Swordfish Scalable Storage Management API Specification

Version: 1.2.7

Abstract: The Swordfish Scalable Storage Management API defines a RESTful interface and a standardized data model to provide a scalable, customer-centric interface for managing storage and related data services.

SNIA Standard

Last Updated: 13 August 2024

Contents

List of Tables

List of Figures

USAGE

DISCLAIMER

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

The evolution of this document is summarized in Table 1.

Table 1: Revision history

Date	Rev	Notes
19September2016	1.0.0	Initial Release
12 October 2016	1.0.1	Errata release for general clean up and formatting consistency
1 November 2016	1.0.2	Errata release to change multiple collections' types from collections (arrays) to ResourceCollections to conform to Redfish usage guidelines
		Change multiple collections' types from collections (arrays) to ResourceCollections to conform to Redfish usage guidelines and move NavigationProperties from Links section.
24 January 2017	1.0.3	Errata release to move complex types and enum to versioned namespace
		Schedule schema: add property
		json schema fix (Swordfish to swordfish)

Date	Rev	Notes
		Specification enhancements, multiple areas
		User's guide: multiple new use cases and new document section
25 April 2017	1.0.4	Errata release with minor updates to schema: move FileShare collection, integrate DMTF and SNIA versions of Volume, fix incorrect property references and update descriptions. Update mockups. User's guide: Update cross-references.
3 October 2017	1.0.5	Errata release to include schema simplifications and other lessons from initial implementations, as well as general cleanup of specification.
13 February 2018	1.0.6	Updated Storage Systems model – added notion of Integrated Service Configuration in addition to (and named) Hosted Service Configuration
		Added ComplexType common definition section
		Added/updated common Redfish property definitions
		Updates to conform to new SNIA templates.
12 October 2018	1.0.7	Enhanced Spare Capacity Management Model; Deprecated Remaining Capacity
		Added OpenAPI support: schema references and OpenAPI YAML files
		Added iSCSI properties for CHAP
		Event usage enhancements and guidance
		Volume schema updates – RAID Type enum (deprecating VolumeType usage), add ReplicaTargets

Date	Rev	Notes
		Schema updates: Annotations enhancements: Capabilities designations, owning entities, Redfish.Required usage
		Clarified and updated ClassOfService IsDefault property usage
		Updated Capabilities location in hierarchy
		Fix cardinality issue of StorageReplicaInfo usage in StorageGroups and Volume
		Consolidate Client and Server Endpoint Groups into single Endpoint Group entity (deprecate usage of separate Client Endpoint Group and Server Endpoint Group)
		Add MappedVolume construct to StorageGroup – adds LUN info and other properties
		Clarified and updated ClassOfService IsDefault property usage
		Updated Capabilities location in hierarchy
		Fix cardinality issue of StorageReplicaInfo usage in StorageGroups and Volume
		Consolidate Client and Server Endpoint Groups into single Endpoint Group entity (deprecate usage of separate Client Endpoint Group and Server Endpoint Group)
		Add MappedVolume construct to StorageGroup – adds LUN info and other properties
8 November 2018	1 .0.7a	Restored RAIDType property that was missing from 1.0.7
		Minor correction to schema versioning
22 August 2019	1.1.0	Restructured to add features and profiles

Date	Rev	Notes
		Add description of SupportedFeatures usage and requirements
		Add requirements for subsets of Add language to clarify support for use with and without the class of service (now an optional feature)
		Added descriptions of support for seamless extension of Redfish Storage model to Swordfish
		Add updated model diagrams to reflect new model permutations
		Added descriptions of new constructs (e.g., Consistency Groups)
		Cleaned up references to Redfish Specification based on latest version
		Add Status Codes clarification and constraints section
12 November 2019	1.1.0	Released as Technical Position
12 November 2019	1 .1.0a	Released as Corrected Technical Position
		Formatting fixes – word wrap in pdf doc format to fix truncated lines
		Consistent object labeling in images (replace drive with disk)
		Editorial and grammar changes and cleanup to status code guidance section
24 March 2020	1 .1.0b	Released as Corrected Technical Position
		TLS requirements now based on both ISO and SNIA standards
		Redfish references now based on both ISO and SNIA standards
		Bibliography added

Date	Rev	Notes
29 May 2020	1.2.0	Note: This release is done in conjunction with the DMTF's Redfish Forum Work-in-Progress June 2020 release of DSP-IS0014 (v0.95), which contains multiple schema to support this work. Both are released as Working Drafts / work-in-progress for public review, and plan simultaneous releases in early fall 2020 to support full technical specification level capability and availability.
		Functionality availability in Swordfish includes:
		• Enhancements to Volume, StoragePools
		• New schema: NVMeDomain
		Other supporting documentation released in conjunction with this specification and schema bundle:
		• Multiple mockups reflecting multiple implementation permutation options (available on swordfishmockups.com)
		• Model overview documentation (NVMe to RF/SF Model Mapping Working Draft, dated May 2020)
18 August 2020	1.2.1	Note: This release is done in conjunction with the DMTF's Redfish Forum 2020.3 Release of the Redfish Specification, schema bundle and other supporting materials.
		Functionality availability in Swordfish includes:
		• NVMe Mapping Support, Enhancements to Volume, StoragePools
		Additional Enhancements in the Specification and schema:
		• Added InitializeMethod property to Volume.

Date	Rev	Notes
		• Made DedicateSpareDrives ReadWrite-able
		• Added enhanced Volume Access Capabilities and usage in StorageGroup.
		• Fix multiple URI issues across various schema.
		Updated formatting of tables to support automatic table numbering and ISO compatible table representation.
29 September 2020	1 .2.1a	Added bibliography and updated TLS references
20 October 2020	1 .2.1c	Updated with additional Redfish.URI annotations.
31 October 2020	1 .2.1c	Released as SNIA Standard
2 March 2021	1.2.2	Added sections to document use of complex types.
		Updated common properties sections.
		Schema changes:
		Add actions to Add and Remove drives directly from StoragePool.
		Split NVMeFirmwareImage and NVMeDomains schemas.
		Deprecate use of NetworkPort; replace with Port.
		Update Redfish.URI references.
		Corrected $ref references in JSON schema files.
		Fix incorrect references in deprecated JSON files.
30 August 2021	1.2.3	Adds updates / corrections to Redfish.URI annotations
		Add IsBootCapable to Volume

Date	Rev	Notes
		Add SupportedPoolTypes to StoragePool
5 December 2021	1.2.3	Release as SNIA Standard
12 April 2022	1.2.4	Release as Working Draft. Schema changes:
		• FeaturesRegistry: Errata fix – make Features property a collection.
		• IOStatistics: clarify intent regarding reset / wrap.
		• StoragePool: errata fixes for Actions.
		• Volume: errata fixes for Actions. Add: LBAFormatsSupported property to NVMeNamespaceProperties.
12 July 2022	1 .2.4a	Release as SNIA Standard.
		Includes Errata fixes to multiple profiles.
20 June 2023	1 .2.5a	Release as SNIA Standard.
		• Corrected missing section import.
		• Fix typos in schema.
22 January 2024	1.2.6	Release as Working Draft. Schema changes:
		• Copyrights updated to 2024.
		• Updates SNIA Front Matter.
		• Adds new Metrics for FileSystem, StoragePool, StorageService, and enhancements to VolumeMetrics. IOStatistics property moved to Metrics, and deprecated in FileSystem, StoragePool, StoragePool, StorageService (and Volume).
		• Fully adds new mapping and masking models using Connections in RF Fabrics, deprecates StorageGroups URIs.
		• Adds support for new volume properties: ProvidingStoragePool, ChangeStripSize, ALUA to manage reservations.

Date	Rev	Notes
		• NVMe: Adds NVMe Domain Management, including ALUA support with Domain Contents and Firmware Images.
		• Updates and simplifies LBAFormat representation, and adds multiple namespace management.
		• Deprecates NVMeFirmwareImage schema.
9 April 2024	1.2.6	Release as SNIA Standard
21 May 2024	1.2.7	Release as Working Draft
		- Move `IOStatistics` to unversioned namespace. This is used in:
		`FileSystemMetrics`
		`FileSystem`
		`StoragePoolMetrics`
		`StoragePool`
		`StorageServiceMetrics`
		`VolumeMetrics`
		`Volume`
		- Add guidance for Operations usage
		- Add guidance for volume state and status during long-running operations
		- Expansion of $metadata description
13 August 2024	1.2.7	Release as SNIA Standard
		- Errata change in FeaturesRegistry: set Resources to type Resource.Resource
		- Incorporates inherited RF schema from 2042.2

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

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

Table 2: Contributors

Member	Representatives | (* – prior employer)
Broadcom Inc.	Richelle Ahlvers(*)
Celestica	Krishnakumar Gowravaram
Cisco Systems, Inc.	Krishnakumar Gowravaram (*)|
Dell Inc.	Patrick Boyd
	George Ericson
	Jim Pendergraft
	Sean McGinnis
	Michael Raineri
	Rich Roscoe
Futurewei Inc.	Sean McGinnis(*)
Hitachi Data Systems	Eric Hibbard
Hewlett Packard Enterprise	Jeff Hilland
	Chris Lionetti
	John Mendonca

	Doug Voigt
Inova Development Inc.	Karl Schopmeyer
Intel Corporation	Richelle Ahlvers
	Rajalaxmi Angadi
	Phil Cayton
	Klaudia Jablonska
	Mariusz Krzywienski
	Mateusz Mania
	Slawek Putyrski
	Paul von Behren
Kioxia	James Borden
Microsemi Corporation	Anand Nagarjan
Microsoft Corporation	Hector Linares
	Jim Pinkerton
	Michael Pizzo
	Scott Seligman
NetApp, Inc.	Don Deel
	Nilesh Maheshwari
ScienceLogic	Patrick Strick
VMware, Inc.	Murali Rajagopal

1 Abstract

The Swordfish Scalable Storage Management API ("Swordfish") defines a RESTful interface and a standardized data model to provide a scalable, customer-centric interface for managing storage and related data services. It extends the Redfish Scalable Platforms Management API Specification from the DMTF.

2 Scope

This document defines the Swordfish Scalable Storage Management API.

Swordfish is designed as an extension of the Redfish specification. This document is written with the presumption that the reader has a detailed understanding of the Redfish specification. This document cannot be fully understood without that context.

Swordfish extends the Redfish Scalable Platforms Management API Specification to define a comprehensive, RESTful API for storage management that addresses block storage, file systems, object storage, and storage network infrastructure. It is centered around common operational and business concerns of storage management, including:

- Configuration and provisioning
- Monitoring
- Event and log management
- Performance assessment
- Diagnostics
- Fault detection and remediation
- Security
- Accounting and resource consumption

The Redfish specification provides the protocols and a core set of data models and behaviors for the management of systems. It defines the elements and behaviors that are mandatory for all Redfish implementations. Additionally it defines additional elements and behaviors that can be chosen by system vendors or manufacturers. The specifications also defines points at which OEM (system vendor) extensions can be provided by a given implementation. The specifications specifies normative requirements for Redfish Services and associated materials, such as Redfish Schema files. The Redfish specifications does not set requirements for Redfish clients, but will indicate what a Redfish client should do in order to access and utilize a Redfish Service successfully and effectively.

A Swordfish implementation conforms to all requirements specified in the Redfish specifications.

The Swordfish specification defines additional data models and behaviors for the management of storage systems and storage infrastructure.

Swordfish is suitable for a wide range of storage, from small-scale object drives, integrated RAID cards or RBODs providing storage services, to external disk arrays or

file servers, to infrastructure providing storage services for converged, hyperscale and large scale cloud environments. It includes support for multiple protocol types, including NVMe and NVMe-oF. Supporting documentation is provided to detail the mapping of Swordfish objects and properties to NVMe and NVMe-oF standard interfaces.

In addition to its storage model, Swordfish includes emerging functionality, such as clearly-defined classes of service, which provide a consistent means to map high-level business goals and objectives to specific, storage-based actions and requirements, and can be applied uniformly across a broad spectrum of storage configurations and storage types (e.g., block storage, file systems, object stores). Common storage management functionality covered by a class of service interface includes snapshots, replication, mapping and masking, and provisioning.

3 Normative References

3.1 Overview

The documents referenced 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

The approved references that contribute to this document are summarized in Table 3.

Table 3: Approved normative references

Tag	Title (Version)	Author	URL
ISO - 8601	Data elements and interchange formats – Information interchange – Representation of dates and times – Part 1: Basic rules	ISO / IEC	< http://www.iso.org/iso/home/store/catalogue_ics/catalogue_detail_ics.htm?csnumber=70907>
ISO - Direct	ISO / IEC Directives, Part 2: Principles and rules for the structure and drafting of ISO and IEC documents	ISO / IEC	<https://www.iso.org/sites/directives/current/part2/index.xhtml>
Redfish	Redfish Scalable Platforms Management API Specification (v1.18.0)	DMTF	<https://www.dmtf.org/sites/default/files/standards/documents/DSP0266_1.18.0.pdf>

Tag	Title (Version)	Author	URL
OData	Open Data Protocol (v. 4.01)	OASIS	<http://docs.oasis-open.org/odata/odata/v4.01/odata-v4.01-part1-protocol.html>
RFC3986	Uniform Resource Identifier (URI): Generic Syntax (2005)	The Internet Society	<http://www.rfc-base.org/txt/rfc-3986.txt>
CSDL	Common Schema Definition Language (4.01)	OASIS	<https://docs.oasis-open.org/odata/odata/v4.01/odata-v4.01-part3-csdl.pdf>
ITIL	ITIL Glossary (2011)	ITIL	<https://www.alaska.edu/files/oit/ITIL_2011_English_glossary_v1.0.pdf>
Units	The Unified Code for Units of Measure (v2.0.1)	Regenstrief Institute, Inc. and the UCUM Organization	<http://unitsofmeasure.org/trac>
I SO-20648	Information technology — TLS specification for storage systems	ISO/IEC	<https://www.iso.org/standard/68622.html>
SPC-4	SCSI Primary Commands - 4 (SPC-4) INCITS 513-2015	T10	<http://www.techstreet.com/cgi-bin/joint.cgi/incits>
Features	Swordfish Features Registry, version 1.5.1	SNIA	<https://redfish.dmtf.org/registries/swordfish/v1/SwordfishFeatureRegistry.1.6.0.json>

Tag	Title (Version)	Author	URL
Messages	Swordfish Message Registry, version 1.0.2	SNIA	<http s://redfish.dmtf.org/r egistries/swordfish/v1 /Swordfish.1.0.2.json>
En ergyStar	ENERGY STAR Data Center Storage Version 2.1 Final Specification – January 19, 2022	EPA	<h ttps://www.energystar. gov/sites/default/file s/ENERGY%20STAR%20Data %20Cen- ter%20Storage%20 Version%202.1%20Final% 20Specification_0.pdf>
NVMe Command	NVM Command Set Specification	NVM Express	<h ttps://nvmexpress.org/ developers/nvme- comman d-set-specifications/>

3.3 References under development

None defined in this document.

3.4 Other references

None defined in this document.

4 Terms and Definitions

4.1 Overview

In this document, some terms have a specific meaning beyond the normal English meaning. Those terms are defined in this clause. New terms, frequently used Redfish terms.

4.2 Swordfish-specific Terms

4.2.1 Definitions

The terms listed in Table 4 are used in this document.

Table 4: Swordfish terms

Term	Definition
Model	A set of resources and the relationships between them that define their semantics, behavior, and state.
Resource	A central element in a model, which represents a physical construct or a logical service, and is further defined by other model elements.
Schema	A textual representation of the resources, relationships, and interactions within a model.
Swordfish service	An extension to the Redfish Service that conforms to the Swordfish specification, and provides RESTful storage management functionality.

4.2.2 Symbols and abbreviated terms

None in this document.

4.3 Reference to Redfish terms

Many terms in this document were originally defined in the Redfish Specification. Some of the more common terms and definitions are reproduced in Table 5, as an aid to the reader.

Table 5: Redfish terms

Term	Definition (as of 16 August 2019)
OData	The Open Data Protocol, as defined in OData-Protocol.
OData Service Document	Resource that provides information about the service root for generic OData clients.
Redfish schema	A set of human and machine-readable documents that define Redfish resources using one or more of the supported schema definition languages (OData Common Schema Definition Language - CSDL, JSON, OpenAPI).
Redfish service	Implementation of the protocols, resources, and functions that deliver the interface that this specification defines and its associated behaviors for one or more managed systems.
Request	A message from a client to a service.
Service Root	Starting-point resource for locating and accessing the other resources and associated metadata that make up an instance of a Redfish service.

