 **NVMeEnduranceGroupProperties.EndGrpLifetime.ErrorInformationLogEntryCount**

The mapping for `NVMeEnduranceGroupProperties.EndGrpLifetime.ErrorInformationLogEnt`
is summarized in Table 234.

Table 234:

NVMeEnduranceGroupProperties.EndGrpLifetime.ErrorInformationLogEntryCount

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeEnduranceGroupPropertie s.EndGrpLifetime.ErrorInfor mationLogEntryCount	N/A
Type	Int64	N/A
Descript ion	This property contains the number of error information log entries over the life of the controller for the endurance group.	N/A
LongDesc ription	This property shall contain the number of error information log entries over the life of the controller for the endurance group.	N/A
Mandator y Notes	Do Not Implement	DNI

6.7.2.20 **NVMeSetProperties.SetIdentifier**   The mapping for `NVMeSetProper-ties.SetIdentifier` is summarized in Table 235.

Table 235: NVMeSetProperties.SetIdentifier

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeSetProperties.SetIdenti fier	**NVM Spec Property / Field:** Identify Command / NVM Set List (CNS 04h) / NVM Set Attributes Entry / NVM Set Identifier **NVM Spec: Section:Figure** NVMe 2.0: Section 5.15.2.4, Figure 278: NVM Set Attributes List / Figure 279: NVM Set Attributes Entry: Bytes 00 - 01 (NVM Set Identifier)
Type	String	16 Bit value
Descript ion	A 16-bit hex value that contains the NVMe Set identifier.	This field specifies the NVM Set ID.
LongDesc ription	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.	This field specifies the NVM Set ID. An NVM Set Identifier is a 16-bit value that specifies the NVM Set
Mandator y	Mandatory	Mandatory
Notes		Return as hex value as described in the Swordfish schema.

6.7.2.21 NVMeSetProperties.OptimalWriteSizeBytes The mapping for `NVMeSet-Properties.OptimalWriteSizeBytes` is summarized in Table 236.

Table 236: NVMeSetProperties.OptimalWriteSizeBytes mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeSetProperties.OptimalWriteSizeBytes	**NVM Spec Property / Field:** Identify Command / NVM Set List (CNS 04h) / NVM Set Attributes Entry / Optimal Write Size **NVM Spec: Section:Figure** NVMe 2.0: Section 5.15.2.4, Figure 278: NVM Set Attributes List / Figure 279: NVM Set Attributes Entry: Bytes 12 - 15 (Optimal Write Size)
Type	Int64	4-Bytes
Description	This property contains the Optimal Write Size in Bytes for this NVMe Set.	This field indicates the size in bytes for optimal write performance.
LongDescription	This property shall contain the Optimal Write Size in Bytes for this NVMe Set.	This field indicates the size in bytes for optimal write performance. A value of 0h indicates that no Optimal Write Size is specified. This field should be cleared to 0h when namespaces within an NVM Set have different User Data Formats that do not allow an Optimal Write Size to be specified.
Mandatory	Mandatory	Mandatory
Notes		

6.7.2.22 NVMeSetProperties.EnduranceGroupIdentifier The mapping for NVMe-
`SetProperties.EnduranceGroupIdentifier` is summarized in Table 237.

Table 237: NVMeSetProperties.EnduranceGroupIdentifier mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeSetProperties.Endurance GroupIdentifier	**NVM Spec Property / Field:** Identify Command / NVM Set List (CNS 04h) / NVM Set Attributes Entry / Endurance Group Identifier **NVM Spec: Section:Figure** NVMe 2.0: Section 5.15.2.4, Figure 278: NVM Set Attributes List / Figure 279: NVM Set Attributes Entry: Bytes 02 - 03 (Endurance Group Identifier)
Type	String	2iBytes
Descript ion	A 16-bit hex value that contains the endurance group identifier.	This field indicates the Endurance Group for this NVM Set.
LongDesc ription	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.	This field indicates the Endurance Group for this NVM Set.
Mandator y	Mandatory	Mandatory
Notes		

6.7.2.23 NVMeSetProperties.Random4kReadTypicalNanoSeconds   The   mapping
for `NVMeSetProperties.Random4kReadTypicalNanoSeconds` is summarized
in Table 238.

Table 238: NVMeSetProperties.Random4kReadTypicalNanoSeconds mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeSetProperties.Random4kReadTypicalNanoSeconds	**NVM Spec Property / Field:** Identify Command / NVM Set List (CNS 04h) / NVM Set Attributes Entry / Random 4 KiB Read Typical **NVM Spec: Section:Figure** NVMe 2.0: Section 5.15.2.4, Figure 278: NVM Set Attributes List / Figure 279: NVM Set Attributes Entry: Bytes 08 - 11 (Random 4 KiB Read Typical)
Type	Int64	4 Bytes
Descript ion	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.	This field indicates the typical time to complete a 4 KiB random read in 100 nanosecond units

	Redfish/Swordfish	NVMe / NVMe-oF
LongDescription	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.	This field indicates the typical time to complete a 4 KiB random read in 100 nanosecond units when the NVM Set is in a Predictable Latency Mode Deterministic Window and there is 1 outstanding command per NVM Set
Mandatory	Mandatory	
Notes		Convert from 100 nanosecond units to nanosecond units.

6.7.2.24 NVMeSetProperties.UnallocatedNVMNamespaceCapacityBytes   The mapping for NVMeSetProperties.UnallocatedNVMNamespaceCapacityBytes is summarized in Table 239.

Table 239: NVMeSetProperties.UnallocatedNVMNamespaceCapacityBytes mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMeSetProperties.Unallocat edNVMNamespaceCapacityBytes	**NVM Spec Property / Field:** Identify Command / NVM Set List (CNS 04h) / NVM Set Attributes Entry / Unallocated NVM Set Capacity **NVM Spec: Section:Figure** NVMe 2.0: Section 5.15.2.4, Figure 278: NVM Set Attributes List / Figure 279: NVM Set Attributes Entry: Bytes 32 - 47 (Unallocated NVM Set Capacity)
Type	Int64	Int 64
Descript ion	Indicates the unallocated capacity of the NVMe Set in bytes.	This field indicates the unallocated NVM capacity in this NVM Set.
LongDesc ription	This property shall contain the unallocated capacity of the NVMe Set in bytes.	This field indicates the unallocated NVM capacity in this NVM Set. The value is in bytes.
Mandator y	Mandatory	
Notes		

6.7.2.25 Status.State   The mapping for Status.State is summarized in Table 240.

Table 240: Status.State mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Status.State	N/A
Type	Resource.State (enum)	N/A
Descript ion	The known state of the resource, such as, enabled.	N/A
LongDesc ription	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.	N/A
Mandator y	Optional	DNI
Notes	Possible values: Enabled / Disabled / StandbyOffline / StandbySpare / InTest / Starting / ABsent / UnavaialableOffline / Deferring / Quiesced / Updating / Qualified	

6.7.2.26 Status.Health   The mapping for `Status.Health` is summarized in Table 241.

Table 241: Status.Health mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Status.Health	N/A
Type	Resource.Health	N/A
Descript ion	The health state of this resource in the absence of its dependent resources.	N/A
LongDesc ription	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.	N/A
Mandator y	Optional	DNI
Notes	Possible Values: OK / Warning / Critical	

6.7.3 NVMe Domains

6.7.3.1 Overview   NVMe Domains are used to subdivide an NVM Subsystem. NVMe domains are different from traditional fault domains, and provide a basis for a broader range of resource groupings. For example, if there are multiple power sources, the domain is used to represent the scope of each power source.

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

NVMeDomains also may support Asymmetric Namespace Access for domain members.

6.7.3.2 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;

- a set of capacity information properties about this set of domain members.

6.7.4 Mockup

The following mockup shows a sample representation of an NVMe Domain, which contains an IO Controller and a single namespace.

```
{
    "@odata.type": "#NVMeDomain.v1_2_0.NVMeDomain",
    "Id": "NVMeDomain1",
    "Name": "NVMeDomain 1",
  "Description": "NVMeDomain contains NVMe I/O controllers from 1 device.",
    "DomainContents": {
        "Controllers": [
            {
        "@odata.id": "/redfish/v1/Systems/Sys-1/Storage/NVMeSSD-EG/Controll
            }
        ],
        "Namespaces": [
```

```
            {
              "@odata.id": "/redfish/v1/Storage/NVMeoF-SS1/Volumes/LogicalNS1"
              }
          ]
      },
      "ANAGroupId": 2346,
      "TotalDomainCapacityBytes": 8575650934756566,
      "UnallocatedDomainCapacityBytes": 0,
    "MaximumCapacityPerEnduranceGroupBytes": 4287825467378283,
      "MaxNamespacesSupportedPerController": 10,
      "FirmwareImages": [
          {
          "@odata.id": "/redfish/v1/UpdateService/FirmwareInventory/NVMeSSD-1.2
          },
          {
          "@odata.id": "/redfish/v1/UpdateService/FirmwareInventory/NVMeSSD-1.2
          }
      ],
      "@odata.id": "/redfish/v1/NVMeDomains/NVMeDomain1",
    "@Redfish.Copyright": "Copyright 2015-2024 SNIA. All rights reserved."
}
```

The following mockup shows a sample representation of an NVMe namespace within an
NVMe Domain.

```
{
    "@odata.type": "#Volume.v1_10_0.Volume",
    "Id": "LogicalNS1",
    "Name": "LogicalNamespace1",
    "Status": {
        "State": "Enabled"
    },
    "ALUA": {
        "ANAGroupId": 2346
    },
    "CapacitySources": [
```

```
        {
            "@odata.type": "#Capacity.v1_2_0.CapacitySource",
            "Id": "NVMeoF_Source1",
            "Name": "NVMeoF Source1",
            "ProvidedCapacity": {
                "Data": {
                    "AllocatedBytes": 10737418240
                }
            },
            "ProvidingPools": {
          "@odata.id": "/redfish/v1/Storage/NVMeoF-SS1/Volumes/LogicalNS1/Cap
            },
        "@odata.id": "/redfish/v1/Storage/NVMeoF-SS1/Volumes/LogicalNS1/Capac
        }
    ],
    "@odata.id": "/redfish/v1/Storage/NVMeoF-SS1/Volumes/LogicalNS1",
    "@Redfish.Copyright": "Copyright 2015-2023 SNIA. All rights reserved."
}
```

6.7.5 Property Mapping

6.7.5.1 ANAGroupID   The mapping for ANAGroupID is summarized in Table 242.

Table 242: ANAGroupID mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Prope rty	ANAGroupID	Retrieved in the Identify CommandANA Group ID retrieved in the I/O Command Set Independent Identify Namespace data structure (CNS 08h) or in the Identify Namespace data structure for the specified NSID or the NVM Command Set Identify Namespace Data Structure (CNS 00h)NVMe Base Specification: Section 5.17.2.8: Figure 280: Bytes 07:04 - ANA Group Identifier (ANAGRPID) NVMe NVM Command Set Specification: Section 4.1.5.1: Figure 97: Bytes 92:95 ANA Group Identifier (ANAGRPID)
Type	Edm.Decimal	DWORD (4 bytes)
Descr iptio n	The ANA group id for all namespaces within the domain.	The ANA Group Identifier associated with all namespaces in an ANA Group described by an ANA Group Descriptor. All namespaces in an ANA Group described by an ANA Group Descriptor shall have the same ANA Group ID (ANAGroupID).