4.4 Keywords (normative language terms)

This document conforms to ISO/IEC Directives, Part 2 for keyword usage. The most common terms and their intended meanings are summarized Table 6.

Table 6: Normative language terms

Term(s)	Meaning
shall / shall not	Used to identify objectively verifiable criteria to be fulfilled and from which no deviation is permitted if compliance with the document is to be claimed
should / should not	Used to identify a suggested possible choice or course of action deemed to be particularly suitable without necessarily mentioning or excluding others

Term(s)	Meaning
may / need not	Used to convey consent or liberty (or opportunity) to do something
can / cannot	Expected or conceivable material, physical or causal outcome
must	Identifies a constraint or obligation on the user of the document, typically due to one or more legal requirements or laws of nature, that is not stated as a provision of the standard *NB:* "must" is not an alternative for "shall", and should only be used for constraints that arise from outside this standard

5 Swordfish Overview

5.1 Introduction

The Swordfish Scalable Storage Management API ("Swordfish") defines a RESTful interface and a standardized data model to provide a scalable, customer-centric interface for managing storage and related data services. It extends the Redfish Scalable Platforms Management API Specification from the DMTF.

5.2 Relation to Redfish
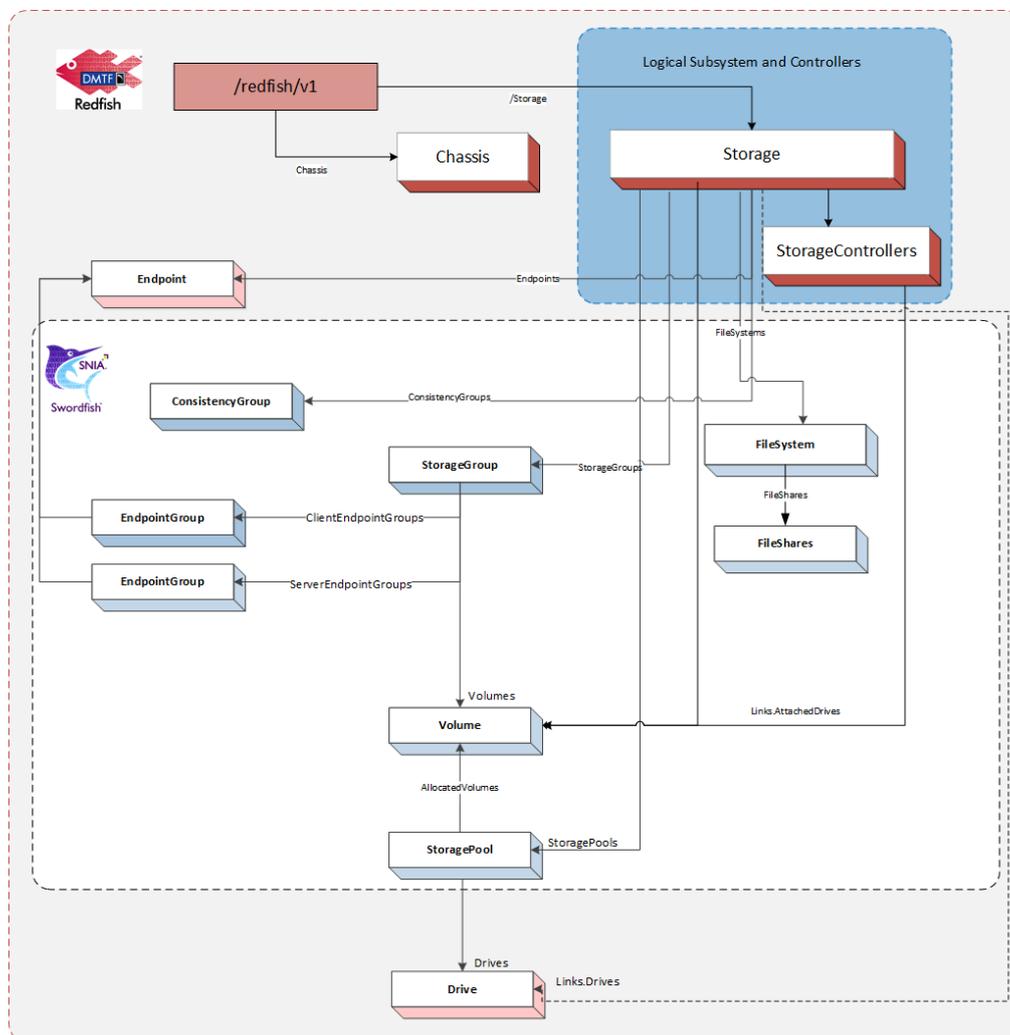
Figure 1: Model Overview

The Swordfish service interface extends the Redfish service interface. As such, a Swordfish service is a Redfish service and includes all required elements of the Redfish model, as illustrated by Figure 1.

The storage systems shall be instantiated in one of two places in the hierarchy:

- directly in the `Storage` resource collection, or
- attached to a `ComputerSystems`, with an associated reference link in the `StorageSystems` resources collection at the `ServiceRoot`. In this case, there shall also be a reference link to the `Storage` resource in the `Storage` resource collection at the `ServiceRoot`.

As a result, a Swordfish client is always able to locate the storage systems managed by the Swordfish service in the `ServiceRoot` via the `Storage` resource collection; this may be a combination of references to instances and instantiated instances.

The physical infrastructure is modeled using Redfish `Chassis`.

As modeling for storage systems may cover both logical and physical constructs, Swordfish management clients that are focused on logical storage management use cases may choose to manage functionality entirely by way of logical resources.

Each Swordfish service is accessed via well known URLs on the system supporting the Swordfish Service. Since Swordfish is an extension of Redfish, these URLs are the same as for accessing the Redfish defined aspects of the service.

5.3 Storage System Models

Swordfish has been designed to support a broad range of configurations, requirements, size and complexity, as well as logical and physical architectures. As a result, there are two primary methods of modelling the storage system for a Swordfish implementation:

1. Swordfish Standalone Configuration

The standalone configuration instantiates the logical storage system instance representation in the `Storage` resource collection directly off the Service Root. The logical storage system is modeled using the Redfish `Storage` and `StorageController` resources, as shown in Figure 2. Managed resources are connected to the `Storage` resource, including `Volumes` and `StoragePools`.

Figure 2: Logical Subsystem in Swordfish Standalone Configuration

This configuration works well for standalone devices or storage systems. An example of a Storage System for an standalone configuration is shown in Figure 3.
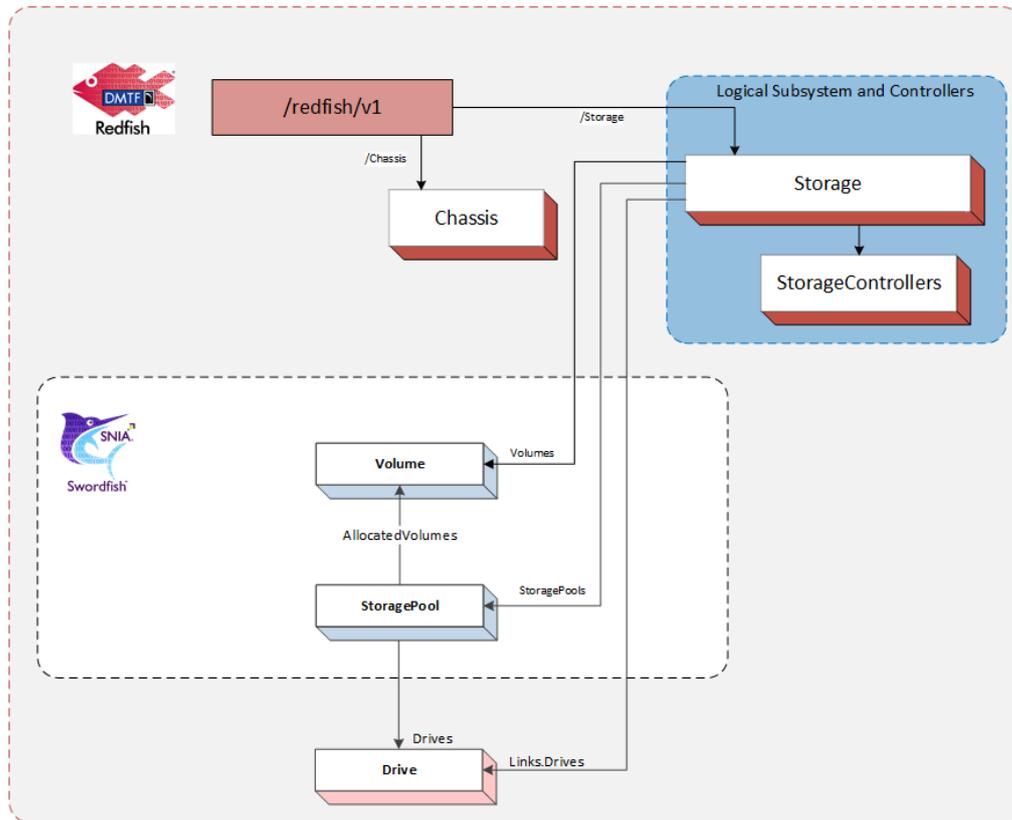
Figure 3: Swordfish Standalone Configuration Example

2. Swordfish Integrated Configuration

The integrated configuration attaches to the `Storage` collection within the same ComputerSystem model instantiation as the server where the physical element resides.

The logical storage system is modeled using the Redfish `Storage` and `Storage-Controller` resources. The `Storage` resource is located in the Redfish hierarchy contained by `ComputerSystems`, typically running as ApplicationServers. The physical infrastructure is modeled using Redfish `Chassis`. Managed resources are connected to the `Storage` resource, including `Volumes` and `StoragePools`.

The integrated configuration is illustrated in Figure 4.
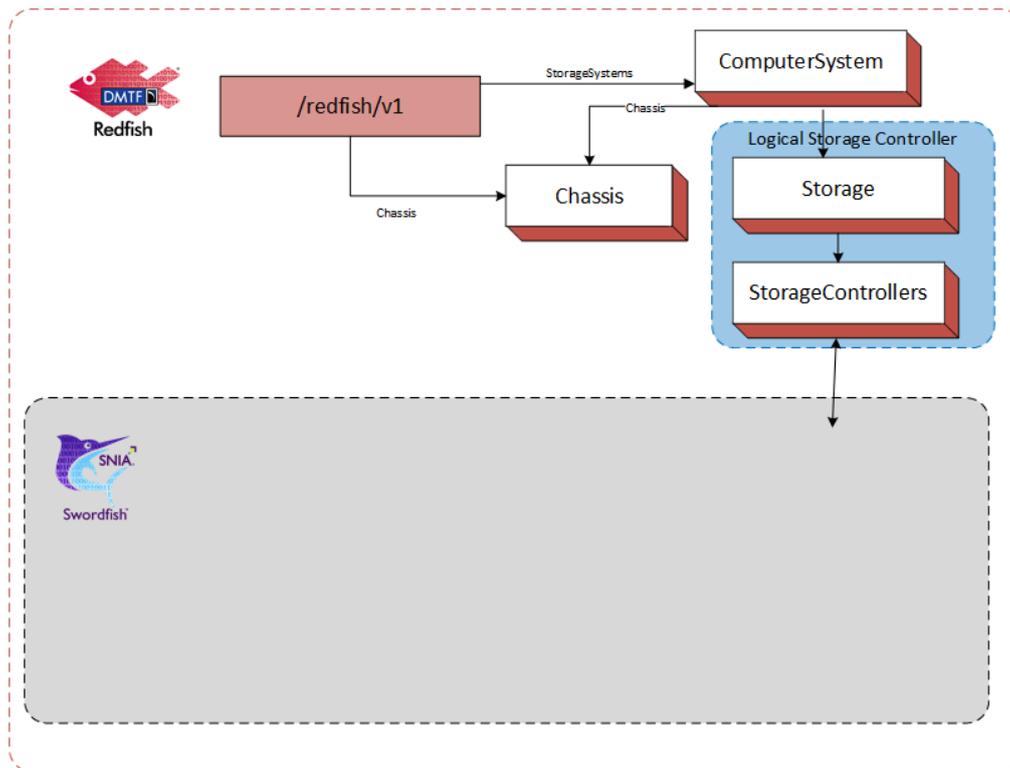
Figure 4: Logical Subsystem in Swordfish Integrated Configuration

This configuration works well when the storage system can be modeled by simply instantiating a new Storage object within an existing computer system. An example of a Storage System for an integrated configuration is shown in Figure ref{Figure_5}.

Figure 5: Swordfish Integrated Configuration Example

5.4 Swordfish model overview

MODEL OVERVIEW DESCRIPTION ### The ServiceRoot

The service root serves as the base for any Redfish or Swordfish service. It is always published at a well-known URI of `/redfish/v1`. The base service root can be accessed without a secure connection.

There are multiple required services and resources defined in the Redfish specification for any Redfish or Swordfish service; these include components such as the OData service document at `/redfish/v1/odata`, the schema metadata at `/redfish/v1/$metadata`, and services to provide authentication. Refer to the Redfish Specification for full details.

For Swordfish management, the following elements are common:

- `Storage`: A reference to the Storage resource collection.
- `Systems`: A reference to a Systems resource collection.
- `Chassis`: A reference to a Chassis resource collection.

- `Registries`: A set of administrative elements containing a Features registry and other components.
- `Managers`: A reference to a Managers resource collection.
- `Fabrics`: A reference to a Fabrics resource collection.
- `EventService`: The event management service.
- `StorageSystems`: An optional reference to a `StorageSystems` resource collection.

5.4.0.1 The Storage resource collection   A resource collection that references a set of `Storage` resources that each represents a storage subsystem. This collection can contain either resources or references to instances of `Storage` resources. Each `Storage` resource represents an instance of a storage subsystem. For Swordfish subsystems, refer to the Swordfish model overview for details on required elements.

5.4.0.2 The Systems resource collection   A resource collection that references a set of `ComputerSystem` resources that each represents a general purpose application server. Each `ComputerSystem` resource will have an entry with the value of "ApplicationServer" in its `HostingRoles` property. A particular `ComputerSystem` resource can be in both the `StorageSystems` and `Systems` collections.

5.4.0.3 The Chassis resource collection   A resource collection that references a set of `Chassis` resources. Each Chassis resource represents physical containers (i.e., sheet-metal confined spaces and logical zones like racks, enclosures, chassis, and all other containers). Subsystems (e.g., sensors), which operate outside of a system's data plane (meaning the resources are not accessible to software running on the system) are linked either directly or indirectly through this resource.

5.4.1 The StorageSystems resources collection

An optional reference to a `ComputerSystemCollection` with members of type `ComputerSystem` that have Swordfish managed storage attached. They will have an entry with the value of "StorageServer" in their `HostingRoles` property. This collection, then, is a resource collection that references a set of `ComputerSystem` resources that each represents a storage server. Each `ComputerSystem` resource will have an entry with the value of "StorageServer" in its `HostingRoles` property. A particular `ComputerSystem` resource can be a member of both the `StorageSystems` resource collection and the `Systems` resource collection.

5.4.2 The Storage resource

The storage system exposes logical storage, associated resources and related functionality. Storage resources can be found in the service root or service container via the `Storage` resource collection, and are attached to the `Storage` object within the `Storage` resource collection.

The storage system typically provides the ability to create, manage and present block, file or object store from a set of back-end media, presented to one or more hosts. Storage controllers can work in coordinated sets of one or more to present value-add capabilities, such as failover, data protection, and data path management within the storage system, that are represented through the various resources within the storage system.

The following are the principal properties of `Storage` that point to resources managed or defined by the storage system:

- `Controllers`: A reference to a collection of `StorageController` resources.
- `Drives`: A reference to a collection of `Drive` resources used for storage.
- `Enclosures`: A reference to a collection of Chassis resources that contain storage related resources.
- `FileSystems`: A reference to a collection of `FileSystem` resources.
- `StorageGroups`: Deprecated in favor of `ConsistencyGroups`.
- `ConsistencyGroups`: A reference to a collection of `ConsistencyGroup` resources.
- `StoragePools`: A reference to a collection of `StorageGroup` resources.
- `Volumes`: A reference to a collection of `Volume` resources.