	Redfish/Swordfish	NVMe / NVMe-oF
LongD escri ption	This shall contain the ANA group id which applies to all namespaces within the domain. This corresponds to the value in the ANAGroupId field in volume.	The ANA Group Identifier associated with all namespaces in an ANA Group described by an ANA Group Descriptor. When retrieved via the Identify Command (Command dword 11) For NSID other than FFFFFFFFh, this field indicates the ANA Group Identifier of the ANA group of which the namespace is a member. Each namespace that is attached to a controller that supports Asymmetric Namespace Access Reporting (refer to the CMIC field) shall report a valid ANAGRPID.
Optio nal		Optional
Notes		The ANA Group Identifier (ANAGRPID) for each ANA Group shall be unique within an NVM subsystem. If the controller does not support Asymmetric Namespace Access Reporting, then this field shall be cleared to 0h.

6.7.5.2 FirmwareImages   The mapping for `FirmwareImages` is summarized in Table 243.

Table 243: FirmwareImages mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	FirmwareImages	N/A
Type	Collection(SoftwareInventory.SoftwareInventory)	N/A
Descript ion	Contains an array of pointers to available firmware images.	N/A
LongDesc ription	This property shall contain an array of pointers to available firmware images.	N/A
Mandator y	Mandatory	
Notes	This contains a pointer to another collection within the RF/SF service (the update service). Within this service, the firmware images may contain pointers to images corresponding to either NVMe device slots or to external firmware blobs.	

6.7.5.3 TotalDomainCapacityBytes   The mapping for `TotalDomainCapacityBytes` is summarized in Table 244.

Table 244: TotalDomainCapacityBytesmapping

	Redfish/Swordfish	NVMe / NVMe-oF
Prope rty	TotalDomainCapacityBytes	Retrieved in the Identify CommandFor an NVM subsystem that does not support multiple domains, the capacity information reported in the TNVMCap field of the Identify Controller data structure (CNS 01h) for the controller processing the command. For an NVM subsystem that supports multiple domains, the Identify command may be used to access the Domain List data structure (CNS 18h) to determine the domains that are accessible by the controller and the Total Domain Capacity for each of those domains.For an NVM subsystem that does not support multiple domains: NVMe Base Specification: Section 5.17.2.1: Figure 275: Bytes 295:280 - Total NVM Capacity (TNVMCAP)For an NVM subsystem that supports multiple domains: Section 5.17.2.17: Figure 286 - Domain List; select the entry for the Domain beginning at Byte 128 (domain entries are 256 Bytes in length) and retrieving the Total Domain Capacity for the Domain (Bytes 16:31) in the Domain Attributes Entry data structure
Type	Edm.Int64	Bytes (16)

	Redfish/Swordfish	NVMe / NVMe-oF
Descr iptio n	The total capacity in bytes of this NVMe Domain.	Indicates the total NVM capacity in this Domain.
LongD escri ption	This property shall contain the total capacity in bytes of this NVMe Domain.	For an NVM subsystem that does not support multiple domains: This field indicates the total NVM capacity that is accessible by the controller.For an NVM subsystem that supports multiple domains: This field indicates the total NVM capacity in this Domain.
Manda tory		For an NVM subsystem that does not support multiple domains: Total NVM Capacity (TNVMCAP) is mandatory if the Namespace Management capability is supported or if the Capacity Management capability is supported. For an NVM subsystem that supports multiple domains: The Domain List is Mandatory for controllers that support Capacity Management in an NVM subsystem that supports multiple domains.
Notes		

6.7.5.4 UnallocatedDomainCapacityBytes   The mapping for `UnallocatedDomainCapacityBytes` is summarized in Table 245.

Table 245: UnallocatedDomainCapacityBytes

	Redfish/Swordfish	NVMe / NVMe-oF
Prope rty	UnallocatedDomainCapacityByte s	Retrieved in the Identify CommandFor an NVM subsystem that does not support multiple domains, the capacity information reported in the Unallocated NVM Capacity (UNVMCAP) field of the Identify Controller data structure (CNS 01h) for the controller processing the command.For an NVM subsystem that supports multiple domains, the Identify command may be used to access the Domain List data structure (CNS 18h) to determine the domains that are accessible by the controller and the Unallocated Domain Capacity for each of those domains.For an NVM subsystem that does not support multiple domains: NVMe Base Specification: Section 5.17.2.1: Figure 275: Bytes 296: 311 - Unallocated NVM Capacity (UNVMCAP)For an NVM subsystem that supports multiple domains: Section 5.17.2.17: Figure 286 - Domain List; select the entry for the Domain beginning at Byte 128 (domain entries are 256 Bytes in length) and retrieving the Unallocated Domain Capacity for the Domain (Bytes 32:47) in the Domain Attributes Entry data structure

	Redfish/Swordfish	NVMe / NVMe-oF
Type	Edm.Int64	Bytes (16)
Description	The total capacity in bytes of this NVMe Domain.	Indicates the unallocated NVM capacity in this Domain.
LongDescription	This property shall contain the total capacity in bytes of this NVMe Domain.	For an NVM subsystem that does not support multiple domains: This field indicates the unallocated NVM capacity that is accessible by the controller.For an NVM subsystem that supports multiple domains: This field indicates the total NVM capacity in this Domain.
Mandatory		For an NVM subsystem that does not support multiple domains: Unallocated NVM Capacity (UNVMCAP) is mandatory if the Namespace Management capability is supported or if the Capacity Management capability is supported.For an NVM subsystem that supports multiple domains: The Domain List and the Unallocated Domain Capacity for the domain is Mandatory for controllers that support Capacity Management in an NVM subsystem that supports multiple domains (refer to section 3.2.4).
Notes		

6.7.5.5 MaximumCapacityPerEnduranceGroupBytes   The mapping for `MaximumCapacityPerEnduranc`
is summarized in Table 246.

Table 246: MaximumCapacityPerEnduranceGroupBytes

	Redfish/Swordfish	NVMe / NVMe-oF
Prope rty	MaximumCapacityPerEnduranceGr oupBytes	The list of Endurance Groups may be retrieved in the Identify Command with the Endurance Group ID CNS value (19h)The Total Endurance Group Capacity (TEGCAP) may be retrieved in the Endurance Group Log Page (Log ID 09h) by supplying a valid Endurance Group IDNVMe Basse Specification 2.0 Section 5.16.1.0: Figure 217 - Total Endurance Group Capacity (TEGCAP): Bytes 160:175)
Type	Edm.Int64	Bytes (16)
Descr iptio n	The maximum capacity per endurance group in bytes of this NVMe Domain.	
LongD escri ption	This property shall contain the maximum capacity per endurance group in bytes of this NVMe Domain.	
Optio nal		Optional: An NVM subsystem that does not support multiple Endurance Groups does not require reporting of Endurance Groups
Notes		

6.7.5.6 MaxNamespacesSupportedPerController The mapping for `MaxNamespacesSupportedPerCon`
is summarized in Table 247.

Table 247: MaxNamespacesSupportedPerController

	Redfish/Swordfish	NVMe / NVMe-oF
Prope rty	MaxNamespacesSupportedPerCont roller	**Retrieved in the Identify CommandNVM Spec Property / Field:** Identify Controller Data Structure, I/O Command Set Independent / Maximum I/O Domain Namespace Attachments (MAXDNA) NVMe Base Specification 2.0: Section 5.17.2.1: Figure 275: Identify – Identify Controller Data Structure, I/O Command Set Independent: Bytes 559:544 - Maximum I/O Domain Namespace Attachments (MAXDNA)
Type	Edm.Decimal	Bytes (4)
Descr iptio n	The maximum number of namespace attachments of this NVMe Domain.	The maximum of the sum of the number of namespaces attached to each I/O controller in the Domain.
LongD escri ption	This property shall contain the maximum number of namespace attachments supported in this NVMe Domain. If there are no limits imposed, this property should not be implemented.	The maximum of the sum of the number of namespaces attached to each I/O controller in the Domain.
Optio nal		Optional

Redfish/Swordfish	NVMe / NVMe-oF
Notes	If this field is cleared to 0h, then no maximum is specified. The value of this field shall be the same value for all I/O controllers in the Domain.