Key properties of `Storage` used for access rights management, connectivity management, and consistency management, include:

- `Endpoints`: A reference to a collection of `Endpoint` resources used to access storage.
- `EndpointGroups`: A reference to a collection of `EndpointGroup` resources.

The following properties are created in the /redfish/v1/Fabrics to support access rights management, connectivity management, and consistency management, include:

- `Fabrics`: A reference to a collection of `Fabric` resources.

- `Connections`: A reference to a collection of `Connection` resources.
- `Zones`: A reference to a collection of `Zone` resources.
- `Switches`: A reference to a collection of `Switch` resources.

5.4.2.1 The `StorageController` resource The storage controller presents the foundational resources used by the storage system. It generally contains connectivity resources between the system and connected consumers.

A `StorageController` may represent either a physical or logical controller.

For direct-attach configurations, a `StorageController` instance is commonly used to represent the host-side physical controller, such as an HBA, or RAID controller. This includes properties such as `Port` to enumerate connectivity information.

Other key properties, when representing a physical controller:

- `SpeedGbps`
- `SupportedControllerProtocols` and `SupportedDeviceProtocols`
- Controller capabilities, such as cache, battery backup information, and `SupportedRAIDTypes`
- `AttachedVolumes`
- `EnvironmentMetrics` - only used for physical representations
- Ports

For NVMe devices, a storage controller is the interface between a host and an NVM subsystem. When PCI Express is used as the transport, a controller is also a PCI Express function.

It may represent the following capabilities:

- Modeling the interface between hosts and Volumes (IO controller)
- Access to administrative / management capabilities (admin controller)
- Network discovery of available volumes, and subsystems (discovery controller)

Key properties when representing a logical NVMe controller:

- `Endpoints`
- `SupportedControllerProtocols`
- `AttachedVolumes`
- `NVMeControllerAttributes`, `NVMeControllerProperties`

5.4.2.2 The Endpoint resource   Endpoints represent one end of a protocol specific connection that supports sending or receiving messages according to a particular protocol. Endpoints are used for access rights management, connectivity management, and consistency management. `Endpoint` objects are instantiated in the /redfish/v1/Fabrics/Endpoints collection, and referenced elsewhere.

For access rights management, the `Endpoint` is used in conjunction with `Connections`. This use case reflects the target's view of configured access to initiators.

For connectivity rights management, the `Endpoint` is used in conjunction with `Switches` and `Zones`. In this case, the endpoints reflect the fabric view of both the initiator and target endpoint identifiers.

Key properties for Endpoints for both access rights management and connectivity management include:

- `EndpointProtocol`
- identifiers
- `ConnectedEntities`

5.4.2.3 The Endpoint Collection resource   The `EndpointCollection` is a resource collection that references a set of `Endpoint` resources.

5.4.2.4 The EndpointGroup resources   `EndpointGroup` resources contain a set of Endpoints to be used for the same purpose, such as for access management, connectivity management, and consistency management. EndpointGroups can be used to simplify operations against large groups of Endpoints.

For consistency management, Endpoints are used to group together to provide single access and connectivity management for volumes in consistency groups.

5.4.3 The ConsistencyGroup resource

`ConsistencyGroups` represent a set of volumes that are managed as a group. Group level management allows system and application level activities to be performed on a set of data that spans volumes. `ConsistencyGroups` are typically used to enforce write-order consistent behavior throughout a set of members. Other activities include device-level replication operations, as well as system level functions, such as reset.

When `ConsistencyGroups` are implemented, they are attached to a `Storage` resource, and its internal `Volumes` collection is constructed from a subset of the `Volumes` collection of the `Storage` resource. The `ConsistencyGroup` can also be used to manage across replicas.

`ConsistencyGroups` should be used in place of the deprecated `Storage-Groups`.

5.4.4 The ConsistencyGroup Collection resource

The `ConsistencyGroupCollection` is a resource collection that references a set of `ConsistencyGroup` resources.

5.4.5 The StorageGroup resource

`StorageGroup` usage has been deprecated in favor of `ConsistencyGroup` usage.

StorageGroups represent a set of volumes that are managed as a group in order to facilitate mapping and masking, in which the volumes of a storage group are collectively exposed or hidden to a set of clients.

The set of volumes is specified by the `Mapped  Volumes` attribute. `MappedVolumes` is a resource collection of the `Mapped  Volume` construct (a tuple of a pointer to a volume and a corresponding Logical Unit Number for that volume).

The set of client endpoints to which the volumes can be exposed is specified by the `ClientEndpointGroups` attribute. The `ClientEndpointGroup` resource specifies a collection of `EndpointGroup` resources.

The set of server endpoints to which the volumes can be exposed is specified by the `ServerEndpointGroups` attribute. The `ServerEndpointGroup` resource specifies a collection of `EndpointGroup` resources.

5.4.6 The StoragePool resource

The `StoragePool` resource represents unassigned storage capacity that can be used to produce storage volumes or other storage pools.

The following are the principal properties of `StoragePool` that are used to create or identify resources provisioned or supported by the storage pool:

- `AllocatedVolumes`: A reference to a collection of `Volume` resources that have been provisioned from the storage pool.

- `AllocatedPools`: A reference to a collection of `StoragePool` resources that have been provisioned from the storage pool.

- `CapacitySources`: A reference to a resource collection that provides pointers to the capacity sources that are used to provide the underlying capacity for this storage pool.

- `RAIDTypes[]`: The set of RAIDTypes supported by this StoragePool. This may be set upon StoragePool creation, or may be a reflection of the implementation's ability to support different RAID types. Consumers may use this property to determine what RAID types are available from specific StoragePool instances for additional Volume creation requests, or what RAIDTypes have been applied to Volumes already allocated.

- Capacity (allocated and consumed)

- IO / performance and other metrics

- Spare management

- Pool type

Key operations include managing efficiency, protection and replication capabilities (e.g., deduplication, compression, encryption).

For NVMe and NVMe-oF devices, StoragePools continue to represent the underlying storage capacity. StoragePools are used to map to collections of storage capacity that have specific, related properties. These include NVMe Endurance Groups and NVM Sets.

5.4.6.1 The Volume resource `Volume` resource represents an addressable container of storage, sometimes referred to as a "Logical Unit", "LU", "LUN", or "StorageVolume" in the storage industry.

`Volume` is a foundational object for storage, and is extended in many ways, dependent on the desired system feature set. Potential usages for `Volume` include:

- providing for block-addressable storage, `Volumes` can also be further refined for particular implementations, such as key-value, object store, or other technologies;

- serving as building block for FileSystems, FileShares, datastores, and object stores;
- being grouped into write-order consistent ConsistencyGroups;
- virtualizing physical media (e.g., drives, memory);
- providing protection from underlying media failures;
- allowing for aggregation or slicing of physical resources into desired size;
- facilitating storage efficiency and protection representation and management (e.g., compression, deduplication);
- supportng replication and protection (e.g., encryption).

Key related elements: - `Endpoint`: Access to the volume is modeled through `Endpoints`. These cover direct connect protocols (e.g., SAS and SATA), as well as fabric-capable connection protocols. - `StoragePool`: Many of the characteristics of a Volume can be also managed across a set of Volumes from its `StoragePool`.

For block-addressable applications, key properties include information representing the block usage:

- BlockSizeBytes
- Capacity, including allocated and consumed
- RAIDType
- Access capabilities - modeled through Endpoints
- Relationship to underlying storage constructs (physical or logical) is typically represented through the `StoragePool`
- Usage (when known by the system)
- IO / performance and other metrics

Operations for block-addressable `Volume` may include:

- Expand
- Create / manage replicas
- Encrypt / SecureErase
- Initialize
- Change RAID layout and / or type

The `Operations` property presents a list of Operations currently running on the Volume. As Operations are both long-running and may be invoked from either the Redfish/Swordfish interface or via independent means, the `Operations` property provides a mechanism independent of the task service to show their status.

A GET on the Volume during specific long-running operations should also clearly indicate the in-process change, setting the Volume's Status.Health and State.State correspondingly.

- While transformational changes are in progress, the Status.Health should be set to "Updating" and Status.State should be set to "Warning". This includes:

  - Change RAID layout and / or type
  - Encrypt / decrypt

- Set Status.State to "UnavailableOffline" when in the process of creating a Volume and the volume is not yet available for use:

  - Initialize

Volumes are also central to application-specific extensions to storage modeling. For example:

- When using replication, StorageReplicaInfo contains descriptions and pointers to the Volumes employed.
- For storage devices with NVMe interfaces, Volumes:

  - Represent NVMe namespaces
  - Namespaces can be block, object, ZNS (zoned namespace), and KV (key value), which can be determined by the Type property within NVMeNamespaceProperties
  - Contain indications of NVMe capabilities via NamespaceFeatures
  - Show relationships to NVMe logical controllers

Additional information can be found in the NVMe Command Set Specification.

5.4.7 The FileSystem resource

This FileSystem resource represents a file system. Each FileSystem may contain a collection of FileShares that can be presented to hosts.

5.4.7.1 The Fabric resource   The Fabric resource is used to reflect both access rights and connectivity.

When representing access rights, a Fabric contains:

- Endpoints
- Connections
- EndpointGroups, as needed

In this role, the Fabric contains objects that define access capabilities, and the scope that they apply to. This includes the logical data storage entities (e.g., volumes,

filesystems) to which the defined access rights apply, as well as the protocol connection information for each included object.

A fabric modelling connectivity management contains objects that define the corresponding transport protocol's access and configuration information, as well as the protocol connection information for logical objects within the storage system. Connectivity is reflected by describing the allowed or restricted connections. Connectivity-related objects within a `Fabric` include:

- `Endpoints`
- `EndpointGroups`
- `Switches`
- `Zones`

5.4.7.2 The `Connection` resource The `Connection` resource describes the access permissions that `Endpoints` (or groups of endpoints) have with other resources. It also describes the association of access capabilities to specified logical data storage entities.

Key properties for Connections include:

- `ConnectionType`
- `VolumeInfo`
- `Endpoints` and `EndpointGroups` (in `Links`)

5.4.7.3 The Zone resource   The Zone resource describes the allowed / disallowed connectivity between endpoints.

5.4.7.4 The Switch resource   The `Switch` resource describes physical interconnect information, including physical port configuration information.

Key properties include:

- `SupportedProtocols`
- `SwitchType`
- `Ports`
- Other properties as needed to describe the physical switch attributes for monitoring, asset management, and configuration

6 Features and Profiles

6.1 Overview

Features are high-level descriptions of functionality which an implementation uses to advertise what functionality it currently supports, and for some features, is capable of supporting.

The detailed definitions required to describe to implementers how to implement a feature are written in profile definition files. A feature is generally represented in one (but may be more) profile definition file, or profile.

Profiles are detailed descriptions that describe down to the individual property level what functionality is required in order to advertise features. Different profile definitions can exist for the same feature type but for various types of storage configurations: Swordfish.Block.Provisioning, Swordfish.File.Provisioning

The Swordfish Features Registry shall be used to advertise what standard and Oem Features an implementation supports.

6.2 Requirement for SupportedFeatures

`SupportedFeatures` entries in the `Features` registry represent the client's primary initial runtime view of the capabilities of a Swordfish implementation. Without properly formed entries in this registry, there is no visibility to an implementation's functionality.

Swordfish implementations shall implement the `Features` registry and advertise at least one supported feature from the `SwordfishFeaturesRegistry` in order to be considered a Swordfish implementation.

Features define coarse-grained sets of functionality. In order to advertise a feature (using the `SupportedFeature` mechanism in the SupportedFeatures Registry), the implementation must support the complete set of functionality as defined in the corresponding profile.

The Swordfish Features Registry publishes the official list of supported SNIA Features, and provides a high-level description of their functionality. Many of those features are self-explanatory (e.g., local replication, remote replication), but there are some features where additional context is appropriate:

- Energy Star for Storage
- NVMe and NVMe-oF

- Class of Service

6.3 EnergyStar for Storage Feature

The EnergyStar for Storage Feature and profile has been created to formalize the requirements from the ENERGY STAR Data Center Storage Program Requirements on storage products. The profile indicates what properties Swordfish implementations need to support in order to properly instrument EnergyStar reporting capability. This functionality is intended to support EnergyStar data gathering requirements as part of the EnergyStar certification process.

6.4 NVMe and NVMe-oF Features

NVMe features and profiles are defined to provide definitions for specific NVMe devices and configurations. The NVMe profiles also include references to non-NVMe specific profiles, such as those specifying the features for discovery, access rights, connectivity rights, and management controllers.

Profiles defined specifically for NVMe device types all use a common naming scheme, and typically start with SwordfishNVMe.

6.5 Class of Service Feature

6.5.1 Overview

Swordfish supports a `ClassOfService` feature. The `ClassOfService` functionality supports systems that are capable of providing a greater level of management automation, where a higher-level set of goals is provided as direction rather than requiring parameterized inputs for all configuration actions.

The Class of Service feature uses a combination of device-defined capabilities to structure LinesOfService, which are sets of available functionality in a given system, that can then be grouped together to provide classes of service.

When Class of service functionality is implemented, the Swordfish functionality may be entirely exposed through the StorageService resource. Each Swordfish Storage-Service is located in the ServiceRoot (and ServiceContainer) via the StorageServices resource collection.

6.5.2 Class of Service Model

For Swordfish with a class of service interface, the following two models apply. Either model choice results in the same storage service, regardless of the storage system model.

1. Integrated Service Configuration

The storage systems managed by the Swordfish storage service are modeled using the Redfish `Storage` resource and `StorageController` resource collections. The `Storage` resource is located in the Redfish hierarchy contained by `ComputerSystems`, typically running as ApplicationServers. The physical infrastructure is modeled using Redfish `Chassis`, as shown in Figure 6.
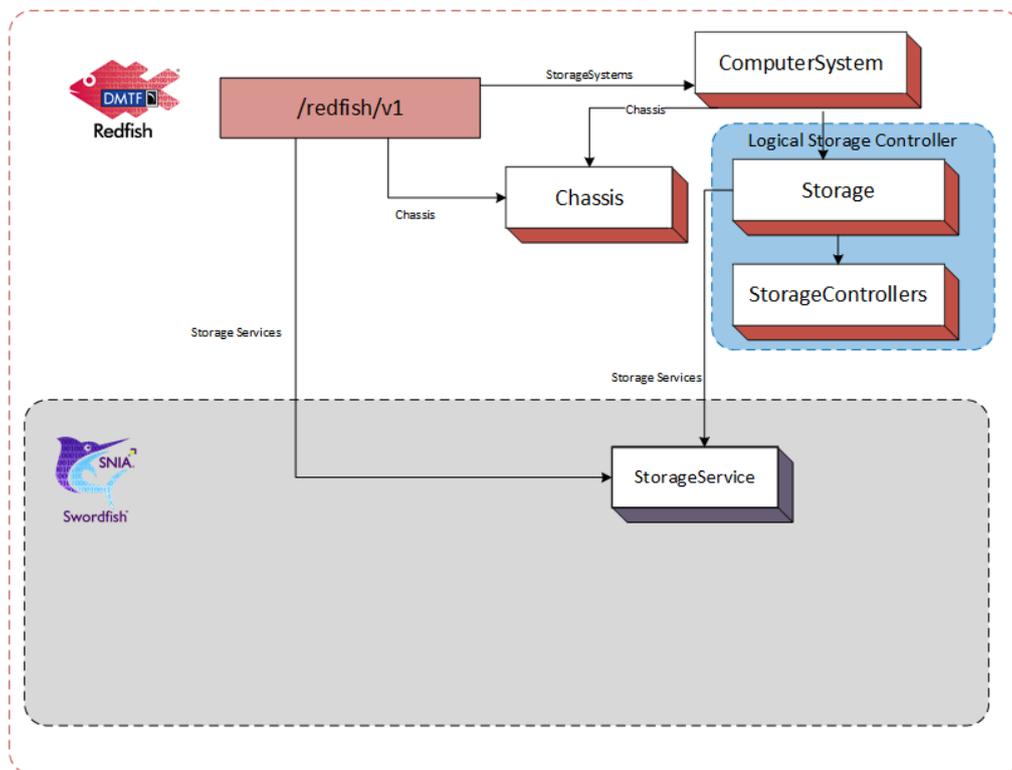
Figure 6: Logical Subsystem in Integrated Service Configuration

This configuration works well when the storage service is hosted by a storage resource within a computer system. An example of a Storage Service for an integrated service configuration is shown in Figure 7.

Note: This diagram and the discussion of the configuration description have been simplified slightly to avoid confusion. A full implementation would likely include additional links to the logical storage controller resources.
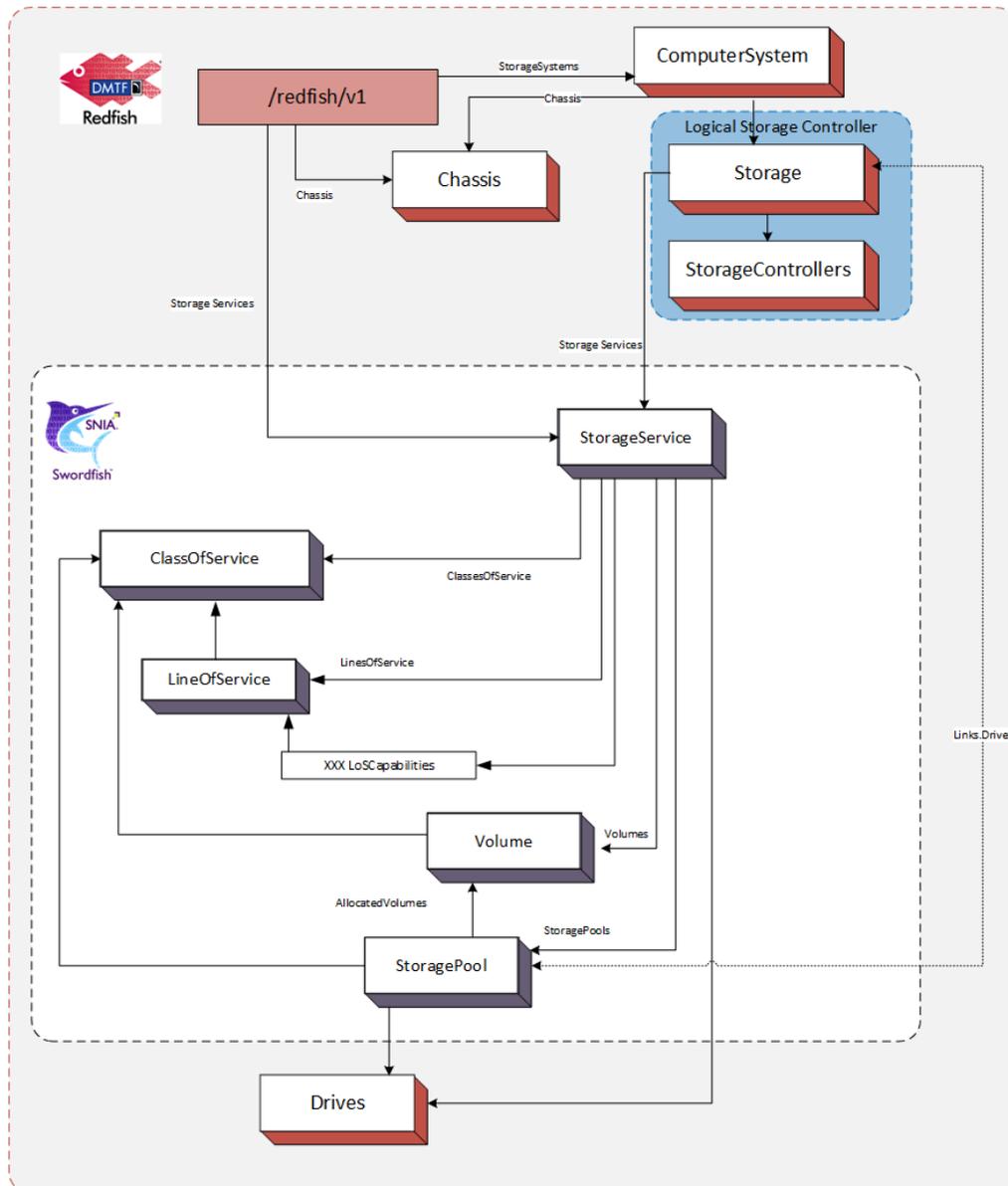
Figure 7: Integrated Service Configuration Example

2. Standalone Service Configuration

The storage systems managed by the Swordfish storage service are located in the ServiceRoot (and ServiceContainer) via the Storage resource collection.

They model the logical storage system using Redfish `Storage` and 'StorageController' resources. The physical infrastructure is modeled using Redfish `Chassis`. This is shown in Figure 8.
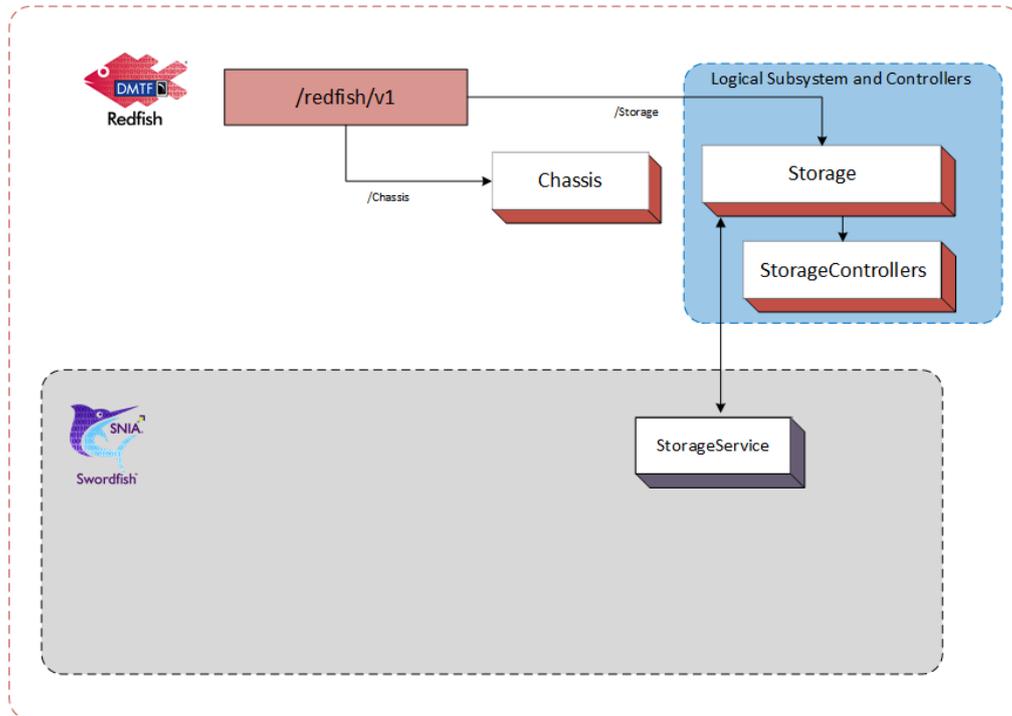
Figure 8: Logical Subsystem in Standalone Service Configuration

This configuration works well when the standalone storage system directly hosts the storage service(s). An example of a Storage Service for a hosted service configuration is shown in Figure 9.

> Note: This diagram and the discussion of the configuration description have been simplified slightly to avoid confusion. A full implementation would likely include additional links to the logical storage controller resources.
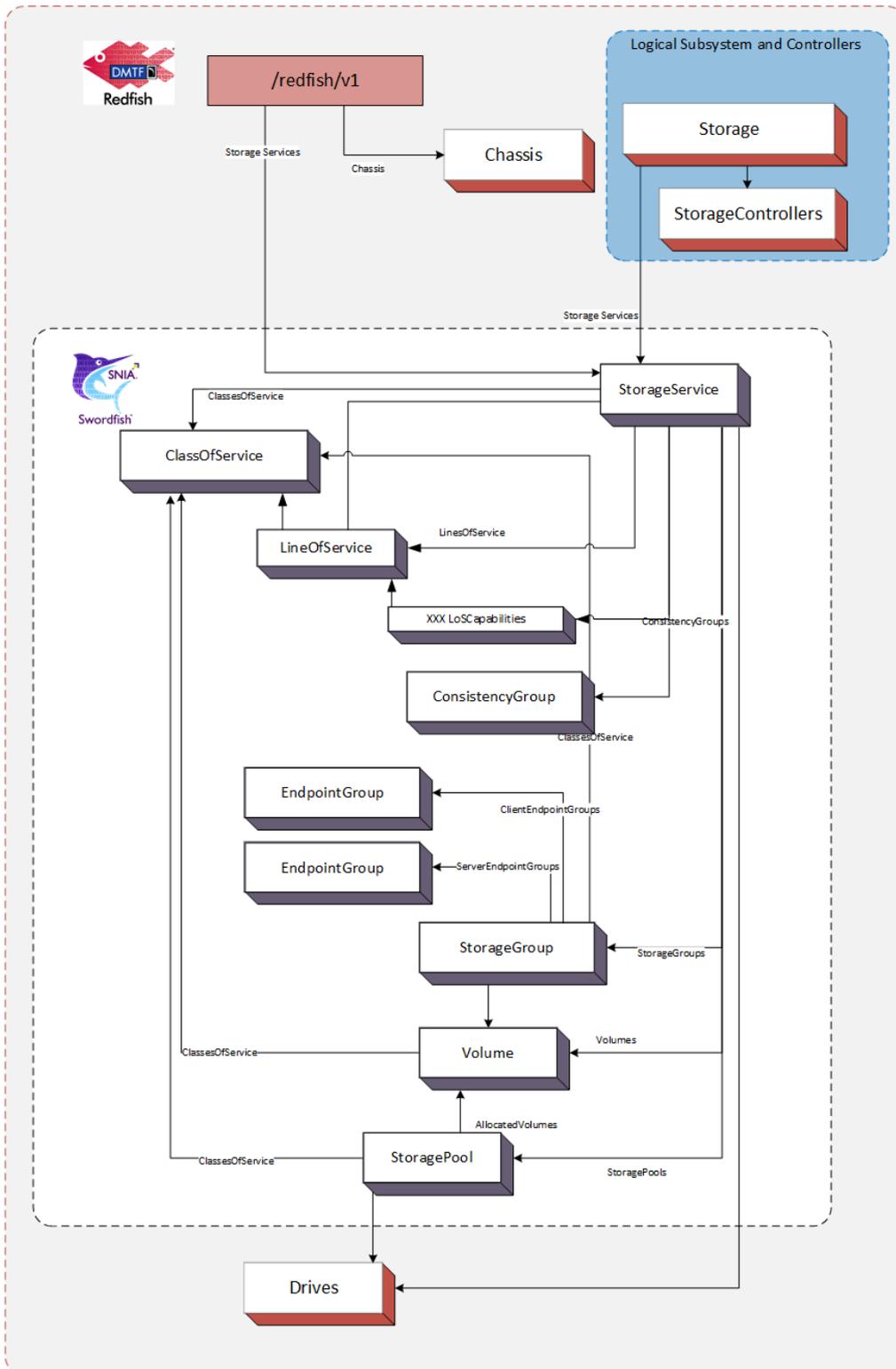
Figure 9: Standalone Service Configuration Example

6.5.3 ServiceRoot Additions

When the StorageService feature is implemented, the following is added to the `ServiceRoot`:

- `StorageServices`: A resource collection that references a set of `StorageService` resources. Each `StorageService` resource represents the resources and behaviors supported by that storage service.

6.5.4 The StorageService resource

6.5.4.1 Principal Properties   The storage service is hosted on a storage system and exposes logical storage, associated resources and related functionality. Storage service resources can be found in the service root or service container via the `StorageServices` resource collection.

The following are the principal properties of `StorageService` that point to resources managed or defined by the storage service:

- `ClassesOfService`: A reference to a resource collection that specifies the supported `ClassOfService` resources.
- `Drives`: A reference to a resource collection that collects `Drive` resources used for storage.
- `Enclosures`: A reference to a resource collection that collects Chassis resources that contain storage related resources.
- `Endpoints`: A reference to a resource collection that collectsEndpoint resources used to access storage.
- `FileSystems`: A reference to a resource collection that collects `FileSystem` resources.
- `EndpointGroups`: A reference to a resource collection that collects `EndpointGroups` resources.
- `StorageGroups`: A reference to a resource collection that collects `StorageGroup` resources.
- `StoragePools`: A reference to a resource collection that collects `StorageGroup` resources.
- `Volumes`: A reference to a resource collection that collects `Volume` resources.
- `HostingSystem`: A reference to the `ComputerSystem` instance that hosts this `StorageService`.

6.5.4.2 Capabilities and Lines of ServiceRoot   The following properties each define a set of attributes, which describe capabilities that the storage service may support:

- `DataProtectionLoSCapabilities`: Replicas that protects data from loss.
- `DataSecurityLoSCapabilities`: Data security service level requirements. The data security characteristics enable the storage system to be used in an environment where compliance with an externally-specified security standard or standards is required. Examples of such standards include FIPS-140, HIPAA and PCI.
- `DataStorageLoSCapabilities`: Provisioning and access characteristics for storage of the data.
- `IOConnectivityLoSCapabilities`: IO connectivity requirements for access to the data.
- `IOPerformanceLoSCapabilities`: IO performance requirements for access to the data.

In each of the above, not all combinations of attribute values defined within a capability are likely to be supported by the storage service.

Known, supported combinations of attribute values are used to construct entries in the `LinesOfService` array property. Not all attributes of a line of service entry need be specified (i.e. some may be Null). If an attribute has no value, the storage service may choose any supported values when provisioning for that entry. Otherwise, the line of service attribute values specifies the kind or level of service to be provided.

6.5.4.3 The ClassOfService resource   A class of service represents a choice of utility or warranty offered to customers by a service. (ITIL uses the term `service option`. See the Normative References.)

Each `ClassOfService` resource is a uniquely named description of the characteristics of one choice of utility or warranty for a service. Each `ClassOfService` is a description of the kind and quality of service to provide and is not intended to describe how the service provides that service.

Each `ClassOfService` is defined by an aggregation of lines of service. Supported lines of service are listed in the corresponding capabilities attributes of the storage service, (see above).

Currently defined lines of service are:

- Data Protection: Describes the characteristics of a replica that protects data from loss.
- Data Security: Describe data security service level requirements. The data security characteristics enable the storage system to be used in an environment where compliance with an externally-specified security standard or standards is required. Examples of such standards include FIPS-140, HIPAA and PCI.
- Data Storage: Describes provisioning and access characteristics for storage of the data.
- IO Connectivity: Describes IO connectivity requirements for access to the data.
- IO Performance: Describes the IO performance requirements for access to the data under a particular workload.

Some advertised `ClassOfService` resources are created by the service implementation. These are generally not changeable and are intrinsic to the implementation.

A service may support creation or modification of `ClassOfService` resources. All must be consistent with the capabilities of the service.

6.5.4.4 The StoragePool resource   When a Swordfish implementation advertises support for the Class of Service feature, the `StoragePool` resource now presents a new method to the client to allocate unassigned storage capacity. This is automated by the system as conformance to one or more classes of service. Requests to `StoragePool` shall automatically allocate capacity based on the constraints of the selected class of service and any other selected parameters, with priority given to the class of service constraints.

The following are the principal properties of `StoragePool` that are used to identify resources provisioned or supported by the storage pool related to Class of Storage:

- `ClassesOfService`: A reference to a resource collection that specifies the set `ClassOfService` resources that can be specified when provisioning resources from the storage pool.
- `DefaultClassOfService`: A reference to the default `ClassOfService` resources used for provisioning from the storage pool.

6.5.4.5 The Volume resource   `Volume` resource represents a block-addressable container of storage, sometimes referred to as a "Logical Unit", "LU", "LUN", or "StorageVolume" in the storage industry. Volumes optionally adhere to a `ClassOfService`, which defines added functionality. Examples include:

- Access capabilities

- Capacity and capacity sources
- Consumption tracking (e.g., LowSpaceWarningThresholdPercents)
- Replication details
- StorageGroup Information

6.5.4.6 The FileSystem resource   In a Swordfish implementation that advertises support for the Class of Service feature, File systems represent file-addressable capacity that are conformant to a `ClassOfService`.

7 Schema Considerations

7.1 Schema Introduction

7.1.1 Overview

A Swordfish implementation is a Redfish implementation, and as such it minimally includes support for some Redfish-defined schema, including `ServiceRoot`, `ComputerSystem`, and `Storage` (which has been enhanced to include many Swordfish-centric properties). Swordfish implementations also include support for Swordfish-defined schema.

The Swordfish model focuses primarily on the logical model of a storage system, and does not require full representation of a physical instantiation. This is left to the implementer to complete from available Redfish schema models.