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.type": "#Drive.v1_17_0.Drive",
    "Id": "0THGR0KP",
    "Name": "Drive 1",
    "Status": {
        "State": "Enabled",
        "Health": "OK"
    },
    "StatusIndicator": "OK",
    "CapacityBytes": 300067890136,
    "ConfigurationLock": "Partial",
    "TargetConfigurationLockLevel": "Standard",
    "NVMe": {
        "ConfigurationLockState": {
        "FirmwareCommit": "Unlocked",
        "Lockdown": "Locked",
        "SecureSend": "LockdownUnsupported",
        "FirmwareImageDownload": "LockdownUnsupported",
        "VPDWrite": "CommandUnsupported"
        }
    },
    "FailurePredicted": false,
    "Protocol": "SAS",
    "MediaType": "HDD",
    "Manufacturer": "HDD-Company",
    "SerialNumber": "0THGR0KP",
```

```
    "PartNumber": "HUC156030CSS200",
    "Identifiers": [
        {
            "DurableNameFormat": "NAA",
            "DurableName": "300062B202B21849"
        }
    ],
    "RotationSpeedRPM": 15000,
    "BlockSizeBytes": 512,
    "NegotiatedSpeedGbs": 12,
    "Metrics": {
      "@odata.id":
      ↪    "/redfish/v1/Chassis/StorageEnclosure1/Drives/0THGR0KP/Metrics"
    },
    "Links": {
        "Volumes": [
            {
                "@odata.id":
                ↪    "/redfish/v1/Systems/Sys-1/Storage/DirectAttachStorageSys
            }
        ],
        "StoragePools": [
            {
                "@odata.id":
                ↪    "/redfish/v1/Systems/Sys-1/Storage/DirectAttachStorageSys
                ↪    SimpleSystemPool"
            }
        ]
    },
    "@odata.id":
    ↪    "/redfish/v1/Chassis/StorageEnclosure1/Drives/0THGR0KP",
    "@Redfish.Copyright": "Copyright 2015-2024 SNIA. All
    ↪    rights reserved."
}
```

6.8.2 Property Mapping

6.8.2.1 Actions.#Drive.Reset   The mapping for `Actions.#Drive.Reset` is summarized in Table 248.

Table 248: Actions.#Drive.Reset mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Actions.#Drive.Reset	**NVM Spec Property / Field:** Set Property Command / NSSR: NVM Subsystem Reset (Controller Property Offset 20h)NSSD: NVM Subsystem Shutdown (Controller Property Offset 64h) **NVM Spec: Section:Figure** NVMe 2.0: Section 3.1.3: Figure 35: Offset 20h and 64h Further information on NVM Subsystem Reset Control (NSSRC): NVMe 2.0: Section 3.1.3.7: Figure 48 and NVM Subsystem Shutdown Control (NSSD): NVMe 2.0: Section 3.1.3.20: Figure 61
Type	Action (Special form of POST)	NVMe Administrative command
Descript ion	This action resets this drive.	This is used to initiate a controller reset or shutdown depending on the control offset used.

	Redfish/Swordfish	NVMe / NVMe-oF
LongDescription	This action shall reset this drive.	NVM Subsystem Reset Control (NSSRC): A write of the value 4E564D65h ("NVMe") to this field initiates an NVM Subsystem Reset. NVM Subsystem Shutdown Control (NSSC): A write of the value 4E726D6Ch ("Nrml") to this field initiates a normal NVM Subsystem Shutdown on every controller (in the domain associated with the controller when CAP.CPS is set to 10b as specified in section 3.6.3.1; or • in the NVM subsystem when CAP.CPS is set to 11b in the NVM subsystem as specified in section 3.6.3.2. A write of the value 41627074h ("Abpt") to this field initiates an abrupt NVM subsystem shutdown on every controller: • in the domain associated with the controller when CAP.CPS is set to 10b as specified in section 3.6.3.1; in the NVM subsystem when CAP.CPS is set to 11b in the NVM subsystem
Mandator y	OptionalMandatory for NVMe Drives	Optional

	Redfish/Swordfish	NVMe / NVMe-oF
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 Subsystem 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 249.

Table 249: 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 2.0: 5.24 Sanitize Command NVM Base Specification 2.0: Section 8.21 Sanitize Operations NVM Base Specification 2.0: Annex A "Sanitize Operation Considerations (Informative)"
Type	Action (Special form of POST)	NVMe Administrative command
Descript ion	This action securely erases the contents of the drive.	The sanitize administrative command operation makes all user data previously written to the device inaccessible.

	Redfish/Swordfish	NVMe / NVMe-oF
LongDescription	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. All user data in the NVM subsystem is altered such that recovery of the previous user data from any cache or the non-volatile media is not possible.
Mandator y	Mandatory	Optional
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. 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)

6.8.2.3 Assembly.BinaryDataURI   The mapping for `Assembly.BinaryDataURI` is summarized in Table 250.

Table 250: Assembly.BinaryDataURI mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Assembly.BinaryDataURI	N/A
Type	String	N/A
Descript ion	The URI at which to access an image of the assembly information.	N/A
LongDesc ription	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.	N/A
Mandator y	Recommended	
Notes		N/A for NVMe 2.0

6.8.2.4 BlockSizeBytes   The mapping for `BlockSizeBytes` is summarized in Table 251.

Table 251: BlockSizeBytes mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	BlockSizeBytes	**NVM Spec Property / Field:** Formatted LBA Size (FLBAS) retrieved in some Identify Namespace data structures for the specified NSID or the common namespace capabilities for the NVM Command Set (CNS 00h).Refer to the applicable NVMe I/O Command Set specification for details. For NVMe I/O Command Sets that don't define this field, it is considered reserved**NVM Spec: Section:Figure** NVMe NVM Command Set Specification 1.0b: Section 4.1.5.1: FIgure 97: Bytes 26 - Formatted LBA Size (FLBAS) and uses the Number LBA Formats (NBLAF) field defined in Section 4.1.5.1: FIgure 97: Bytes 25 (Number of LBA Formats)NVM Express NVM Zoned Namespace Command Set Specification 1.1b: Section A.5: Figure 53 (Size and Capacity Fields): Bytes 26.
Type	Int64	Int-64

	Redfish/Swordfish	NVMe / NVMe-oF
Descript ion	The size, in bytes, of the smallest addressable unit, or block.	The LBA data size & metadata size combination that the namespace has been formatted with.
LongDesc ription	This property shall contain size of the smallest addressable unit of the associated drive or device.	This field indicates the LBA data size & metadata size combination that the namespace has been formatted with.
Mandator y	Mandatory	Only Mandatory / applicable for the NVM Command Set and I/O Command Sets based on the NVM Command Set. Prohibited for all other I/O Command Sets.

Redfish/Swordfish	NVMe / NVMe-oF
Notes	Refer to the applicable NVMe I/O Command Set specification for details. For NVMe I/O Command Sets that don't define this field, it is considered reserved.For NVM Express Command Sets that Support Blocks: Bits 6:5 indicate the most significant 2 bits of the Format Index of the supported LBA Format indicated in this data structure that was used to format the namespace. If the NLBAF field is less than or equal to 16, then the host should ignore these bits.Bit 4 indicates whether or not the metadata is transferred at the end of the data LBA, creating an extended data LBA. (note: Bit 4 is not applicable when there is no metadata).Bits 3:0 indicate the least significant 4 bits of the Format Index of the supported LBA Format indicated in this data structure that was used to format the namespace.

6.8.2.5 CapableSpeedGpbs The mapping for `CapableSpeedGpbs` is summarized in Table 252.

Table 252: CapableSpeedGpbs mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	CapableSpeedGpbs	**NVM Spec Property / Field:** PCI Express Link Capabilities / PXCAP+Ch **NVM Spec: Section:Figure** NVMe PCIe Transport Specification 1.0b: 3.8.5.6: Figure 53: Bits 00:03
Type	Decimal	
Descript ion	The speed, in gigabit per second (Gbit/s), at which this drive can communicate to a storage controller in ideal conditions.	indicates the supported Link speed(s) of the associated port.
LongDesc ription	This property shall contain fastest capable bus speed, in gigabit per second (Gbit/s), of the associated drive.	This field indicates the supported Link speed(s) of the associated port.
Mandator y	Mandatory	Mandatory
Notes		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 254.

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

Table 253: CapacityBytes for single namespace mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	CapacityBytes	**NVM Spec Property / Field:** – Identify Command / Identify Namespace Data Structure (CNS 00h), NVM Command Set / Namespace Size (NSZE) **NVM Spec: Section:Figure** NVMe 2.0: 5.17.1: FIgure 273: CNS Value 00h NVM Command Set Specification 1.0b: 4.1.5.1: Figure 97: Bytes 00:07
Type	Int64	
Descript ion	The size, in bytes, of this drive.	The total size of the NVM allocated its namespace.

	Redfish/Swordfish	NVMe / NVMe-oF
LongDesc ription	This property shall contain the raw size, in bytes, of the associated drive.	The total size of the NVM allocated to this namespace in logical blocks. The value is in bytes. This field shall be supported if the Namespace Management capability 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.
Mandator y	Mandatory	Optional
Notes	Reporting capacity in bytes is the Redfish and Swordfish standard mechanism.	

For drives supporting multiple namespaces:

Table 254: CapacityBytes for multiple namespace mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	CapacityBytes	**NVM Spec Property / Field:** Endurance Group Information Log (Log ID 09h) / Total Endurance Group Capacity (TEGCAP) **NVM Spec: Section:Figure** NVMe 2.0: 5.16.1.10: Figure 217: Bytes 160 : 175
Type	Int64	16 Bytes
Descript ion	The number of bytes currently allocated by the storage system in this data store for this data type.	indicates the total NVM capacity in this Endurance Group.
LongDesc ription	The value shall be the number of bytes currently allocated by the storage system in this data store for this data type.	This field indicates the total NVM capacity in this Endurance Group.
Mandator y	Mandatory	No
Notes		This value is in bytes

6.8.2.7 ConfigurationLock   The mapping for `ConfigurationLock` is summarized in Table 255.

Table 255: ConfigurationLock mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	ConfigurationLock	**NVM Spec: Property / Field:**NVMe 2.0d: Identify Command / Identify Controller data structure (CNS 01h) - Optional Admin Command Support (OACS)**NVM Spec: Section:Figure**NVMe 2.0d: Section 5.17.2.1: Figure 275 - 257:256, Bit 10) Get log page specifying the Command and Feature Lockdown (Log Page Identifier 14h) **NVM Spec: Section:Figure**NVMe 2.0d: Section 5.16.1.20
Type	enum (ConfigurationLock)	Log Page

	Redfish/Swordfish	NVMe / NVMe-oF
Description	The `ConfigurationLock`, `TargetConfigurationLockLevel`, and `ConfigurationLockState` properties are jointly used to manage lockdown of NVMe devices.	The Lockdown command is used to control the Command and Feature Lockdown capability which configures the prohibition or allowance of execution of the specified command or Set Features command targeting a specific Feature Identifier. After a successful completion of a Lockdown command prohibiting a command or Feature Identifier, all controllers, if applicable, and all management endpoints, if applicable, in the NVM subsystem behave as requested.

	Redfish/Swordfish	NVMe / NVMe-oF