Swordfish schema is conformant with the rules used to define Redfish schema. Redfish schema is conformant with the Common Schema Definition Language, see CSDL. This section provides additional definition and context for the CSDL elements used to define Swordfish schema.

7.1.2 Schema Primacy

Redfish and Swordfish schema are made available in multiple formats, including CSDL, JSON and yaml. The development source for the Swordfish specification is CSDL; other variants are derived from the CSDL source. If conflicts and questions arise between the definitions, CSDL shall be treated as the definitive source.

7.1.3 Swordfish Extension of the Redfish ServiceRoot

The Redfish `ServiceRoot` has properties that provide access to Swordfish resources.

XXX - ADD EMERGING TECHNOLOGY TAG.

For implementations that advertise support for the `ClassOfService` feature, the implementation shall instantiate a collection of `StorageServices`at the `ServiceRoot` with at least one member. The collection provides the client an efficient means to search across all `StorageService` resources, regardless of which `ComputerSystem`is supporting the service.

7.2 Default values and NULLABLE attributes

The interaction of `Nullable` and `DefaultValue` needs to be clearly understood by both implementers and client developers. The possible combinations of are summarized in Table 7. The table contains:

- **Nullable**: True, if a given property may be NULL
- **DefaultValue**: True, if a default value is provided for a given property
- **Client**: True, if a client value is supplied for a given property in a query or response
- **Result**: The resultant value of the given property. One of:
    - *C*: The client-provided value
    - *D*: The default value
    - *Null*: Null
    - *I*: Implementation defined
    - *Error*: Error state

Table 7: Schema annotations

Nullable	DefaultValue	Client	Value
T	T	T	C
T	T	F	D
T	F	T	C
T	F	F	I or Null
F	T	T	C
F	T	F	D
F	F	T	C
F	F	F	I or Error

7.3 Common schema annotations

Table 8 lists common annotation used in the definition of Swordfish, for details see OData Capabilities Vocabulary, OData Core Vocabulary, OData Measures Vocabulary, and Redfish Extensions.

Table 8: Default and Nullable Interaction

Name	Applies to	Description
AllowableValues	Parameter	The set of allowable values for a parameter
AutoExpand	NavigationProperty	If true, return expand the target element
AutoExpandReferences	NavigationProperty	If true, return references to the target element
ConformanceLevel	EntityContainer	Specifies OData conformance level
Deprecated	All	Specifies that the element may be removed in future major revisions, but shall continue to be supported as specified in the current revision.
Description	All	A brief description of a model element
LongDescription	All	A normative description of a model element
Maximum	Parameter, Property	Maximum value that an integer property or parameter may have
Minimum	Parameter, Property	Minimum value that an integer property or parameter may have
Pattern	Parameter, Property	Specifies a pattern that the value shall match
Permissions	NavigationProperty, Property	Access permission for the property.

Name	Applies to	Description
Required	NavigationProperty, Property	If true, property is required to be supported by the service. The default is optional. See *Required Properties*
RequiredOnCreate	NavigationProperty, Property	If true, property is required on creation. See *Required Properties*
Unit	Property	The unit of measure for the value.

7.4 Property implementation requirements

The client and the implementer should understand that, regardless of the schema declaration, an implementer may choose to not implement a property. If not implemented, a representation of the property will not be present in a reply. This should not be confused with a response that indicates that a property has been implemented, but has no value (i.e. *propertyName = null*).

There are several factors that could affect the implementation choice. Implementation requirements can be defined in many documents. At a minimum, a developer should review, in order: 1. the Redfish specification, 2. this document, and 3. associated profile specifications.

7.5 Schema repository

The primary online source for the Swordfish schema shall be co-located on the DMTF schema site with the Redfish schema: http://redfish.dmtf.org/schemas/swordfish/v1 Developers may also download the schema as part of the Swordfish bundle from snia.org (refer to snia.org/swordfish for pointers to the bundle locations).

Implementations should refer either to the versions available on the dmtf.org site or to locally provided instances of the schema.

7.6 Referencing other schemas

Swordfish directly references many Redfish schemas when functionality is already defined and can be leveraged. Other Redfish schema may be added by inference or directly to implementations. Examples are available in the Swordfish mockups.

7.7 Swordfish online repositories

7.7.1 Registries

The primary online source for the Swordfish registries shall be co-located on the DMTF registry site with the Redfish registry: http://redfish.dmtf.org/registries/swordfish/v1 Developers may also download the schema as part of the Swordfish bundle from snia.org (refer to snia.org/swordfish for pointers to the bundle locations).

Implementations should refer either to the versions available on the dmtf.org site or to locally provided instances of the registries.

7.7.2 Profiles

The primary online source for the Swordfish registries shall be co-located on the DMTF profile site with the Redfish profile: http://redfish.dmtf.org/profiles/swordfish/v1

Developers may also download the profiles as part of the Swordfish bundle from snia.org (refer to snia.org/swordfish for pointers to the bundle locations).

Implementations should refer either to the versions available on the dmtf.org site or to locally provided instances of the profiles.

8 Implementation requirements

8.1 Security

This document generally adheres to the security requirements defined in the Redfish Specification. It extends the Redfish security model in one important way:

> Swordfish implementations shall implement TLS as per the guidance in ISO/IEC 20648 and the TLS Specification for Storage Systems.

8.2 General constraints

8.2.1 Redfish elements

The Swordfish service interface extends the Redfish service interface. As such, a Swordfish service is a Redfish service and all required elements of the Redfish model shall be present in a Swordfish model.

A Swordfish implementation shall conform to all requirements specified in the Redfish specifications.

Swordfish functionality shall not conflict with any previously defined Redfish functionality but it may add to or extend it, and it may add additional constraints on Redfish functionality.

Additionally, any functionality desired in a Swordfish implementation that is specified in Redfish shall follow the requirements as specified in the Redfish specification.

8.2.2 Storage Events

8.2.2.1 Overview   A Swordfish implementation should implement an event service. Redfish defines the Event Service framework, client subscription model, event delivery mechanism, as well as standard message registries. Swordfish extends the standard message registries to provide additional message registries that correspond to Swordfish-specific services and properties.

The Redfish event service publishes a list of event types supported, and maintains a list of clients that have subscribed. Each subscription maps clients, subscribed events, and the resources that generate them.

8.2.2.2 Message Registry Selection and Management Swordfish constrains the existing event model to provide a more consistent handling of event notifications and the related messages, in order to assure that client systems can easily and consistently parse and respond to system-level events.

8.2.2.3 Required Usage

- The Resource Event Message Registry defines the underlying messaging model, and shall be used to map messages to resources for storage implementations.
- The Redfish Base Message Registry shall be used to support HTTP connection/error/protocol issues, and general errors.
- The Swordfish Message Registry shall be used as a supplement for the resource event message registry.
- If the Swordfish service implements Redfish tasks (i.e., long-running operations), the implementation shall use the messages defined in the Task Event Message Registry to report status.

8.2.2.4 Recommended Usage

- Standard Messages should be used, wherever possible.
- OEM messages should be avoided. Suggestions for clarification or expansion of the existing registries are encouraged. (submissions should be sent to the SNIA Feedback Portal)

8.2.3 Health and HealthRollup Propagation

8.2.3.1 Overview The `Status` object includes both a `Health` property, intended to reflect the health of a given resource or component in a standardized way, and a `HealthRollup` property, which is meant to aggregate the health of a hierachy of subordinate or associated resources or components. Swordfish introduces some additional guidance around the proper use of these two properties.

8.2.3.2 Status.Health The Redfish schema requires that `Status.Health` "represent the health state of the resource without considering its dependent resources." While Swordfish defines no change to that usage, implementors are encouraged to assure that the information reflected in `Status.Health` reflects solely the state of the reporting object. For example, a given `StoragePool` instance can report OK for its own `Status.Health`, even if a volume is reporting a `Status.Health` value

of `Critical` because it has exceeded a space quota and been taken offline, as the storage pool itself has no issues.

8.2.3.3 Status.HealthRollup   The Redfish schema requires that `Status.HealthRollup` "represent the health state of the resource and its dependent resources". As a general rule, `Status.HealthRollup` is intended to aggregate any error conditions found in subordinate levels in a hierarchy of components or subsystems, and allow a client to traverse the hierarchy to locate the underlying source of a problem or error state. At the same time, an implementation should not blindly retain the severity of a component-level problem, when higher-level objects or systems provide additional protection against serious failure or data loss. Accordingly:

- `Status.HealthRollup` shall represent the general health of the reporting resource and its dependent resources, and shall not report an value of OK if the `status.health` value of any dependent resource reports a value other than OK;
- `Status.HealthRollup` should propagate a lower-level error status and its severity, particularly if the reporting resource doesn't introduce additional redundancy or error protection;
- `Status.HealthRollup` may report a lower level of severity than that returned by `Status.Health` of one of its dependent resources, provided that the ability to traverse the resource hierarchy to the root cause of a subordinate error is preserved. In particular, while a `Status.Health` value of `Critical` may be downgraded to `Warning`, a `Warning` value should not be downgraded to OK, as the ability to locate the root cause the subordinate error would be lost.

8.3 Discovering Swordfish resources

Each Swordfish implementation supports the following well-known URLs, as defined in Redfish. Specifically:

- `/Redfish`, which contains one or more version properties for the integrated Swordfish and Redfish implementation, starting with v1.
- `/Redfish/v1`, which addresses a `ServiceRoot` instance, which defines the Redfish default principal starting information for version 1 implementation of an integrated Redfish and Swordfish service. A GET operation to it shall retrieve the value of an instance of a `ServiceRoot EntityType` as defined in the ServiceRoot_v1.xml file.

- `/Redfish/v1/$metadata`, an OData metadata document containing descriptions of all top-level service resources and resource types used in the implementation. `$metadata` contains a list of all the current schema instrumented in the service. Clients querying the $metadata URI will get a comprehensive view of all implemented schema.
- `/Redfish/v1/odata`, which addresses a `ServiceContainer` instance, which defines OData conformant principal starting information for the same version 1 implementation of an integrated Redfish and Swordfish service. A GET operation shall retrieve the value of an instance of a `Service-Container EntityContainer` as defined in the ServiceRoot_v1.xml file.

> Note: Since the `ServiceContainer` is required to return an `@odata.context` value of `/redfish/v1`, all other elements accessed via it will be the same elements found via the `ServiceRoot`.

A Swordfish service is a Redfish service with extensions to support storage management. No additional service entry-points are required by Swordfish.

Both the `ServiceRoot` and `ServiceContainer` provide access to the same resource collections. The `ServiceContainer` additionally has a Service attribute that references the `ServiceRoot` resource. Regardless of starting point, the property values of the `ServiceRoot` instance enable navigation to all other resources exposed by the Swordfish service.

8.3.1 Required Collections for Storage implementations

Swordfish implementations shall include the use of either `Storage` or `Storage-Service` objects.

If an implementation contains a `Storage` instance, the `redfish/v1/Storage` collection shall be implemented, and shall contain references to all `Storage` objects. This collection may also contain the primary instances of `Storage` objects.

If an implementation contains a `StorageService` instance, the `redfish/v1/StorageServices` collection shall be implemented, and contain references to all `StorageService` objects. This collection shall serve as the primary collection to contain `StorageService` instances.

8.4 ClassOfService requirements

Each `ClassOfService` shall include at least one line of service. The providing server shall assure that the line of service values of a `ClassOfService` collectively represent a supported choice of service.

8.5 HTTP status codes

8.5.1 Overview

Status codes are generally defined as part of the general HTTP protocol definition. In addition, the Redfish specification calls out general usage for HTTP status codes. This section provides additional usage guidance and constraints for Swordfish implementations.

In some instances, Redfish and Swordfish expand the standard use of HTTP status codes by associating additional system status with specific status codes. In addition, error response data may be included via standardized message registry entries. The specific messaging requirements will be defined in the following sections.

In cases where Swordfish adds additional constraints or expands on the Redfish handling of a given issue, this document will include a small wording extract from the Redfish specification for additional context. For example:

Swordfish refines the requirements in the *Redfish Specification*: `Redfish has no constraint on external storage functionality` to require that all references to external storage functionality shall be compliant with the current release of Swordfish.

8.5.2 Create

If a request to create a resource can be completed successfully without additional time, the Redfish service shall return a status code of 201, and the body of the response shall contain the JSON representation of the newly created resource.

If the create resource request has been accepted, but no information about the resource can be returned at this point, the Redfish service shall return a status code of 204. The payload of the response shall be empty, but the Location header shall contain the resource URI. The client will be required to poll the appropriate resource to determine both when and if the operation is complete.

Swordfish refines the requirements in of the *Redfish Specification* (see *"Data modification requests overview"* and *"Asynchronous operations"*).

If a request to create a resource cannot be completed without additional time, the implementation shall:

- Populate an initial object. It shall contain, at a minimum, a valid URI, required properties (e.g., ID, name), and Status.State;

- Set Status.State of the partially populated resource to "Creating";

- Return the appropriate status code, based on the following guidance:

  - If a Task Service has been implemented, the Redfish service shall return a status code of 202, with the Location header set to the URI of the Task Monitor. Once the provider has returned a Task Monitor to the client, the Client can then query the provided task URI to track the task completion status. Upon task completion, a GET against the task monitor may return a status code of 201, and the body of the message shall contain the created resource, provided the task monitor URI remains valid . Refer to the Redfish Task Manager documentation for the lifecycle of the task monitor URI.

  - If a Task Service has not been implemented, the Redfish service shall return a status code of 201, and the body of the response shall contain the URI of the skeletal resource created as part of accepting the request. The client will be required to poll the URI provided to determine when the operation is complete.

- Update Status.State for the object, once the create operation completes.

8.5.3 Update, Replace, Delete

If a request to modify or delete a resource can be completed without additional time, the Redfish service shall return a status code of 200, and the body of the response shall contain the JSON representation of the modified (or deleted) resource.

If the resource modification or deletion request has been accepted, but no information about the resource can be returned at this point, the Redfish service shall return a status code of 204. The payload of the response shall be empty. The client will be required to poll the appropriate collection to determine both when and if the operation is complete.

If a request to modify a resource cannot be completed without additional time, the implementation shall:

- Set Status.State of the partially populated resource to "Updating" or "Deleting", as appropriate;

- Return the appropriate status code, based on the following guidance:

- If a Task Service has been implemented, the Redfish service shall return a status code of 202, with the Location header set to the URI of the Task Monitor. Once the provider has returned a Task Monitor to the client, the Client can then query the provided task URI to track the task completion status. Upon task completion, a GET against the task monitor may return a status code of 201, and the body of the message shall contain the created resource, provided the task monitor URI remains valid . Refer to the Redfish Task Manager documentation for the lifecycle of the task monitor URI.

- If a Task Service has not been implemented, the Redfish service shall return a status code of 200, and the body of the response shall contain the URI of the skeletal resource created as part of accepting the request. The client will be required to poll the URI provided to determine when the operation is complete.

- For an update or replace request, the implementation shall update Status.State for the resource, once the modify operation completes.

8.5.4 Actions

Swordfish supports the approach to Actions in the *Redfish Specification*:

> Actions are Redfish operations that do not easily map to RESTful interface semantics. These types of operations may not directly affect properties in the Redfish Resources.

Swordfish refines the requirements in the *Redfish Specification*: `Services shall support the POST method to send actions to Resources.`

If a Task Service has been implemented, the Redfish service shall return a status code of 202, with the Location header set to the URI of the Task Monitor. Once the provider has returned a Task Monitor to the client, the Client can then query the provided task URI to track the task completion status. Once the task has completed successfully, a GET against the task monitor shall return the created object.

If a Task Service has not been implemented, the Redfish service shall return a status code of 200, and the body of the response shall contain the URI of the skeletal resource created as part of accepting the request. The client will be required to poll the URI provided to determine when the operation is complete. When processing actions, the handling of HTTP status codes is slightly different than that seen when processing CREATE or MODIFY requests. The HTTP status code is used to reflect the acceptance and formatting of the request. The outcome of any requested processing is reflected in the body of the returned message and its associated Error response structure. For example, a properly formatted request to execute a system reset may return an HTTP status code of 200 (OK), to reflect that the request has been received, was validly formatted, and has been accepted for processing, while the reset of the system may not complete successfully. The Error response structure would contain further detail of the success of failure of the system reset. The implementation must check both the HTTP status code and the underlying Error response message structure to confirm the successful execution of the action.