LongDescription	This has three states: Enabled, Disabled, and Partial. • `Enabled`: All supported properties, as reported in the `ConfigurationLockState`, that are part of the `TargetConfigurationLockLevel` are locked. • `Disabled`: All supported properties, as reported in the `ConfigurationLockState`, that are part of the `TargetConfigurationLockLevel` are unlocked.• `Partial`: The properties, as reported in the `ConfigurationLockState`, that are part of the `TargetConfigurationLockLevel` are not consistently locked or unlocked. Services shall reject modification requests that contain the value `Partial`.	Command and Feature Lockdown is used to prohibit the execution of commands submitted to NVM Express controllers and/or Management Endpoints in an NVM subsystem. Commands and Feature Identifiers are defined with the following scopes: • Admin Command Set commands defined by the Opcode field;• Set Features command features defined by the Feature Identifier field;• Management Interface Command Set commands defined by the Opcode field; and• PCIe Command Set commands defined by the Opcode field (refer to the NVM Express Management Interface Specification).
Mandatory	Optional	Mandatory to implement OACS; Lockdown feature is optional.

	Redfish/Swordfish	NVMe / NVMe-oF
Notes	This property is a duplicate of the property in the Subsystem, and should reflect that state. See section `Mapping NVMe Command and F eature Lockdown to "Configu rationLocked"` for detailed behavior and mapping description.	

6.8.2.8 Description   The mapping for `Description` is summarized in Table 256.

Table 256: Description mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Description	N/A
Type	String	N/A
Descript ion	The description of this resource.	N/A
LongDesc ription	This object represents the description of this resource. The resource values shall comply with the Redfish Specification-described requirements.	N/A
Mandator y	Mandatory	
Notes	In Redfish, Description is a read-only field.	

6.8.2.9 EncryptionAbility   The mapping for `EncryptionAbility` is summarized in Table 257.

Table 257: EncryptionAbility mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	EncryptionAbility	N/A
Type	Enum (EncryptionAbility)	N/A
Descript ion	The encryption ability of this drive.	N/A
LongDesc ription	This property shall contain the encryption ability for the associated drive.	N/A
Mandator y	Mandatory	
Notes	Required when encryption is supported. Available values: None/SelfEncryptingDrive/Ot her	Maps to vendor capabilities.

6.8.2.10 EncryptionStatus   The mapping for `EncryptionStatus` is summarized in Table 258.

Table 258: EncryptionStatus mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	EncryptionStatus	N/A
Type	Drive.EncryptionStatus	N/A
Descript ion	The status of the encryption of this drive.	N/A
LongDesc ription	This property shall contain the encryption status for the associated drive.	N/A
Mandator y	OptionalMandatory when EncryptionAbility != None	
Notes	Must be implemented and set when Encryption is enabled (EncryptionAbility will indicate encryption capability type.) Possible values: Unlocked/locked/foreign/une ncrypted	Set according to vendor specs / mapping.

6.8.2.11 FailurePredicted   The mapping for `FailurePredicted` is summarized in Table 259.

Table 259: FailurePredicted mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	FailurePredicted	**NVM Spec: Property / Field:** – Persistent Event Log Log Identifier 0Dh) / NVM Subsystem Hardware Error Event (Event Type 05h)**NVM Spec: Section:Figure**NVMe 2.0: 5.16.1.14: Persistent Event Log (Log Identifier 0Dh) / Figure 224: Get Log Page – Persistent Event Log (Log Identifier 0Dh)NVMe 2.0: 5.16.1.14.1.5: FIgure 232: NVM Subsystem Hardware Error Event Format: Bytes 00:01 (NVM Subsystem Hardware Error Event Code)
Type	Boolean	If any of the NVM Subsystem Hardware Events exist then this is a binary "true"
Descript ion	An indication of whether this drive currently predicts a failure in the near future.	The Persistent Event Log page contains information about significant events not specific to a particular command.

	Redfish/Swordfish	NVMe / NVMe-oF
LongDescription	This property shall indicate whether this drive currently predicts a manufacturer-defined failure.	The Persistent Event Log page contains information about significant events not specific to a particular command. The information in this log page shall be retained across power cycles and resets.
Mandatory	Recommended	Optional for Admin and I/O Controllers. Prohibited for Discovery Controllers
Notes	Implements the results of the SMART log data.	Reference NVMe 2.0: Section 5.16.1.14.1.5 Figure 233 for possible NVM Subsystem Hardware Error Event Codes

6.8.2.12 Identifiers   The mapping for `Identifiers` is summarized in Table 260.

Table 260: Identifiers mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Identifiers	N/A
Type	Collection(Resource.Identif ier)	N/A
Descript ion	The Durable names for the subsystem.	N/A
LongDesc ription	This property shall contain a list of all known durable names for the associated subsystem.	N/A
Mandator y	Mandatory	
Notes	This is an array of unique identifiers for the NVM Subsystem.	This is an array of unique identifiers for the NVM Subsystem

6.8.2.13 Identifiers.DurableNameFormat   The mapping for `Identifiers.DurableNameFormat` is summarized in Table 261.

Table 261: Identifiers.DurableNameFormat mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Identifiers.DurableNameForm at	N/A
Type	Resource.v1_1_0.DurableNa meFormat	N/A
Descript ion	The format of the Durable names for the subsystem.	N/A
LongDesc ription	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).	N/A
Mandator y	Optional	
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.	

6.8.2.14 Identifiers.DurableName   The mapping for `Identifiers.DurableName` is summarized in Table 262.

Table 262: Identifiers.DurableName mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Identifiers.DurableName	**NVM Spec: Property / Field:**NVMe 2.0: Identify Command / Identify Controller data structure (CNS 01h) / NVM Subsystem NVMe Qualified Name (SUBNQN)**NVM Spec: Section:Figure**NVMe 2.0: Section 5.17.2.1: Figure 275 - Bytes 768 - 1023)
Type	Edm.String	UTF-8 null-terminated string. Refer to NVMe 2.0: Section 4.5.1 Unique Identifier
Descript ion	The format of the Durable names for the subsystem.	NVM Subsystem NVMe Qualified Name (SUBNQN)
LongDesc ription	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).	Used to uniquely describe a host or NVM subsystem for the purposes of identification and authentication.

	Redfish/Swordfish	NVMe / NVMe-oF
Mandator y	Optional	Support for this field is mandatory if the controller supports revision 1.2.1 or later as indicated in the Controller Properties Version register (refer to section 3.1.3), Figure 35, Offset 8h.
Notes	For this particular usage in Subsystem, there will only be one instance populated in the identifiers array.	

6.8.2.15 IndicatorLED   The mapping for `IndicatorLED` is summarized in Table 263.

Table 263: IndicatorLED mapping

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

6.8.2.16 Links.Volumes   The mapping for `Links.Volume` is summarized in Table 264.

Table 264: Links.Volume mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Links.Volume	N/A
Type	Collection(Volume.Volume)	N/A
Descript ion	An array of links to the volumes that this drive either wholly or only partially contains.	N/A
LongDesc ription	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`.	N/A
Mandator y	Mandatory	
Notes	This array shall contain links to all namespaces associated with this physical drive.	

6.8.2.17 Links.Volumes@odata.count   The mapping for `Links.Volumes@odata.count` is summarized in Table 265.

Table 265: Links.Volumes@odata.count mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Links.Volumes@odata.count	N/A
Type	(odata property)int64	N/A
Descript ion	Count of the number of items in the Links.Volume array.	N/A
LongDesc ription		N/A
Mandator y	Mandatory	
Notes		The number of namespaces is available from NVMe on a per controller basis.

6.8.2.18 Location   The mapping for Location is summarized in Table 266.

Table 266: Location mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Location	N/A
Type	Collection(Resource.Locatio n)	N/A
Descript ion	The location of the drive.	N/A
LongDesc ription	This property shall contain location information of the associated drive.	N/A
Mandator y	Do Not Implement	
Notes	This property has been deprecated. See guidance / implement the PhysicalLocation property instead.	

6.8.2.19 LocationIndicatorActive The mapping for `LocationIndicatorActive` is summarized in Table 267.

Table 267: LocationIndicatorActive mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	LocationIndicatorActive	N/A
Type	Boolean	N/A
Descript ion	An indicator allowing an operator to physically locate this resource.	N/A
LongDesc ription	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
Mandator y	Recommended	
Notes	This property replaces the IndicatorLED, which has been deprecated.	Comes from vendor.

6.8.2.20 Manufacturer   The mapping for `Manufacturer` is summarized in Table 268.

Table 268: Manufacturer mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Manufacturer	**NVM Spec Property / Field:** Identify Command / Identify Controller Data structure (CNS 01h) / PCI Vendor ID (VID)**NVM Spec: Section: Figure** NVMe 2.0: Section 5.17.2.1: Figure 275: Bytes 00:01
Type	String	16-bit number in little endian format.
Descript ion	The manufacturer of this drive.	The company vendor identifier
LongDesc ription	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
Mandator y	Recommended	Mandatory

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.

6.8.2.21 MediaType   The mapping for MediaType is summarized in Table 269.

Table 269: MediaType mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	MediaType	N/A
Type	enum (MediaType)	N/A
Descript ion	The type of media contained in this Drive.	N/A
LongDesc ription	This property shall contain the type of media contained in the associated drive.	N/A
Mandator y	Mandatory	
Notes	Possible values: HDD/SSD/SMR.	NVMe SSD Drives to report SSD.

6.8.2.22 Metrics   The mapping for `Metrics` is summarized in Table 270.

Table 270: Metrics mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Metrics	The DriveMetrics object contains multiple properties that map to NVMe properties, but the object itself does not have a direct mapping.
Type	DriveMetrics.DriveMetrics	
Descript ion	The link to the metrics associated with this drive.	
LongDesc ription	This property shall contain a link to the metrics associated with this drive.	
Mandator y	Optional	
Notes		

6.8.2.23 Model   The mapping for Model is summarized in Table 271.

Table 271: Model mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Model	**NVM Spec Property / Field:** Identify Command / Identify Controller Data Structure (CNS 01h) / Model Number (MN) **NVM Spec: Section: Figure** NVMe 2.0: Section 5.17.2.1: Figure 275: Bytes 24:63
Type	String	String
Descript ion	The model number for the Drive.	Model Number (MN)
LongDesc ription	This property shall contain the name by which the manufacturer generally refers to the Drive.	Contains the model number for the NVM subsystem that is assigned by the vendor as an ASCII string.
Mandator y	Mandatory	
Notes		Refer to NVMe Base specification 2.0 section 4.5.1 for unique identifier requirements. Refer to NVMe Base specification 2.0 section 1.4.2 for ASCII string requirements.

6.8.2.24 Multipath   The mapping for `Multipath` is summarized in Table 272.

Table 272: Multipath mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Multipath	**NVM Spec Property / Field:** Identify Command / Idnetify Controller Data Structure (CNS 01h) / Controller Multi-Path I/O and Namespace Sharing Capabilities**NVM Spec: Section:Figure** NVMe 2.0: Section 5.17.2.1: Figure 275: Byte 76
Type	Boolean	N/A
Descript ion	An indication of whether the drive is accessible from multiple paths.	This field specifies multi-path I/O and namespace sharing capabilities of the controller and NVM subsystem

	Redfish/Swordfish	NVMe / NVMe-oF