9 Swordfish type definitions

9.1 Overview

The following sections define the schema and type definitions that make up a Sword-fish implementation. Each data type or entity within the schema includes a description that defines its implementation requirements and their interaction.

9.2 Introduction

9.3 Universal properties

The properties summarized in Table 9 are defined for inclusion in every Redfish schema, and therefore may be encountered in any response payload. They are documented here to avoid repetition in the property tables. Note that several of these properties are payload annotations, but appear here because they are required for all Redfish and Swordfish Resources.

9.3.0.1 Properties

Table 9: Universal properties

Property	Type	Attributes	Notes
@odata.context	s tring (URI)	read-only*	• The value of this property shall be the context URL that describes the resource according to OData-Protocol and shall be of the form defined in the Redfish specification.
@odata.etag	s tring	read-only*	• The value of this property shall be a string that is defined by the ETag HTTP header definition in RFC7232.
@odata.id	s tring (URI)	read-only required	The value of this property shall be the unique identifier for the resource and it shall be of the form defined in the Redfish specification.

Property	Type	Attributes	Notes
@odata.type	s tring	*read-only required*	The value of this property shall be a URI fragment that specifies the type of the resource and it shall be of the form defined in the Redfish specification.
Description	s tring	• *read-only*	This property shall contain the description of this resource. The value shall conform with the 'Description' clause of the Redfish Specification.
• *Id**	s tring	*read-only required*	This property shall contain the identifier for this resource. The value shall conform with the 'Id' clause of the Redfish Specification.
N ame	s tring	*read-only required*	This property shall contain the name of this resource or array member. The value shall conform with the 'Name' clause of the Redfish Specification.
Oem {}	o bject		The manufacturer- or provider-specific extension moniker that divides the Oem object into sections.

9.4 Frequently used properties

In addition, the properties summarized in Table 10 are frequently defined in Redfish schemas. Their definition and usage is the same throughout the Redfish data model.

9.4.0.1 Properties

Table 10: Frequent properties

Property	Type	Attributes	Notes
• *Actions** {}	object		The Redfish actions available for this Resource.
Links {}	object		The links associated with the Resource, as defined by that Resource's schema definition. All associated reference properties defined for a Resource are nested under the Links property. Find all directly referenced, or subordinate, Resource properties from the root of the Resource.
RelatedItem [{	array		An array of links. Each link points to a Resource or part of a Resource as defined by that Resource's schema. This representation is not intended to be a strong linking methodology like other references. Instead, it shows a relationship between elements or subelements in disparate parts of the service. For example, fans might be in one area of the system and processors in another. The relationship between the two might not be obvious. This property can show that one is related to the other. In this example, it might indicate that a specific fan cools a specific processor.
@odata.id }]	string (URI)	• read-only*	The value of this property shall be the unique identifier for the resource and it shall be of the form defined in the Redfish specification.

9.5 Common Swordfish Objects

The following structures are included in multiple Swordfish schema, and therefore may be encountered in any Response payload. They are documented here to avoid repetition in the Swordfish Specification tables for each schema.

9.5.1 Capacity

9.5.1.1 Description   This composition may be used to represent storage capacity. The sum of the values in Data, Metadata, and Snapshot shall be equal to the total capacity for the data store.

9.5.1.2 Properties   The properties defined for the Capacity schema are summarized in Table 11.

Table 11: Capacity properties

Property	Type	Att ributes	Notes
Data {}	object		The value shall be capacity information relating to provisioned user data. For property details, see CapacityInfo.
IsThinProvisioned	boolean	*re ad-only (null)*	If the value is false, the capacity shall be fully allocated. The default value shall be false.
Metadata {}	object		The value shall be capacity information relating to provisioned system (non-user accessible) data. For property details, see CapacityInfo.

Property	Type	Att ributes	Notes
Snapshot {}	object		The value shall be capacity information relating to provisioned snapshot or backup data. For property details, see CapacityInfo.

9.5.2 CapacityInfo

9.5.2.1 Description   This composition may be used to represent the utilization of storage capacity.

9.5.2.2 Properties   The properties defined for the CapacityInfo schema are summarized in Table 12.

Table 12: CapacityInfo properties

Property	Type	Att ributes	Notes
AllocatedBytes	integer (By)	*re ad-writ e\<br\> {=html} (null)*	The value shall be the number of bytes currently allocated by the storage system in this data store for this data type.
ConsumedBytes	integer (By)	*r ead-onl y\<br\> {=html} (null)*	The value shall be the number of logical bytes currently consumed in this data store for this data type.
GuaranteedBytes	integer (By)	*re ad-writ e\<br\> {=html} (null)*	The value shall be the number of bytes the storage system guarantees can be allocated in this data store for this data type.

Property	Type	Att ributes	Notes
ProvisionedBytes	integer (By)	*re ad-writ e {=html} (null)*	The value shall be the maximum number of bytes that can be allocated in this data store for this data type.

9.5.3 Identifier

9.5.3.1 Description   This type shall contain any additional identifiers for a resource.

9.5.3.2 Properties   The properties defined for the Identifier schema are summarized in Table 13.

Table 13: Identifier properties

Property	Type	Att ributes	Notes
DurableName *(v1.1+)*	string	*re ad-only (null)*	This property shall contain the world-wide unique identifier for the resource. The string shall be in the format described by the value in the `DurableNameFormat` property.
DurableNameFormat *(v1.1+)*	string (enum)	*r ead-onl y {=html} (null)*	This property shall represent the format of the `DurableName` property. *For the possible property values, see DurableNameFormat in Property details.*

9.5.3.3 Property details

9.5.3.3.1 DurableNameFormat   The defined property values are listed in Table 14.
This property shall represent the format of the `DurableName` property.

Table 14: DurableNameFormat property values

string	Description
EUI	This durable name shall contain the hexadecimal representation of the IEEE-defined 64-bit Extended Unique Identifier (EUI), as defined in the IEEE's Guidelines for 64-bit Global Identifier (EUI-64) Specification. The `DurableName` property shall follow the regular expression pattern `^([ 0-9A-Fa-f]{2}[:-]){7}([0-9A-Fa-f]{2})$`, where the most significant octet is first.
FC_WWN	This durable name shall contain a hexadecimal representation of the World-Wide Name (WWN) format, as defined in the T11 Fibre Channel Physical and Signaling Interface Specification. The `DurableName` property shall follow the regular expression pattern `^([ 0-9A-Fa-f]{2}[:-]){7}([0-9A-Fa-f]{2})$`, where the most significant octet is first.
GCXLID *(v1.15+)*	This durable name shall be in the globally unique CXL logical device identifier (GCXLID). The `DurableName` property shall follow the regular expression pattern `^([0-9A-Fa-f]{2 }-){7}[0-9A-Fa-f]{2}:([0-9A-Fa-f]{4})$`, where the first eight hyphen-delimited octets contain the PCIe serial number, where the most significant octet is first, and the remaining 16-bit field contains the CXL Logical Device Identifier, with the most significant byte first.

string	Description
iQN	This durable name shall be in the iSCSI Qualified Name (iQN) format, as defined in RFC3720 and RFC3721.
MACAddress *(v1.14+)*	This durable name shall be a media access control address (MAC address), which is a unique identifier assigned to a network interface controller (NIC) for use as a network address. This value should not be used if a more specific type of identifier is available. The `DurableName` property shall follow the regular expression pattern `^([ 0-9A-Fa-f]{2}[:-]){5}([0-9A-Fa-f]{2})$`, where the most significant octet is first.
NAA	This durable name shall contain a hexadecimal representation of the Name Address Authority structure, as defined in the T11 Fibre Channel - Framing and Signaling - 3 (FC-FS-3) specification. The `DurableName` property shall follow the regular expression pattern `^(([0-9A-Fa-f]{2}){8}){1,2}$`, where the most significant octet is first.
NGUID *(v1.10+)*	This durable name shall be in the Namespace Globally Unique Identifier (NGUID), as defined in the NVN Express Specification. The `DurableName` property shall follow the regular expression pattern `^([0-9A-Fa-f]{2}){16}$`, where the most significant octet is first.
NQN *(v1.6+)*	This durable name shall be in the NVMe Qualified Name (NQN) format, as defined in the NVN Express over Fabric Specification.

string	Description
NSID *(v1.6+, deprecated v1.12)*	This durable name shall be in the NVM Namespace Identifier (NSID) format, as defined in the NVN Express Specification. *Deprecated in v1.12 and later. This value has been deprecated due to its non-uniqueness and NGUID should be used.*
UUID	This durable name shall contain the hexadecimal representation of the UUID, as defined by RFC4122. The `DurableName` property shall follow the regular expression pattern '([0-9a-f A-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{12})'.

9.5.4 IOStatistics

9.5.4.1 Description   The properties of this type shall be used to represent the IO statistics of the requested object.

9.5.4.2 Properties   The properties defined for the IOStatistics schema are summarized in Table 15.

Table 15: IOStatistics properties

Property	Type	Att ributes	Notes
NonIORequests	integer ({tot})	*re ad-writ e\<br\> {=html} (null)*	The value shall represent the total count from the time of last reset or wrap of non IO requests.

Property	Type	Att ributes	Notes
NonIORequestTime	string	*rea d-write (null)*	The value shall be an ISO 8601 conformant duration describing the time that the resource is busy processing non IO requests from the time of last reset or wrap.
ReadHitIORequests	integer ({tot})	*re ad-writ e {=html} (null)*	The value shall represent the total count from the time of last reset or wrap of read IO requests satisfied from memory.
ReadIOKiBytes	integer (KiBy)	*re ad-writ e {=html} (null)*	The value shall represent the total number of kibibytes read from the time of last reset or wrap.
ReadIORequests	integer ({tot})	*re ad-writ e {=html} (null)*	The value shall represent the total count from the time of last reset or wrap of read IO requests satisfied from either media or memory (i.e. from a storage device or from a cache).
ReadIORequestTime	string	*rea d-write (null)*	The value shall be an ISO 8601 conformant duration describing the time that the resource is busy processing read requests from the time of last reset or wrap.

Property	Type	Att ributes	Notes
WriteHitIORequests	integer ({tot})	*re ad-writ e {=html} (null)*	The value shall represent the total count from the time of last reset or wrap of write IO requests coalesced into memory.
WriteIOKiBytes	integer (KiBy)	*re ad-writ e {=html} (null)*	The value shall represent the total number of kibibytes written from the time of last reset or wrap.
WriteIORequests	integer ({tot})	*re ad-writ e {=html} (null)*	The value shall represent the total count from the time of last reset or wrap of write IO requests.
WriteIORequestTime	string	*rea d-write (null)*	The value shall be an ISO 8601 conformant duration describing the time that the resource is busy processing write requests from the time of last reset or wrap.

9.5.5 IOWorkload

9.5.5.1 Description   This structure may be used to describe an IO Workload.

9.5.5.2 Properties   The properties defined for the IOWorkload schema are summarized in Table 16.

Table 16: IOWorkload properties

Property	Type	Att ributes	Notes
Components [{ }]	array (object)	* (null)*	The value shall be an array of IO workload component descriptions. For property details, see IOWorkloadComponent.
Name	string	*rea d-write (null)*	The value shall be a name of the workload. It should be constructed as OrgID:WorkloadID. Examples: ACME:DSS, ACME:DSS-REP, ACME:Exchange, ACME:OLTP, ACME:OLTP-REPA. An organization may define a set of well known workloads.

9.5.6 IOWorkloadComponent

9.5.6.1 Description   This structure may be used to describe a component of an IO workload.

9.5.6.2 Properties   The properties defined for the IOWorkloadComponent schema are summarized in Table 17.

Table 17: IOWorkloadComponent properties

Property	Type	Att ributes	Notes
AverageIOBytes	integer (By)	*re ad-writ e {=html} (null)*	The value shall be the expected average I/O size.
Duration	string (s)	*re ad-writ e {=html} (null)*	The value of each entry shall be an ISO 8601 duration that shall specify the expected length of time that this component is applied to the workload. This attribute shall be specified if a schedule is specified and otherwise shall not be specified.
IOAccessPattern	string (enum)	*re ad-writ e {=html} (null)*	The enumeration literal shall be the expected access pattern. *For the possible property values, see IOAccessPattern in Property details.*
PercentOfData	integer (%)	*re ad-writ e {=html} (null)*	The value shall be the expected percent of the data referenced by the workload that is covered by this component.
PercentOfIOPS	integer (%)	*re ad-writ e {=html} (null)*	The value shall be the expected percent of the total IOPS for this workload that is covered by this component.

Property	Type	Att ributes	Notes
Schedule {}	object		The value shall specifies when this workload component is applied to the overall workload. For property details, see Schedule v1.2.5).

9.5.6.3 Property details

9.5.6.3.1 IOAccessPattern   The defined property values are listed in Table 18. The enumeration literal shall be the expected access pattern.

Table 18: IOAccessPattern property values

string	Description
RandomReadAgain	Use of this enumeration literal shall indicate an access pattern of random reads of cached data.
RandomReadNew	Use of this enumeration literal shall indicate an access pattern of random reads of uncached data.
ReadWrite	Use of this enumeration literal shall indicate a Uniform distribution of reads and writes.
SequentialRead	Use of this enumeration literal shall indicate a sequential read pattern of access.
SequentialWrite	Use of this enumeration literal shall indicate a sequential write pattern of access.

9.5.7 Location

9.5.7.1 Description   This type shall describe the location of a resource.

9.5.7.2 Properties The properties defined for the Location schema are summarized in Table 19.

Table 19: Location properties

Property	Type	Att ributes	Notes
AltitudeMeters *(v1.6+)*	number (m)	*re ad-writ e {=html} (null)*	This property shall contain the altitude of the resource, in meter units, defined as the elevation above sea level.
Contacts *(v1.7+)* [{	array		This property shall contain an array of contact information for an individual or organization responsible for this resource.
ContactName *(v1.7+)*	string	*rea d-write (null)*	This property shall contain the name of a person or organization to contact for information about this resource.
EmailAddress *(v1.7+)*	string	*rea d-write (null)*	This property shall contain the email address for a person or organization to contact for information about this resource.
PhoneNumber *(v1.7+)*	string	*rea d-write (null)*	This property shall contain the phone number for a person or organization to contact for information about this resource.

}]

Property	Type	Att ributes	Notes
Info *(v1.1+, deprecated v1.5)*	string	*re ad-only (null)*	This property shall represent the location of the resource. *Deprecated in v1.5 and later. This property has been deprecated in favor of the* `PostalAddress,` `Placement,` *and* `PartLocation` *properties.*
InfoFormat *(v1.1+, deprecated v1.5)*	string	*re ad-only (null)*	This property shall represent the `Info` property format. *Deprecated in v1.5 and later. This property has been deprecated in favor of the* `PostalAddress,` `Placement,` *and* `PartLocation` *properties.*
Latitude *(v1.6+)*	number (deg)	*re ad-writ e {=html} (null)*	This property shall contain the latitude of the resource specified in degree units using a decimal format and not minutes or seconds.
Longitude *(v1.6+)*	number (deg)	*re ad-writ e {=html} (null)*	This property shall contain the longitude of the resource specified in degree units using a decimal format and not minutes or seconds.

Property	Type	Att ributes	Notes
Oem *(v1.1+)* {}	object		This property shall contain the OEM extensions. All values for properties contained in this object shall conform to the Redfish S pecification-described requirements. For property details, see Oem.
PartLocation *(v1.5+)* {	object		This property shall contain the part location for a resource within an enclosure. This representation shall indicate the location of a part within a location specified by the `Placement` property.
LocationOrdinalValue *(v1.5+)*	integer	*re ad-only (null)*	This property shall contain the number that represents the location of the part based on the `LocationType`. `Loca- tionOrdinalValue` shall be measured based on the Orientation value starting with `0`.
LocationType *(v1.5+)*	string (enum)	*r ead-onl y {=html} (null)*	This property shall contain the type of location of the part. *For the possible property values, see LocationType in Property details.*

Property	Type	Att ributes	Notes
Orientation *(v1.5+)*	string (enum)	*r ead-onl y {=html} (null)*	This property shall contain the orientation for the ordering used by the `LocationOrdi- nalValue` property. *For the possible property values, see Orientation in Property details.*
Reference *(v1.5+)*	string (enum)	*r ead-onl y {=html} (null)*	This property shall contain the general location within the unit of the part. *For the possible property values, see Reference in Property details.*
ServiceLabel *(v1.5+)*	string	*re ad-only (null)*	This property shall contain the label assigned for service at the part location.