LongDescription	This property shall indicate whether the drive is accessible by an initiator from multiple paths allowing for failover capabilities upon a path failure.	Bits 7:4 are reserved.Bit 3 if set to "1", then the NVM subsystem supports Asymmetric Namespace Access Reporting (refer to section NVMe Base Specification 2.0 section 8.1). If cleared to "0", then the NVM subsystem does not support Asymmetric Namespace Access Reporting.Bit 2 if set to "1", then the controller is associated with an SR-IOV Virtual Function. If cleared to "0", then the controller is associated with a PCI Function or a Fabrics connection.Bit 1 if set to "1", then the NVM subsystem may contain two or more controllers. If cleared to "0", then the NVM subsystem contains only a single controller. As described in refer to section NVMe Base Specification 2.0 section 2.4.1, an NVM subsystem that contains multiple controllers may be used by multiple hosts, or may provide multiple paths for a single host.Bit 0 if set to "1", then the NVM subsystem may contain more than one NVM subsystem port. If cleared to "0", then the NVM subsystem contains only a single NVM subsystem port.

	Redfish/Swordfish	NVMe / NVMe-oF
Mandator y	RecommendedMandatory if drive is dual-ported.	Admin: Optional I/O: Optional Discovery: Reserved
Notes		

6.8.2.25 Name   The mapping for Name is summarized in Table 273.

Table 273: Name mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Name	N/A
Type	String	N/A
Descript ion	The name of the resource or array member.	N/A
LongDesc ription	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.	N/A
Mandator y	Mandatory	
Notes	In Redfish, Name is a read-only field.	

6.8.2.26 NegotiatedSpeedGbps   The mapping for `NegotiatedSpeedGbps` is summarized in Table 274.

Table 274: NegotiatedSpeedGbps mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NegotiatedSpeedGbps	For NVMe: **NVM Spec Property / Field:** : For PCIe Transport: Offset PXCAP + Ch: PXLCAP – PCI Express Link Capabilities / Supported Link Speeds (SLS)**NVM Spec: Section:Figure** For PCIe Transport: NVM Express PCIe Transport Specification 1.0b: section 3.8.5.6: Figure 53: Bytes 03:00For NVMe-oF: Not Applicable
Type	Decimal	
Descript ion	The speed, in gigabit per second (Gbit/s), at which this drive currently communicates to the storage controller.	This field indicates the supported Link speed(s) of the associated port
LongDesc ription	This property shall contain current bus speed, in gigabit per second (Gbit/s), of the associated drive.	Supported Link Speeds
Mandator y	Mandatory	For NVMe: RequiredFor NVMe-oF: DNI
Notes		For PCIe, this is in the PCIe link capabilities For NVMe-oF this is not specified; use the value for the native capability.

6.8.2.27 NVMe.NVMeConfigurationLockState   The mapping for `NVMe.NVMeConfigurationLockState` is summarized in Table 275.

Table 275: NVMe.NVMeConfigurationLockState mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMe.NVMeConfigurationLockState	N/A (Sub-properties map to NVMe Command and Feature Lockdown)
Type	ComplexType	N/A
Descript ion	Set of configurable features that are able to be locked on an NVMe Subsystem, and their current lock state.	
LongDesc ription		
Mandator y	Optional	
Notes	See section `Mapping NVMe Command and F eature Lockdown to "Configu rationLocked"` for detailed behavior and mapping description.	

6.8.2.28 NVMe.NVMeConfigurationLockState.FirmwareCommit   The mapping for
NVMe.NVMeConfigurationLockState.FirmwareCommit is summarized in
Table 276.

Table 276: NVMe.NVMeConfigurationLockState.FirmwareCommit mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMe.NVMeConfigurationLockState.FirmwareCommit	**NVM Spec: Property / Field:**NVMe 2.0d: Identify Command / Identify Controller data structure (CNS 01h) - Optional Admin Command Support (OACS)**NVM Spec: Section:Figure**NVMe 2.0d: Section 5.17.2.1: Figure 275 - 257:256, Bytes 13:08) Get log page specifying the Command and Feature Lockdown (Log Page Identifier 14h) **NVM Spec: Section:Figure**NVMe 2.0d: Section 5.16.1.20 (Figure 260)

	Redfish/Swordfish	NVMe / NVMe-oF
Type	Drive.v1_20_0.ConfigLockO ptions • `Unlocked`: Command is supported, able to be locked, and the current state is unlocked. • `Locked`: Command is supported, able to be locked, and the current state is locked. • `LockdownUnsupported`: Command is supported, able to be locked, and the current state is locked. • `CommandUnsupported`: Command is not supported, therefore lockdown does not apply.	Log Page
Descript ion	Ability to perform the NVMe defined FirmwareCommit command.	If the FirmwareCommit feature is able to be locked down, it's corresponding opcode, 10h, will be reported in the command and feature lockdown log page (bytes 13:08).

	Redfish/Swordfish	NVMe / NVMe-oF
LongDesc ription	This has three states: Enabled, Disabled, and Partial. • `Enabled`: All supported properties, as reported in the `ConfigurationLockState`, that are part of the `TargetConfigurationLockLev el` are locked. • `Disabled`: All supported properties, as reported in the `ConfigurationLockState`, that are part of the `TargetConfigurationLockLev el` are unlocked.• `Partial`: The properties, as reported in the `ConfigurationLockState`, that are part of the `TargetConfigurationLockLev el` are not consistently locked or unlocked. Services shall reject modification requests that contain the value `Partial`.	Command and Feature Lockdown is used to prohibit the execution of commands submitted to NVM Express controllers and/or Management Endpoints in an NVM subsystem. Commands and Feature Identifiers are defined with the following scopes: • Admin Command Set commands defined by the Opcode field;• Set Features command features defined by the Feature Identifier field;• Management Interface Command Set commands defined by the Opcode field; and• PCIe Command Set commands defined by the Opcode field (refer to the NVM Express Management Interface Specification).
Mandator y	Optional	Mandatory to implement OACS; FirmwareCommit feature lockdown support is optional. If able to be supported (in log page 14), use the lockdown command to lock and unlock correspond to client requests.

	Redfish/Swordfish	NVMe / NVMe-oF
Notes	See section `Mapping NVMe Command and F eature Lockdown to "Configu rationLocked"` for detailed behavior and mapping description.	

6.8.2.29 NVMe.NVMeConfigurationLockState.FirmwareImageDownload   The mapping for `NVMe.NVMeConfigurationLockState.FirmwareImageDownload` is summarized in Table 277.

Table 277: NVMe.NVMeConfigurationLockState.FirmwareImageDownload mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMe.NVMeConfigurationLockState.FirmwareImageDownload	**NVM Spec: Property / Field:**NVMe 2.0d: Lockdown - **NVM Spec: Section:Figure**NVMe 2.0d: Section 5.19: Figure 292 - 06:00) Get log page specifying and controlling the Command and Feature Lockdown (Log Page Identifier 14h) **NVM Spec: Section:Figure**NVMe 2.0d: Section 5.16.1.20 (Figure 260), bytes 13:08 Log page indicating which commands and features are supported and currently set.
Type	Drive.v1_20_0.ConfigLockOptions • `Unlocked`: Command is supported, able to be locked, and the current state is unlocked. • `Locked`: Command is supported, able to be locked, and the current state is locked. • `LockdownUnsupported`: Command is supported, able to be locked, and the current state is locked. • `CommandUnsupported`: Command is not supported, therefore lockdown does not apply.	Log Page

	Redfish/Swordfish	NVMe / NVMe-oF
Descript ion	Ability to perform the NVMe defined FirmwareImageDownload command.	If the FirmwareImageDownload feature is able to be locked down, it's corresponding opcode, 11h, will be reported in the command and feature lockdown log page (bytes 13:08).
LongDesc ription	This has three states: Enabled, Disabled, and Partial. • `Enabled`: All supported properties, as reported in the `ConfigurationLockState`, that are part of the `TargetConfigurationLockLev el` are locked. • `Disabled`: All supported properties, as reported in the `ConfigurationLockState`, that are part of the `TargetConfigurationLockLev el` are unlocked.• `Partial`: The properties, as reported in the `ConfigurationLockState`, that are part of the `TargetConfigurationLockLev el` are not consistently locked or unlocked. Services shall reject modification requests that contain the value `Partial`.	Command and Feature Lockdown is used to prohibit the execution of commands submitted to NVM Express controllers and/or Management Endpoints in an NVM subsystem. Commands and Feature Identifiers are defined with the following scopes: • Admin Command Set commands defined by the Opcode field;• Set Features command features defined by the Feature Identifier field;• Management Interface Command Set commands defined by the Opcode field; and• PCIe Command Set commands defined by the Opcode field (refer to the NVM Express Management Interface Specification).

	Redfish/Swordfish	NVMe / NVMe-oF
Mandator y	Optional	Mandatory to implement OACS; FirmwareImageDownload feature lockdown support is optional. If able to be supported (in log page 14), use the lockdown command to lock and unlock correspond to client requests.
Notes	See section `Mapping NVMe Command and F eature Lockdown to "Configu rationLocked"` for detailed behavior and mapping description.	

6.8.2.30 NVMe.NVMeConfigurationLockState.Lockdown The   mapping   for `NVMe.NVMeConfigurationLockState.Lockdown` is summarized in Table [278](#).

Table 278: NVMe.NVMeConfigurationLockState.Lockdown mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMe.NVMeConfigurationLockState.Lockdown	**NVM Spec: Property / Field:**NVMe 2.0d: Identify Command / Identify Controller data structure (CNS 01h) - Optional Admin Command Support (OACS)**NVM Spec: Section:Figure**NVMe 2.0d: Section 5.17.2.1: Figure 275 - 257:256, Bit 10) Get log page specifying the Command and Feature Lockdown (Log Page Identifier 14h) **NVM Spec: Section:Figure**NVMe 2.0d: Section 5.16.1.20

	Redfish/Swordfish	NVMe / NVMe-oF
Type	Drive.v1_20_0.ConfigLockO ptions • `Unlocked`: Command is supported, able to be locked, and the current state is unlocked. • `Locked`: Command is supported, able to be locked, and the current state is locked. • `LockdownUnsupported`: Command is supported, able to be locked, and the current state is locked. • `CommandUnsupported`: Command is not supported, therefore lockdown does not apply.	Log Page
Descript ion	Ability to perform the NVMe defined Lockdown command.	The Lockdown command is used to control the Command and Feature Lockdown capability which configures the prohibition or allowance of execution of the specified command or Set Features command targeting a specific Feature Identifier. After a successful completion of a Lockdown command prohibiting a command or Feature Identifier, all controllers, if applicable, and all management endpoints, if applicable, in the NVM subsystem behave as requested.

	Redfish/Swordfish	NVMe / NVMe-oF
LongDesc ription	This has three states: Enabled, Disabled, and Partial. • `Enabled`: All supported properties, as reported in the `ConfigurationLockState`, that are part of the `TargetConfigurationLockLev el` are locked. • `Disabled`: All supported properties, as reported in the `ConfigurationLockState`, that are part of the `TargetConfigurationLockLev el` are unlocked.• `Partial`: The properties, as reported in the `ConfigurationLockState`, that are part of the `TargetConfigurationLockLev el` are not consistently locked or unlocked. Services shall reject modification requests that contain the value `Partial`.	Command and Feature Lockdown is used to prohibit the execution of commands submitted to NVM Express controllers and/or Management Endpoints in an NVM subsystem. Commands and Feature Identifiers are defined with the following scopes: • Admin Command Set commands defined by the Opcode field;• Set Features command features defined by the Feature Identifier field;• Management Interface Command Set commands defined by the Opcode field; and• PCIe Command Set commands defined by the Opcode field (refer to the NVM Express Management Interface Specification).