}

Property	Type	Att ributes	Notes
• *PartLocationContext** (v1.16+)	string	re ad-only (null)	This property shall contain a human-readable string to enable differentiation between `PartLocation` values for parts in the same enclosure, which may include hierarchical information of containing `PartLocation` values for the part. The value of this property shall not include values of the `PartLocation` properties for the part itself. The purpose of this value, in conjunction with the `PartLocation` of the part itself, is to allow clients to determine the physical location of the part without tracing through the `PartLocation` of multiple resources.

Property	Type	Att ributes	Notes
PhysicalAddress *(v1.17+)* {	object		This property shall contain a physical address for a resource. This property should be present for resources that represent physical objects that can operate without requiring physical containment by another resource. For example, a server chassis might be contained by a rack, but it might also be deployed individually, while a drive is always contained by a chassis and therefore is described by the containing resource.
City *(v1.17+)*	string	*rea d-write (null)*	This property shall contain the city, township, or shi (JP) location for this resource.
Country *(v1.17+)*	string	*rea d-write (null)*	This property shall contain the country location for this resource.
ISOCountryCode *(v1.17+)*	string	*rea d-write (null)*	This property shall contain the ISO 3166-1-defined alpha-2 or alpha-3 country code. Pattern: `^([A-Z]{2}|[A-Z]{3})$`

Property	Type	Att ributes	Notes
ISOSubdivisionCode (v1.17+)	string	*rea d-write (null)*	This property shall contain the ISO 3166-2-defined state, province, or territory subdivision code for this resource. Pattern: `^[A-Z0-9]{1,3}$`
PostalCode (v1.17+)	string	*rea d-write (null)*	The property shall contain the postal code for this resource. The value shall conform to the RFC5139-defined requirements of the PC field.
StateOrProvince (v1.17+)	string	*rea d-write (null)*	This property shall contain the state, province, or territory location within the country for this resource.
StreetAddress (v1.17+)	string	*rea d-write (null)*	This property shall contain the street-level physical address of the resource, including identifiers such as apartment, room, or building to further locate the resource within a given street address.
}			
Placement (v1.3+) {	object		This property shall contain a place within the addressed location.

Property	Type	Att ributes	Notes
AdditionalInfo *(v1.7+)*	string	*rea d-write (null)*	This property shall contain additional information, such as Tile, Column (Post), Wall, or other designation that describes a location that cannot be conveyed with other properties defined for the Placement object.
Rack *(v1.3+)*	string	*rea d-write (null)*	This property shall contain the name of the rack within a row.
RackOffset *(v1.3+)*	integer	*rea d-write (null)*	The vertical location of the item in the rack. Rack offset units shall be measured from bottom to top, starting with 0.
RackOffsetUnits *(v1.3+)*	string (enum)	*re ad-writ e\<br\> {=html} (null)*	This property shall contain a RackUnit enumeration literal that indicates the type of rack units in use. *For the possible property values, see RackOffsetUnits in Property details.*
Row *(v1.3+)*	string	*rea d-write (null)*	This property shall contain the name of the row.

}

Property	Type	Att ributes	Notes
PostalAddress *(v1.3+, deprecated v1.17)* {	object		This property shall contain a postal address of the resource. *Deprecated in v1.17 and later. This object and its properties have been deprecated in favor of* `PhysicalAddress`.
AdditionalCode *(v1.3+)*	string	*rea d-write (null)*	The value shall conform to the RFC5139-defined requirements of the ADDCODE field.
AdditionalInfo *(v1.7+)*	string	*rea d-write (null)*	The value shall conform to the requirements of the LOC field as defined in RFC5139. Provides additional information.
Building *(v1.3+)*	string	*rea d-write (null)*	The value shall conform to the RFC5139-defined requirements of the BLD field. Names the building.
City *(v1.3+)*	string	*rea d-write (null)*	The value shall conform to the RFC5139-defined requirements of the A3 field. Names a city, township, or shi (JP).
Community *(v1.3+)*	string	*rea d-write (null)*	The value shall conform to the RFC5139-defined requirements of the PCN field. A postal community name.

Property	Type	Att ributes	Notes
Country *(v1.3+)*	string	*rea d-write (null)*	The value shall conform to the RFC5139-defined requirements of the Country field.
District *(v1.3+)*	string	*rea d-write (null)*	The value shall conform to the RFC5139-defined requirements of the A2 field. Names a county, parish, gun (JP), or district (IN).
Division *(v1.3+)*	string	*rea d-write (null)*	The value shall conform to the RFC5139-defined requirements of the A4 field. Names a city division, borough, city district, ward, or chou (JP).
Floor *(v1.3+)*	string	*rea d-write (null)*	The value shall conform to the RFC5139-defined requirements of the FLR field. Provides a floor designation.

Property	Type	Att ributes	Notes
GPSCoords *(v1.3+, deprecated v1.6)*	string	*rea d-write (null)*	The value shall conform to the RFC5139-defined requirements of the ADDCODE field. Shall contain the GPS coordinates of the location. If furnished, expressed in the `[-][nn]n.n nnnnn,` `[-][nn]n.nnnnn` format. For example, two comma-separated positive or negative numbers with six decimal places of precision. *Deprecated in v1.6 and later. This property has been deprecated in favor of the Longitude and Latitude properties.*
HouseNumber *(v1.3+)*	integer	*rea d-write (null)*	The value shall conform to the RFC5139-defined requirements of the HNO field. The numeric portion of the house number.
HouseNumberSuffix *(v1.3+)*	string	*rea d-write (null)*	The value shall conform to the RFC5139-defined requirements of the HNS field. Provides a suffix to a house number, (F, B, or 1/2).

Property	Type	Att ributes	Notes
Landmark *(v1.3+)*	string	*rea d-write (null)*	The value shall conform to the RFC5139-defined requirements of the LMK field. Identifies a landmark or vanity address.
Le adingStreetDirection *(v1.3+)*	string	*rea d-write (null)*	The value shall conform to the requirements of the PRD field as defined in RFC5139. Names a leading street direction, (N, W, or SE).
Location *(v1.3+, deprecated v1.7)*	string	*rea d-write (null)*	The value shall conform to the RFC5139-defined requirements of the LOC field. Provides additional information. *Deprecated in v1.7 and later. This property has been deprecated in favor of the* `AdditionalInfo` *property.*
Name *(v1.3+)*	string	*rea d-write (null)*	The value shall conform to the RFC5139-defined requirements of the NAM field. Names the occupant.
Neighborhood *(v1.3+)*	string	*rea d-write (null)*	The value shall conform to the RFC5139-defined requirements of the A5 field. Names a neighborhood or block.

Property	Type	Att ributes	Notes
PlaceType *(v1.3+)*	string	*rea d-write (null)*	The value shall conform to the RFC5139-defined requirements of the PLC field. Examples include office and residence.
POBox *(v1.3+)*	string	*rea d-write (null)*	The value shall conform to the RFC5139-defined requirements of the POBOX field. A post office box (PO box).
PostalCode *(v1.3+)*	string	*rea d-write (null)*	The value shall conform to the RFC5139-defined requirements of the PC field. A postal code (or zip code).
Road *(v1.3+)*	string	*rea d-write (null)*	The value shall conform to the RFC5139-defined requirements of the RD field. Designates a primary road or street.
RoadBranch *(v1.3+)*	string	*rea d-write (null)*	The value shall conform to the RFC5139-defined requirements of the RDBR field. Shall contain a post office box (PO box) road branch.
RoadPostModifier *(v1.3+)*	string	*rea d-write (null)*	The value shall conform to the RFC5139-defined requirements of the POM field. For example, Extended.

Property	Type	Att ributes	Notes
RoadPreModifier *(v1.3+)*	string	*rea d-write (null)*	The value shall conform to the RFC5139-defined requirements of the PRM field. For example, Old or New.
RoadSection *(v1.3+)*	string	*rea d-write (null)*	The value shall conform to the RFC5139-defined requirements of the RDSEC field. A road section.
RoadSubBranch *(v1.3+)*	string	*rea d-write (null)*	The value shall conform to the RFC5139-defined requirements of the RDSUBBR field.
Room *(v1.3+)*	string	*rea d-write (null)*	The value shall conform to the RFC5139-defined requirements of the ROOM field. A name or number of a room to locate the resource within the unit.
Seat *(v1.3+)*	string	*rea d-write (null)*	The value shall conform to the RFC5139-defined requirements of the SEAT field. A name or number of a seat, such as the desk, cubicle, or workstation.
Street *(v1.3+)*	string	*rea d-write (null)*	The value shall conform to the RFC5139-defined requirements of the A6 field. Names a street.

Property	Type	Att ributes	Notes
StreetSuffix *(v1.3+)*	string	*rea d-write (null)*	The value shall conform to the RFC5139-defined requirements of the STS field. Names a street suffix.
Territory *(v1.3+)*	string	*rea d-write (null)*	The value shall conform to the RFC5139-defined requirements of the A1 field when it names a territory, state, region, province, or prefecture within a country.
TrailingStreetSuffix *(v1.3+)*	string	*rea d-write (null)*	The value shall conform to the RFC5139-defined requirements of the POD field. Names a trailing street suffix.
Unit *(v1.3+)*	string	*rea d-write (null)*	The value shall conform to the RFC5139-defined requirements of the UNIT field. The name or number of a unit, such as the apartment or suite, to locate the resource.

}

9.5.7.3 Property details

9.5.7.3.1 LocationType   The defined property values are listed in Table 20. This property shall contain the type of location of the part.

Table 20: LocationType property values

string	Description
Backplane *(v1.12+)*	This value shall indicate the part is a backplane in an enclosure.
Bay	This value shall indicate the part is located in a bay.
Connector	This value shall indicate the part is located in a connector or port.
Embedded *(v1.13+)*	This value shall indicate the part is embedded or otherwise permanently incorporated into a larger part or device. This value shall not be used for parts that can be removed by a user or are considered field-replaceable.
Slot	This value shall indicate the part is located in a slot.
Socket	This value shall indicate the part is located in a socket.

9.5.7.3.2 Orientation   The defined property values are listed in Table 21. This property shall contain the orientation for the ordering used by the `LocationOrdinalValue` property.

Table 21: Orientation property values

string	Description
BackToFront	This value shall indicate the ordering for `LocationOrdinalValue` is back to front.
BottomToTop	This value shall indicate the ordering for `LocationOrdinalValue` is bottom to top.
FrontToBack	This value shall indicate the ordering for `LocationOrdinalValue` is front to back.

string	Description
LeftToRight	This value shall indicate the ordering for `LocationOrdinalValue` is left to right.
RightToLeft	This value shall indicate the ordering for `LocationOrdinalValue` is right to left.
TopToBottom	This value shall indicate the ordering for `LocationOrdinalValue` is top to bottom.

9.5.7.3.3 RackOffsetUnits   The defined property values are listed in Table 22. This property shall contain a RackUnit enumeration literal that indicates the type of rack units in use.

Table 22: RackOffsetUnits property values

string	Description
EIA_310	Rack units shall conform to the EIA-310 standard.
OpenU	Rack units shall be specified in terms of the Open Compute Open Rack Specification.

9.5.7.3.4 Reference   The defined property values are listed in Table 23. This property shall contain the general location within the unit of the part.

Table 23: Reference property values

string	Description
Bottom	This value shall indicate the part is in the bottom of the unit.
Front	This value shall indicate the part is in the front of the unit.
Left	This value shall indicate the part is on the left side of the unit.

string	Description
Middle	This value shall indicate the part is in the middle of the unit.
Rear	This value shall indicate the part is in the rear of the unit.
Right	This value shall indicate the part is on the right side of the unit.
Top	This value shall indicate the part is in the top of the unit.

9.5.8 Oem

9.5.8.1 Description   This object represents the OEM properties. The resource values shall comply with the Redfish Specification-described requirements.

9.5.8.2 Properties   The properties defined for the Oem schema are summarized in Table 24.

Table 24: Oem properties

Property	Type	Att ributes	Notes
(pattern) {}	object		Property names follow regular expression pattern "^[A-Za-z0-9_]+$"

9.5.9 ReplicaInfo

9.5.9.1 Description   The value shall define the characteristics of a replica.

9.5.9.2 Properties   The properties defined for the ReplicaInfo schema are summarized in Table 25.

Table 25: ReplicaInfo properties

Property	Type	Att ributes	Notes
ConsistencyEnabled	boolean	*re ad-only (null)*	If true, consistency shall be enabled across the source and its associated target replica(s). The default value for this property is false.
ConsistencyState	string (enum)	*r ead-onl y {=html} (null)*	The ConsistencyState enumeration literal shall indicate the current state of consistency. *For the possible property values, see ConsistencyState in Property details.*
ConsistencyStatus	string (enum)	*r ead-onl y {=html} (null)*	The ConsistencyStatus enumeration literal shall specify the current status of consistency. Consistency may have been disabled or is experiencing an error condition. *For the possible property values, see ConsistencyStatus in Property details.*
ConsistencyType	string (enum)	*r ead-onl y {=html} (null)*	The ConsistencyType enumeration literal shall indicate the consistency type used by the source and its associated target group. *For the possible property values, see ConsistencyType in Property details.*

Property	Type	Att ributes	Notes
DataPro tectionLineOfService *(v1.1+)* {	object		The value shall be a pointer to the data protection line of service that describes this replica. See the *DataPr otectionLineOfService* schema for details on this property.
@odata.id	string	*read -write*	Link to a DataP rotectionLineOfService resource. See the Links section and the *DataPr otectionLineOfService* schema for details.
}			
F ailedCopyStopsHostIO	boolean	*re ad-only (null)*	If true, the storage array shall stop receiving data to the source element if copying to a remote element fails. The default value for this property is false.
PercentSynced	integer (%)	*r ead-onl y {=html} (null)*	Specifies the percent of the work completed to reach synchronization. Shall not be instantiated if implementation is not capable of providing this information. If related to a group, then PercentSynced shall be an average of the PercentSynced across all members of the group.

Property	Type	Att ributes	Notes
• *RemoteSourceReplica** (v1.4+)	string	re ad-only (null)	The ReplicaFaultDomain enumeration literal shall describe the fault domain (local or remote) of the replica relationship.
Replica {	object		Deprecated - Use Source Replica. The value shall reference the resource that is the source of this replica.
@odata.id	string (URI)	rea d-only	The value of this property shall be the unique identifier for the resource and it shall be of the form defined in the Redfish specification.
}			
ReplicaFaultDomain (v1.3+)	string (enum)	r ead-onl y {=html} (null)	The ReplicaFaultDomain enumeration literal shall describe the fault domain (local or remote) of the replica relationship. *For the possible property values, see ReplicaFaultDomain in Property details.*

Property	Type	Att ributes	Notes
ReplicaPriority	string (enum)	*r ead-onl y {=html} (null)*	The enumeration literal shall specify the priority of background copy engine I/O to be managed relative to host I/O operations during a sequential background copy operation. *For the possible property values, see ReplicaPriority in Property details.*
R eplicaProgressStatus	string (enum)	*r ead-onl y {=html} (null)*	The ReplicaProgressStatus enumeration literal shall specify the status of the session with respect to Replication activity. *For the possible property values, see ReplicaProgressStatus in Property details.*
R eplicaReadOnlyAccess	string (enum)	*r ead-onl y {=html} (null)*	The enumeration literal shall specify whether the source, the target, or both elements are read only to the host. *For the possible property values, see ReplicaReadOnlyAccess in Property details.*

Property	Type	Att ributes	Notes
• *ReplicaRecoveryMode(*enum)	string (enum)	r ead-onl y {=html} (null)	The enumeration literal shall specify whether the copy operation continues after a broken link is restored. *For the possible property values, see ReplicaRecoveryMode in Property details.*
ReplicaRole	string (enum)	r ead-onl y {=html} (null)	The ReplicaRole enumeration literal shall represent the source or target role of this replica as known to the containing resource. *For the possible property values, see ReplicaRole in Property details.*
ReplicaSkewBytes	integer (By)	r ead-onl y {=html} (null)	Applies to Adaptive mode and it describes maximum number of bytes the SyncedElement (target) can be out of sync. If the number of out-of-sync bytes exceeds the skew value, ReplicaUpdateMode shall be switched to synchronous.