Mandator y	Optional	Mandatory to implement OACS; Lockdown feature is optional. If able to be supported (in log page 14), use the lockdown command to lock and unlock correspond to client requests.

	Redfish/Swordfish	NVMe / NVMe-oF
Notes	See section `Mapping NVMe Command and F eature Lockdown to "Configu rationLocked"` for detailed behavior and mapping description.	

6.8.2.31 NVMe.NVMeConfigurationLockState.SecuritySend The mapping for `NVMe.NVMeConfigurationLockState.SecuritySend` is summarized in Table 279.

Table 279: NVMe.NVMeConfigurationLockState.SecuritySend mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMe.NVMeConfigurationLockState.SecuritySend	**NVM Spec: Property / Field:**NVMe 2.0d: Identify Command / Identify Controller data structure (CNS 01h) - Optional Admin Command Support (OACS)**NVM Spec: Section:Figure**NVMe 2.0d: Section 5.17.2.1: Figure 275 - 257:256, Bytes 13:08) Get log page specifying the Command and Feature Lockdown (Log Page Identifier 14h) **NVM Spec: Section:Figure**NVMe 2.0d: Section 5.16.1.20 (Figure 260)

	Redfish/Swordfish	NVMe / NVMe-oF
Type	Drive.v1_20_0.ConfigLockO ptions • `Unlocked`: Command is supported, able to be locked, and the current state is unlocked. • `Locked`: Command is supported, able to be locked, and the current state is locked. • `LockdownUnsupported`: Command is supported, able to be locked, and the current state is locked. • `CommandUnsupported`: Command is not supported, therefore lockdown does not apply.	Log Page
Descript ion	Ability to perform the NVMe defined SecuritySend command.	If the SecuritySend feature is able to be locked down, it's corresponding opcode, 81h, will be reported in the command and feature lockdown log page (bytes 13:08).

	Redfish/Swordfish	NVMe / NVMe-oF
LongDesc ription	This has three states: Enabled, Disabled, and Partial. • `Enabled`: All supported properties, as reported in the `ConfigurationLockState`, that are part of the `TargetConfigurationLockLev el` are locked. • `Disabled`: All supported properties, as reported in the `ConfigurationLockState`, that are part of the `TargetConfigurationLockLev el` are unlocked.• `Partial`: The properties, as reported in the `ConfigurationLockState`, that are part of the `TargetConfigurationLockLev el` are not consistently locked or unlocked. Services shall reject modification requests that contain the value `Partial`.	Command and Feature Lockdown is used to prohibit the execution of commands submitted to NVM Express controllers and/or Management Endpoints in an NVM subsystem. Commands and Feature Identifiers are defined with the following scopes: • Admin Command Set commands defined by the Opcode field;• Set Features command features defined by the Feature Identifier field;• Management Interface Command Set commands defined by the Opcode field; and• PCIe Command Set commands defined by the Opcode field (refer to the NVM Express Management Interface Specification).
Mandator y	Optional	Mandatory to implement OACS; SecuritySend feature lockdown support is optional. If able to be supported (in log page 14), use the lockdown command to lock and unlock correspond to client requests.

	Redfish/Swordfish	NVMe / NVMe-oF
Notes	See section `Mapping NVMe Command and F eature Lockdown to "Configu rationLocked"` for detailed behavior and mapping description.	

6.8.2.32 NVMe.NVMeConfigurationLockState.VPDWrite The mapping for NVMe.NVMeConfigurationLo
is summarized in Table 280.

Table 280: NVMe.NVMeConfigurationLockState.VPDWrite mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	NVMe.NVMeConfigurationLockState.VPDWrite	**NVM Spec: Property / Field:**NVMe 2.0d: Identify Command / Identify Controller data structure (CNS 01h) - Optional Admin Command Support (OACS)**NVM Spec: Section:Figure**NVMe 2.0d: Section 5.17.2.1: Figure 275 - 257:256) Get log page specifying the Command and Feature Lockdown (Log Page Identifier 14h) **NVM Spec: Section:Figure**NVMe 2.0d: Section 5.16.1.20 (Figure 260)
Type	Drive.v1_20_0.ConfigLockOptions • Unlocked: Command is supported, able to be locked, and the current state is unlocked. • Locked: Command is supported, able to be locked, and the current state is locked. • LockdownUnsupported: Command is supported, able to be locked, and the current state is locked. • CommandUnsupported: Command is not supported, therefore lockdown does not apply.	Log Page

	Redfish/Swordfish	NVMe / NVMe-oF
Descript ion	Ability to perform the NVMe defined VPDWrite command.	If the VPDWrite feature is able to be locked down, it's corresponding opcode, 81h, will be reported in the command and feature lockdown log page (bytes 13:08).
LongDesc ription	This has three states: Enabled, Disabled, and Partial. • `Enabled`: All supported properties, as reported in the `ConfigurationLockState`, that are part of the `TargetConfigurationLockLevel` are locked. • `Disabled`: All supported properties, as reported in the `ConfigurationLockState`, that are part of the `TargetConfigurationLockLevel` are unlocked.• `Partial`: The properties, as reported in the `ConfigurationLockState`, that are part of the `TargetConfigurationLockLevel` are not consistently locked or unlocked. Services shall reject modification requests that contain the value `Partial`.	Command and Feature Lockdown is used to prohibit the execution of commands submitted to NVM Express controllers and/or Management Endpoints in an NVM subsystem. Commands and Feature Identifiers are defined with the following scopes: • Admin Command Set commands defined by the Opcode field;• Set Features command features defined by the Feature Identifier field;• Management Interface Command Set commands defined by the Opcode field; and• PCIe Command Set commands defined by the Opcode field (refer to the NVM Express Management Interface Specification).

	Redfish/Swordfish	NVMe / NVMe-oF
Mandator y	Optional	Mandatory to implement OACS; VPDWrite feature lockdown support is optional. If able to be supported (in log page 14), use the lockdown command to lock and unlock correspond to client requests.
Notes	See section `Mapping NVMe Command and F eature Lockdown to "Configu rationLocked"` for detailed behavior and mapping description.	

6.8.2.33 PhysicalLocation.Info   The mapping for `PhysicalLocation.Info` is summarized in Table 281.

Table 281: PhysicalLocation.Info mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	PhysicalLocation.Info	N/A
Type	String	N/A
Mandatory	Do Not Implement	
Notes	This property has been deprecated.	

6.8.2.34 PhysicalLocation.InfoFormat   The mapping for `PhysicalLocation.InfoFormat` is summarized in Table 282.

Table 282: PhysicalLocation.InfoFormat mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	PhysicalLocation.InfoFormat	N/A
Type	String	N/A
Mandatory	Do Not Implement	
Notes	This property has been deprecated.	

6.8.2.35 PhysicalLocation.PartLocation   The mapping for `PhysicalLocation.PartLocation` is summarized in Table 283.

Table 283: PhysicalLocation.PartLocation mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	PhysicalLocation.PartLocati on	N/A
Type	Boolean	N/A
Descript ion	An indication of whether the drive is accessible from multiple paths.	N/A
LongDesc ription	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
Mandator y	RecommendedMandatory 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.	

6.8.2.36 PredictedMediaLifetLeftPercent   The mapping for `PredictedMediaLifetLeftPercent` is summarized in Table 284.

Table 284: PredictedMediaLifetLeftPercent mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	PredictedMediaLifetLeftPerc ent	**NVM Spec Property / Field:** Get Log Page – SMART, Health Information Log (Log Identifier 02h) / Percentage Used **NVM Spec: Section:Figure** NVMe 2.0: Section 5.16.1.3: Figure 207: Byte 05
Type	Decimal	Decimal range: 0% - 255%
Descript ion	The percentage of reads and writes that are predicted to be available for the media.	Contains a vendor specific estimate of the percentage of NVM subsystem life used based on the actual usage and the manufacturer's prediction of NVM life.

	Redfish/Swordfish	NVMe / NVMe-oF
LongDesc ription	This property shall contain an indicator of the percentage of life remaining in the drive's media.	Contains a vendor specific estimate of the percentage of NVM subsystem life used based on the actual usage and the manufacturer's prediction of NVM life. A value of 100 indicates that the estimated endurance of the NVM in the NVM subsystem has been consumed, but may not indicate an NVM subsystem failure. The value is allowed to exceed 100. Percentages greater than 254 shall be represented as 255. This value shall be updated once per power-on hour (when the controller is not in a sleep state).
Mandator y	Mandatory	Optional
Notes	Maps to percentage used in SMART information log	This value shall be scaled to a value between 0% - 100%Calculate resulting value as 100 - value reported (PercentageUsed).Refer to the JEDEC JESD218A standard for SSD device life and endurance measurement techniques.

6.8.2.37 Protocol   The mapping for `Protocol` is summarized in Table 285.

Table 285: Protocol mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Protocol	N/A
Type	Protocol.Protocol	N/A
Descript ion	The protocol that this drive currently uses to communicate to the storage controller.	N/A
LongDesc ription	This property shall contain the protocol that the associated drive currently uses to communicate to the storage controller for this system.	N/A
Mandator y	Mandatory	
Notes	Possible values (long list)	NVMe Drives shall report "NVMe".

6.8.2.38 Revision   The mapping for `Revision` is summarized in Table 286.

Table 286: Revision mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Revision	**NVM Spec Property / Field:** Identify Command / Identify Controller Data Structure (CNS 01h) / Firmware Revision (FR) **NVM Spec: Section:Figure** NVMe 2.0: section 5.17.2.1: Figure 275: Bytes 64:71
Type	String	String
Descript ion	The revision of this drive. This is typically the firmware or hardware version of the drive.	Contains the currently active firmware revision, as an ASCII string, for the domain of which this controller is a part.
LongDesc ription	This property shall contain the manufacturer-defined revision for the associated drive.	Contains the currently active firmware revision, as an ASCII string, for the domain of which this controller is a part.
Mandator y	Mandatory	Mandatory
Notes		Return the currently active firmware revision information.This is the same revision information that may be retrieved with the Get Log Page command, refer to NVMe Base specification 2.0, section 5.16.1.4

6.8.2.39 RotationSpeedRPM   The mapping for `RotationSpeedRPM` is summarized in Table 287.

Table 287: RotationSpeedRPM mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	RotationSpeedRPM	**NVM Spec Property / Field:** Get Log Page - Rotational Media Information Log (Log ID 16h) / Nominal Rotational Speed (NRS)**NVM Spec: Section:Figure** NVMe 2.0: section 5.16.1.22: Figure 263: Bytes 4:5
Type	Decimal	Decimal
Descript ion	An indication of whether the drive is accessible from multiple paths.	Nominal rotational speed in revolutions per minute while the current Power State is 0
LongDesc ription	This property shall indicate whether the drive is accessible by an initiator from multiple paths allowing for failover capabilities upon a path failure.	This log page provides rotational media information (refer to section 8.20) for Endurance Groups that store data on rotational media. The information provided is retained across power cycles and resets.
Mandator y	OptionalIf MediaType == SSD, Do Not Implement.	I/O Controller: OptionalAdmin Controller: ProhibitedDiscovery Controller: Prohibited

Redfish/Swordfish	NVMe / NVMe-oF
Notes	Nominal rotational speed in revolutions per minute while the current Power State is 0Supported Values / Descriptions:0000h Not reported0001h This value shall not be used.FFFFh ReservedAll other values Nominal rotational speed in revolutions per minute while the current Power State is 0

6.8.2.40 SKU   The mapping for SKU is summarized in Table 288.

Table 288: SKU mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	SKU	N/A
Type	String	N/A
Descript ion	The SKU for this drive.	N/A
LongDesc ription	This property shall contain the stock-keeping unit (SKU) number for this drive.	N/A
Mandator y	Mandatory	
Notes	The drive should support this property to be filled in by a layered process (e.g., OEM manufacturing).	

6.8.2.41 SerialNumber   The mapping for `SerialNumber` is summarized in Table 289.

Table 289: SerialNumber mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	SerialNumber	**NVM Spec Property / Field:** Identify Command / Identify Controller Data Structure (CNS 01h): Serial Number (SN) **NVM Spec: Section:Figure** NVMe 2.0: section 5.17.2.1: Figure 275: Bytes 04:23
Type	String	String
Descript ion	The serial number for this drive.	Contains the serial number for the NVM subsystem that is assigned by the vendor as an ASCII string.
LongDesc ription	This property shall contain the manufacturer-allocated number that identifies the drive.	Contains the serial number for the NVM subsystem that is assigned by the vendor as an ASCII string.
Mandator y	Mandatory	I/O Controller: MandatoryAdmin Controller: MandatoryDiscovery Controller: Restricted
Notes		Each identifier is in big endian format.Refer to NVMe 2.0 Base specification section 4.5.1 for unique identifier requirements. Refer to NVMe 2.0 Base specification section 1.4.2 for ASCII string requirements

6.8.2.42 Status.State   The mapping for `Status.state` is summarized in Table 290.

Table 290: Status.State mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Status.State	**NVM Spec Property / Field:** Property Get Command / CSTS (Offset 1Ch) – Controller Status: Ready (RDY) - **NVM Spec: Section:Figure** NVMe 2.0: section 3.1.3.6: Figure 47: Bit 00
Type	Resource.State (enum)	Enum
Descript ion	The known state of the resource, such as, enabled.	This bit is set to "1" when the controller is ready to process submission queue entries after CC.EN is set to "1". This bit shall be cleared to "0" when CC.EN is cleared to "0" once the controller is ready to be re-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.	This bit is set to "1" when the controller is ready to process submission queue entries after CC.EN is set to "1". This bit shall be cleared to "0" when CC.EN is cleared to "0" once the controller is ready to be re-enabled
Mandatory	OptionalMandatory for NVM Drives	Mandatory
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".

6.8.2.43 Status.Health   The mapping for `Status.Health` is summarized in Table 291.

Table 291: Status.Health mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	Status.Health	**NVM Spec Property / Field:** Get Log Page – SMART / Health Information Log (Log ID 02h) / Critical Warning**NVM Spec: Section:Figure** NVMe 2.0: Section 5.16.1.3, Figure 207: Byte 00
Type	Resource.Health	Byte field
Descript ion	The health state of this resource in the absence of its dependent resources.	This field indicates critical warnings for the state of the controller. Each bit corresponds to a critical warning type
LongDesc ription	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.	This field indicates critical warnings for the state of the controller. Each bit corresponds to a critical warning type; multiple bits may be set to "1". If a bit is cleared to "0", then that critical warning does not apply. Critical warnings may result in an asynchronous event notification to the host. Bits in this field represent the current associated state and are not persistent.
Mandator y	OptionalMandatory for NVM Drives.	I/O Controller: MandatoryAdmin Controller: OptionalDiscovery Controller: Prohibited

	Redfish/Swordfish	NVMe / NVMe-oF
Notes	Possible Values: OK / Warning / Critical	Report to same value as set for worst-case controller Status.Health. Bits / Definitions7:6 Reserved5 If set to "1", then the Persistent Memory Region has become read-only or unreliable.4 If set to "1", then the volatile memory backup device has failed. This field is only valid if the controller has a volatile memory backup solution.3 If set to "1", then all of the media has been placed in read only mode. The controller shall not set this bit to "1" if the read-only condition on the media is a result of a change in the write protection state of a namespace.2 If set to "1", then the NVM subsystem reliability has been degraded due to significant media related errors or any internal error that degrades NVM subsystem reliability.1 If set to "1", then a temperature is: a) greater than or equal to an over temperature threshold; or b) less than or equal to an under temperature threshold.0 If set to "1", then the available spare capacity has fallen below the threshold.

6.8.2.44 StatusIndicator   The mapping for `StatusIndicator` is summarized in Table 292.

Table 292: StatusIndicator mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	StatusIndicator	N/A
Type	enum (StatusIndicator)	N/A
Descript ion	An indication of whether the drive is accessible from multiple paths.	N/A
LongDesc ription	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
Mandator y	Mandatory	
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).

6.8.2.45 TargetConfigurationLockLevel   The mapping for `TargetConfigurationLockLevel` is summarized in Table 293.

Table 293: TargetConfigurationLockLevel mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	TargetConfigurationLockLeve l	N/A
Type	enum (TargetConfigurationLockLev el)	N/A
Descript ion	Indicates the target configuration lock level for the drive resource. This corresponds to the `ConfigurationLock` and `ConfigurationLockState` for the storage subsystem type.	N/A
LongDesc ription	This property shall indicate the target configuration lock level for the drive resource. This corresponds to the `ConfigurationLock` and `ConfigurationLockState` for the storage subsystem type.	N/A
Mandator y	Optional; Use when the subsystem contains endurance groups and sets, represented by StoragePool resources.	

	Redfish/Swordfish	NVMe / NVMe-oF
Notes	The `Standard` enum is defined as the standard configuration lock level, corresponding to applying firmware, and updating security keys. See section `Mapping NVMe Command and F eature Lockdown to "Configu rationLocked"` for detailed behavior and mapping description.	

6.8.2.46 WriteCacheEnabled   The mapping for `WriteCacheEnabled` is summarized in Table 294.

Table 294: WriteCacheEnabled mapping

	Redfish/Swordfish	NVMe / NVMe-oF
Property	SerialNumber	**NVM Spec Property / Field:** Get Features / Volatile Write Cache (Feature Identifier 06h) / Volatile Write Cache Enable (WCE)**NVM Spec: Section: Figure** NVMe 2.0: Section 5.27.1.4: Figure 321: Bit 00
Type	Boolean	Boolean
Descript ion	An indication of whether the drive write cache is enabled.	This property shall indicate whether the drive write cache is enabled.
LongDesc ription	This property shall indicate whether the drive write cache is enabled.	This property shall indicate whether the drive write cache is enabled.
Mandator y	OptionalMandatory for NVMe Drives.	Optional
Notes		If set to "1", then the volatile write cache is enabled. If cleared to "0", then the volatile write cache is 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 295 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 295: 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 MultipartHttp-PushUri 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.

7.3 Security Management Protocols

NVMe specifies an optional channel for security protocol communication with controller devices, implemented as a pair of send and receive functions: Security Send (described in section 5.26 of NVMe) and Security Receive (section 5.25). These provide functionality equivalent to the SPC-5 `SECURITY PROTOCOL IN` and `SECURITY PROTOCOL OUT` commands.

Swordfish maps these Security Send and Security Receive commands to two StorageController actions:

- `#StorageController.SecuritySend`
- `#StorageController.SecurityReceive`

Swordfish restricts the allowable security protocol numbers to 0, 1 or 2 for the Security Receive action, and 1 or 2 for the Security Send action.

Where the security protocol data is passed as a base64-encoded string, opaque to the Swordfish/Redfish implementation.

7.4 Reporting Redfish/Swordfish Service URI to NVMe Subsystem

The NVMe specification provides a mechanism to set the address of the management agent for an NVMe Subsystem, which may then be set and retrieved using the Embedded Management Controller Address feature. Other management consumers may need to have a pointer to the Redfish/Swordfish service responsible for managing an NVMe Subsystem accessible through alternate management interfaces, such as via NVMe-MI.

The address of a management agent contained in a fabric interface manager is indicated in the NVMe 2.0d Base Specification Management Address List log page, in a Management Address Descriptor indicating a Management Address Type of 2h.

7.4.1 Implementation Guidance

The Swordfish service shall send the fully qualified system and redfish root URI as a string to the NVM device, https://<Systemname>:<port>/redfish/v1, where <systemname> and <port> correspond to the Swordfish service.

<Systemname> may be filled in with either FQDN or IP address, depending on the implementation. Implementations should note network constraints such as certificates and DNS support or registrations when determining how to populate this field.

7.4.2 Background

Various entities on a network are able to request a management agent to perform management operations on NVM subsystems. A controller in an NVM subsystem uses the Management Addresses capability to indicate the network addresses of those management agents. When an NVM subsystem is provisioned in a storage system, the management addresses are established in the controller.

Management agents are able to be located in various networked entities, including:

- NVM subsystems;
- Fabric interface managers;
- Embedded management controllers; and
- Host software.

Each management address is represented as a uniform resource indicator as defined by
RFC 3986.

7.5 Mapping NVMe Command and Feature Lockdown to "ConfigurationLocked"

7.5.1 NVMe Lockdown Command

In NVMe 2.0d, the Lockdown was added. This provides a mechanism for granular control
over execution of NVMe commands or Feature Identifiers.

There are many potential NVMe commands and Feature Identifiers that may be exposed,
including vendor specific commands. In addition, the entire Lockdown command struc-
ture is optional within the NVMe command set.

From a Redfish/Swordfish client perspective, this can be confusing, and contains much
more information than needed. In addition, locking the entire set of commands could
leave some systems in an unusable state. Current client use cases are only interested in
locking a subset of the commands. To address this, a set of properties has been added
to Redfish and Swordfish to provide reporting and control.

7.5.2 ConfigurationLock, TargetConfigurationLockLevel, and ConfigurationLockState

The ConfigurationLock, TargetConfigurationLockLevel, and ConfigurationLockState
properties are jointly used to manage lockdown of NVMe devices. These properties
are enabled on the NVMe Subsystem, and may be mirrored to the Drive object (see
"Implementing ConfigurationLock in Storage and Drive Resources" below).

The following example shows the properties in use for the NVMe Subsystem:

```
"ConfigurationLock": "Partial",
"TargetConfigurationLockLevel": "Standard",

"NVMeSubsystemProperties": {
    "ConfigurationLockState": {
    "FirmwareCommit": "Unlocked",
```

```
    "Lockdown": "Locked",
    "SecureSend": "LockdownUnsupported",
    "FirmwareImageDownload": "LockdownUnsupported",
    "VPDWrite": "CommandUnsupported"
    }
}
```

7.5.2.1 ConfigurationLock The `ConfigurationLock`, `TargetConfigurationLockLevel`, and `ConfigurationLockState` properties are jointly used to manage lockdown of NVMe devices. These properties are enabled on the NVMe Subsystem, and may be mirrored to the Drive object (see further information below).

The primary command to see the overall state of lockdown is `ConfigurationLock`. This has three states: Enabled, Disabled, and Partial.

- Enabled: All supported properties, as reported in the `ConfigurationLockState`, that are part of the `TargetConfigurationLockLevel` are locked.
- Disabled: All supported properties, as reported in the `ConfigurationLockState`, that are part of the `TargetConfigurationLockLevel` are unlocked.
- Partial: The properties, as reported in the `ConfigurationLockState`, that are part of the `TargetConfigurationLockLevel` are not consistently locked or unlocked.

If lockdown is not supported on the subsystem, ConfigurationLock shall not be implemented.

If any or all of the properties or commands that are part of the `TargetConfigurationLockLevel` are not supported on the subsystem, the service shall ignore the unsupported features when determining whether the lock states match the client requested target state as part of setting `ConfigurationLock` to either `Enabled` or `Disabled`.

The `Partial` setting indicates that the device is in a state that does not map to either `Enabled` or `Disabled`. This may mean that the device has been configured from another interface, or it may indicate an error condition. `Partial` is to be rejected if specified in PATCH requests.

On initial configuration, if a user sees a `Partial` value in a response, it's an indicator they should PATCH the property to Enabled or Disabled to put the drive in a known state; however this may result in a failure if the Lockdown itself is one of the locked commands,

such as when the Lockdown NVMe command has been Enabled from the management interface to the drive.

> Note: a standard event will be added to the Redfish/Swordfish events in a future release for this scenario.

7.5.2.2 TargetConfigurationLockLevel As previously noted, the specific set of commands desired to be locked may vary. Groups of these are mapped to the `TargetConfigurationLockLevel` property for convenience.

The `TargetConfigurationLockLevel` property is a list of types that each map to a specific set of NVMe command and feature identifiers.

The `TargetConfigurationLockLevel` of `Standard` maps to the following commands and feature identifiers:

- FirmwareCommit
- FirmwareImageDownload
- Lockdown
- SecuritySend
- VPDWrite

Other enums with diffent mappings will be added over time.

If the service receives a request to PATCH `TargetConfigurationLockLevel` when `ConfigurationLock` is not `Disabled`, the service shall attempt to apply the new settings. Any appropriate event notifications should be sent, such as ResourceEvents indicating changes to the `TargetConfigurationLockLevel` property.

If lockdown is not supported on the subsystem, TargetConfigurationLockLevel shall not be implemented.

7.5.2.3 ConfigurationLockState The device may not support all of the requested properties in the selected level, particularly if the selected level has many properties, which is why the `ConfigurationLockState` property is required to also be implemented. The `ConfigurationLockState` property shows what command and feature identifiers the device has actually implemented, and their current state. It contains properties that correspond to NVMe command and feature identifiers.

The current state of each property may be reported as one of the following four states:

- Unlocked - Command is supported, able to be locked, and the current state is unlocked.
- Locked - Command is supported, able to be locked, and the current state is locked.
- LockdownUnsupported - Command is supported, able to be locked, and the current state is locked.
- CommandUnsupported - Command is not supported, therefore lockdown does not apply.

Example:

```
"ConfigurationLockState":    {
    "FirmwareCommit": "Unlocked",
    "Lockdown": "Locked",
    "SecuritySend": "LockdownUnsupported",
    "FirmwareImageDownload": "LockdownUnsupported",
    "VPDWriteMI": "CommandUnsupported"
}
```

7.5.3 Lockdown State Management

On the NVMe device, lockdown is not guaranteed to be persistent across resets. Therefore, the service shall routinely update its view of the drive property state relevant to the configuration lock. If changes to the configuration are detected and the service has received an explicit request from a client to set the configuration lock to Enabled or Disabled, the service:

- 1 - shall reapply the client requested configuration lock state
- 2 - should log the activity
- 3 - should send an event indicating the configuration lock was set to the desired state. Note: a standard event will be added to the Redfish/Swordfish events in a future release for this scenario.

7.5.3.1 Expected Behavior from ConfigurationLock   Setting `ConfigurationLock` to `Enabled` may prevent the Redfish/Swordfish service itself from sending the specific commands. For example, if the Swordfish service is running in-band, a request for SecuritySend would fail when the ConfigurationLock is set. In this scenario, the

Redfish/Swordfish service should return the PropertyValueResourceConflict message (Base message registry) to report that a request cannot be completed due to the ConfigurationLock being set. This message would be a more "user friendly" indicator of the failure.

```
"PropertyValueResourceConflict": {
    "Description": "Indicates that the requested write of a
      ↪   property value could not be completed due to the
      ↪   current state or configuration of another
      ↪   resource.",
    "LongDescription": "This message shall indicate that
      ↪   the requested write of a property could not be
      ↪   completed because the requested value of the
      ↪   property, which is a value supported by the
      ↪   implementation, cannot be accepted due to
      ↪   conflicts in the state or configuration of another
      ↪   resource.",
    "Message": "The property '%1' with the requested value
      ↪   of '%2' could not be written because the value
      ↪   conflicts with the state or configuration of the
      ↪   resource at '%3'.",
    "Severity": "Warning",
    "MessageSeverity": "Warning",
    "VersionAdded": "1.10.0",
    "NumberOfArgs": 3,
    "ParamTypes": [
        "string",
        "string",
        "string"
    ],
    "ArgDescriptions": [
        "The name of the property for which a write was
          ↪   requested.",
        "The value of the property that is in conflict.",
        "The URI of the resource in conflict."
    ],
```

```
        "ArgLongDescriptions": [
            "This argument shall contain the name of the
            ↪   property for which a write was requested.",
            "This argument shall contain the value of the
            ↪   property that is in conflict.",
            "This argument shall contain the URI of the
            ↪   resource that is in conflict with the value of
            ↪   the requested property."
        ],
        "Resolution": "None."
    },
```

7.5.4 Implementing ConfigurationLock in Storage and Drive Resources

The ConfigurationLock property of the Subsystem resource (the `Storage` object) shall be implemented as the primary client point of reference for the functionality. Clients should refer to Subsystem, as some types of NVMe devices may not have Drive objects.

For NVMe drive implementations, ConfigurationLock may also be implemented in the `Drive` resource. In this configuration, the property value shall reflect the value for the NVMe device's Subsystem. The `Drive` resource contains `ConfigurationLock`, `TargetConfigurationLevel`, and `NVMe.ConfigurationLockState` properties to mirror those in `Storage`.

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

Table A.1: Related Objects

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

Object	NVMe Device Expected Usage
NetworkProtocol	EBOF/JBOF, Complex devices, Arrays

Required usage of objects and their properties are contained in the relevant Swordfish profiles.

For example, for Ethernet-Attach drives, the NetworkAdapter, Port, and NetworkDevice-Function 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 21 May 2024.

Table A.2: NVMe Use Case Summary

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

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.

Table B.1: Informational References

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