Property	Type	Att ributes	Notes
ReplicaState	string (enum)	*r ead-only {=html} (null)*	The ReplicaState enumeration literal shall specify the state of the relationship with respect to Replication activity. *For the possible property values, see ReplicaState in Property details.*
ReplicaType	string (enum)	*r ead-only {=html} (null)*	The ReplicaType enumeration literal shall describe the intended outcome of the replication. *For the possible property values, see ReplicaType in Property details.*
ReplicaUpdateMode	string (enum)	*r ead-only {=html} (null)*	The enumeration literal shall specify whether the target elements will be updated synchronously or asynchronously. *For the possible property values, see ReplicaUpdateMode in Property details.*

Property	Type	Att ributes	Notes
R equestedReplicaState	string (enum)	*r ead-onl y
 {=html} (null)*	The last requested or desired state for the relationship. The actual state of the relationship shall be represented by ReplicaState. When RequestedState reaches the requested state, this property shall be null. *For the possible property values, see RequestedReplicaState in Property details.*
SourceReplica *(v1.2+)* {	object		The value shall contain the URI to the source replica when located on a different Swordfish service instance.
@odata.id	string (URI)	*rea d-only*	The value of this property shall be the unique identifier for the resource and it shall be of the form defined in the Redfish specification.
}			
SyncMaintained	boolean	*re ad-only (null)*	If true, Synchronization shall be maintained. The default value for this property is false.

Property	Type	Att ributes	Notes
• *UndiscoveredElement*(enum)	string	*r ead-onl y {=html} (null)*	The enumeration literal shall specify whether the source, the target, or both elements involved in a copy operation are undiscovered. An element is considered undiscovered if its object model is not known to the service performing the copy operation. *For the possible property values, see UndiscoveredElement in Property details.*
WhenActivated	string (%)	*r ead-onl y {=html} (null)*	The value shall be an ISO 8601 conformant time of day that specifies when the point-in-time copy was taken or when the replication relationship is activated, reactivated, resumed or re-established. This property shall be null if the implementation is not capable of providing this information.

Property	Type	Att ributes	Notes
WhenDeactivated	string (%)	*r ead-only {=html} (null)*	The value shall be an ISO 8601 conformant time of day that specifies when the replication relationship is deactivated. Do not instantiate this property if implementation is not capable of providing this information.
WhenEstablished	string (%)	*r ead-only {=html} (null)*	The value shall be an ISO 8601 conformant time of day that specifies when the replication relationship is established. Do not instantiate this property if implementation is not capable of providing this information.
WhenSuspended	string (%)	*r ead-only {=html} (null)*	The value shall be an ISO 8601 conformant time of day that specifies when the replication relationship is suspended. Do not instantiate this property if implementation is not capable of providing this information.
WhenSynced	string	*re ad-only (null)*	The value shall be an ISO 8601 conformant time of day that specifies when the elements were synchronized.

Property	Type	Att ributes	Notes
WhenSynchronized	string (%)	*r ead-onl y\<br\> {=html} (null)*	The value shall be an ISO 8601 conformant time of day that specifies when the replication relationship is synchronized. Do not instantiate this property if implementation is not capable of providing this information.

9.5.9.3 Property details

9.5.9.3.1 ConsistencyState　　The defined property values are listed in Table 26. The ConsistencyState enumeration literal shall indicate the current state of consistency.

Table 26: ConsistencyState property values

string	Description
Consistent	This enumeration literal shall indicate that the source and target shall be consistent.
Inconsistent	This enumeration literal shall indicate that the source and target are not required to be consistent.

9.5.9.3.2 ConsistencyStatus　　The defined property values are listed in Table 27. The ConsistencyStatus enumeration literal shall specify the current status of consis- tency. Consistency may have been disabled or is experiencing an error condition.

Table 27: ConsistencyStatus property values

string	Description
Consistent	This enumeration literal shall indicate that the source and target are consistent.
Disabled	This enumeration literal shall indicate that the source and target have consistency disabled.
InError	This enumeration literal shall indicate that the source and target are not consistent.
InProgress	This enumeration literal shall indicate that the source and target are becoming consistent.

9.5.9.3.3 ConsistencyType   The defined property values are listed in Table 28. The ConsistencyType enumeration literal shall indicate the consistency type used by the source and its associated target group.

Table 28: ConsistencyType property values

string	Description
SequentiallyConsistent	This enumeration literal shall indicate that the source and target shall be sequentially consistent.

9.5.9.3.4 ReplicaFaultDomain   The defined property values are listed in Table 29. The ReplicaFaultDomain enumeration literal shall describe the fault domain (local or remote) of the replica relationship.

Table 29: ReplicaFaultDomain property values

string	Description
Local	This enumeration literal shall indicate that the source and target replicas are contained within a single fault domain.
Remote	This enumeration literal shall indicate that the source and target replicas are in separate fault domains.

9.5.9.3.5 ReplicaPriority   The defined property values are listed in Table 30. The enumeration literal shall specify the priority of background copy engine I/O to be managed relative to host I/O operations during a sequential background copy operation.

Table 30: ReplicaPriority property values

string	Description
High	Copy engine I/O shall have a higher priority than host I/O.
Low	Copy engine I/O shall have a lower priority than host I/O.
Same	Copy engine I/O shall have the same priority as host I/O.
Urgent	Regardless of the host I/O requests, the Copy operation shall be performed as soon as possible.

9.5.9.3.6 ReplicaProgressStatus   The defined property values are listed in Table 31. The ReplicaProgressStatus enumeration literal shall specify the status of the session with respect to Replication activity.

Table 31: ReplicaProgressStatus property values

string	Description
Aborting	This enumeration literal shall indicate that replication has an abort in progress.
Completed	This enumeration literal shall indicate that the request is completed. Data flow is idle.
Detaching	This enumeration literal shall indicate that replication has a detach in progress.
Dormant	This enumeration literal shall indicate that the data flow is inactive, suspended or quiesced.
FailingBack	This enumeration literal shall indicate that replication is undoing the result of failover.
FailingOver	This enumeration literal shall indicate that replication is in the process of switching source and target.
Fracturing	This enumeration literal shall indicate that replication has a fracture in progress.
Initializing	This enumeration literal shall indicate that replication is in the process of establishing source/replica relationship and the data flow has not started.
Mixed	This enumeration literal shall indicate that replication status is mixed across element pairs in a replication group. Generally, the individual statuses need to be examined.
Pending	This enumeration literal shall indicate that the flow of data has stopped momentarily due to limited bandwidth or a busy system.
Preparing	This enumeration literal shall indicate that replication has preparation in progress.

string	Description
RequiresActivate	This enumeration literal shall indicate that the requested operation has completed, however, the synchronization relationship needs to be activated before further copy operations can be issued.
RequiresDetach	This enumeration literal shall indicate that the requested operation has completed, however, the synchronization relationship needs to be detached before further copy operations can be issued.
RequiresFracture	This enumeration literal shall indicate that the requested operation has completed, however, the synchronization relationship needs to be fractured before further copy operations can be issued.
RequiresResume	This enumeration literal shall indicate that the requested operation has completed, however, the synchronization relationship needs to be resumed before further copy operations can be issued.
RequiresResync	This enumeration literal shall indicate that the requested operation has completed, however, the synchronization relationship needs to be resynced before further copy operations can be issued.
RequiresSplit	This enumeration literal shall indicate that the requested operation has completed, however, the synchronization relationship needs to be split before further copy operations can be issued.
Restoring	This enumeration literal shall indicate that replication has a restore in progress.
Resyncing	This enumeration literal shall indicate that replication has resynchronization in progress.

string	Description
Splitting	This enumeration literal shall indicate that replication has a split in progress.
Suspending	This enumeration literal shall indicate that replication has a copy operation in the process of being suspended.
Synchronizing	This enumeration literal shall indicate that replication has synchronization in progress.
Terminating	This enumeration literal shall indicate that the replication relationship is in the process of terminating.

9.5.9.3.7 ReplicaReadOnlyAccess   The defined property values are listed in Table 32. The enumeration literal shall specify whether the source, the target, or both elements are read only to the host.

Table 32: ReplicaReadOnlyAccess property values

string	Description
Both	Both the source and the target elements shall be read only to the host.
ReplicaElement	The replica element shall be read-only to the host.
SourceElement	The source element shall be read-only to the host.

9.5.9.3.8 ReplicaRecoveryMode   The defined property values are listed in Table 33. The enumeration literal shall specify whether the copy operation continues after a broken link is restored.

Table 33: ReplicaRecoveryMode property values

string	Description
Automatic	The copy operation shall resume automatically.
Manual	The ReplicaState shall be set to Suspended after the link is restored. It is required to issue the Resume operation to continue.

9.5.9.3.9 ReplicaRole   The defined property values are listed in Table 34. The ReplicaRole enumeration literal shall represent the source or target role of this replica as known to the containing resource.

Table 34: ReplicaRole property values

string	Description
Source	This enumeration literal shall indicate a source element.
Target	This enumeration literal shall indicate target element.

9.5.9.3.10 ReplicaState   The defined property values are listed in Table 35. The ReplicaState enumeration literal shall specify the state of the relationship with respect to Replication activity.

Table 35: ReplicaState property values

string	Description
Aborted	This enumeration literal shall indicate that the copy operation is aborted with the Abort operation. The Resync Replica operation can be used to restart the copy operation.
Broken	This enumeration literal shall indicate that the relationship is non-functional due to errors in the source, the target, the path between the two or space constraints.

string	Description
Failedover	This enumeration literal shall indicate that the reads and writes are sent to the target element. The source element may not be reachable.
Fractured	This enumeration literal shall indicate that the Target is split from the source. The target may not be consistent.
Inactive	This enumeration literal shall indicate that data flow has stopped, writes to source element shall not be sent to target element.
Initialized	This enumeration literal shall indicate that the link to enable replication is established and source/replica elements are associated, but the data flow has not started.
Invalid	This enumeration literal shall indicate that the storage server is unable to determine the state of the replication relationship, for example, after the connection is restored; however, either source or target elements have an unknown status.
Mixed	This enumeration literal shall indicate the ReplicaState of GroupSynchronized. The value indicates the StorageSynchronized relationships of the elements in the group have different ReplicaState values.
Partitioned	This enumeration literal shall indicate that the state of replication relationship can not be determined, for example, due to a connection problem.
Prepared	This enumeration literal shall indicate that initialization is completed, however, the data flow has not started.
Restored	This enumeration literal shall indicate that the source element was restored from the target element.

string	Description
Skewed	This enumeration literal shall indicate that the target has been modified and is no longer synchronized with the source element or the point-in-time view.
Split	This enumeration literal shall indicate that the target element was gracefully (or systematically) split from its source element – consistency shall be guaranteed.
Suspended	This enumeration literal shall indicate that the data flow between the source and target elements has stopped. Writes to source element shall be held until the relationship is Resumed.
Synchronized	This enumeration literal shall indicate that for Mirror, Snapshot, or Clone replication, the target represents a copy of the source.
Unsynchronized	This enumeration literal shall indicate that not all the source element data has been copied to the target element.

9.5.9.3.11 ReplicaType   The defined property values are listed in Table 36. The ReplicaType enumeration literal shall describe the intended outcome of the replication.

Table 36: ReplicaType property values

string	Description
Clone	This enumeration literal shall indicate that replication shall create a point in time, full copy the source.
Mirror	This enumeration literal shall indicate that replication shall create and maintain a copy of the source.

string	Description
Snapshot	This enumeration literal shall indicate that replication shall create a point in time, virtual copy of the source.
TokenizedClone	This enumeration literal shall indicate that replication shall create a token based clone.

9.5.9.3.12 ReplicaUpdateMode   The defined property values are listed in Table 37. The enumeration literal shall specify whether the target elements will be updated synchronously or asynchronously.

Table 37: ReplicaUpdateMode property values

string	Description
Active	This enumeration literal shall indicate Active-Active (i.e. bidirectional) synchronous updates.
Adaptive	This enumeration literal shall indicate that an implementation may switch between synchronous and asynchronous modes.
Asynchronous	This enumeration literal shall indicate Asynchronous updates.
Synchronous	This enumeration literal shall indicate Synchronous updates.

9.5.9.3.13 RequestedReplicaState   The defined property values are listed in Table 38. The last requested or desired state for the relationship. The actual state of the relationship shall be represented by ReplicaState. When RequestedState reaches the requested state, this property shall be null.

Table 38: RequestedReplicaState property values

string	Description
Aborted	This enumeration literal shall indicate that the copy operation is aborted with the Abort operation. The Resync Replica operation can be used to restart the copy operation.
Broken	This enumeration literal shall indicate that the relationship is non-functional due to errors in the source, the target, the path between the two or space constraints.
Failedover	This enumeration literal shall indicate that the reads and writes are sent to the target element. The source element may not be reachable.
Fractured	This enumeration literal shall indicate that the Target is split from the source. The target may not be consistent.
Inactive	This enumeration literal shall indicate that data flow has stopped, writes to source element shall not be sent to target element.
Initialized	This enumeration literal shall indicate that the link to enable replication is established and source/replica elements are associated, but the data flow has not started.
Invalid	This enumeration literal shall indicate that the storage server is unable to determine the state of the replication relationship, for example, after the connection is restored; however, either source or target elements have an unknown status.
Mixed	This enumeration literal shall indicate the ReplicaState of GroupSynchronized. The value indicates the StorageSynchronized relationships of the elements in the group have different ReplicaState values.

string	Description
Partitioned	This enumeration literal shall indicate that the state of replication relationship can not be determined, for example, due to a connection problem.
Prepared	This enumeration literal shall indicate that initialization is completed, however, the data flow has not started.
Restored	This enumeration literal shall indicate that the source element was restored from the target element.
Skewed	This enumeration literal shall indicate that the target has been modified and is no longer synchronized with the source element or the point-in-time view.
Split	This enumeration literal shall indicate that the target element was gracefully (or systematically) split from its source element – consistency shall be guaranteed.
Suspended	This enumeration literal shall indicate that the data flow between the source and target elements has stopped. Writes to source element shall be held until the relationship is Resumed.
Synchronized	This enumeration literal shall indicate that for Mirror, Snapshot, or Clone replication, the target represents a copy of the source.
Unsynchronized	This enumeration literal shall indicate that not all the source element data has been copied to the target element.

9.5.9.3.14 UndiscoveredElement   The defined property values are listed in Table 39. The enumeration literal shall specify whether the source, the target, or both elements involved in a copy operation are undiscovered. An element is considered undiscovered if its object model is not known to the service performing the copy op-

eration.

Table 39: UndiscoveredElement property values

string	Description
ReplicaElement	This enumeration literal shall indicate that the replica element is undiscovered.
SourceElement	This enumeration literal shall indicate that the source element is undiscovered.

9.5.10 ReplicaRequest

9.5.10.1 Description   A ReplicaRequest shall contain information about the ReplicaSource and the ReplicaName.

9.5.10.2 Properties   The properties defined for the ReplicaRequest schema are summarized in Table 40.

Table 40: ReplicaRequest properties

Property	Type	Att ributes	Notes
ReplicaName *(v1.1+)*	string	*rea d-write (null)*	The value shall be the names of the replica.
ReplicaSource *(v1.1+)* {	object		The value shall reference a resource to be replicated.
@odata.id	string (URI)	*rea d-only*	The value of this property shall be the unique identifier for the resource and it shall be of the form defined in the Redfish specification.
}			

9.5.11 Schedule

9.5.11.1 Description   The properties of this type shall schedule a series of occurrences.

9.5.11.2 Properties   The properties defined for the Schedule schema are summarized in Table 41.

Table 41: Schedule properties

Property	Type	Att ributes	Notes
@odata.id	string	*rea d-only*	Link to another Schedule resource.

9.5.12 Status

9.5.12.1 Description   This type shall contain any status or health properties of a resource.

9.5.12.2 Properties   The properties defined for the Status schema are summarized in Table 42.

Table 42: Status properties

Property	Type	Att ributes	Notes
Conditions *(v1.11+)* [{	array		This property shall represent the active conditions requiring attention in this or a related resource that affects the `Health` or `HealthRollup` of this resource. The service may roll up multiple conditions originating from a resource, using the `Condi tionInRelat- edResource` message from the Base Message Registry. The array order of conditions may change as new conditions occur or as conditions are resolved by the service.
LogEntry {	object		This property shall contain a link to a resource of type `LogEntry` that represents the log entry created for this condition.
@odata.id	string (URI)	*rea d-only*	The value of this property shall be the unique identifier for the resource and it shall be of the form defined in the Redfish specification.
}			

Property	Type	Att ributes	Notes
Message	string	*rea d-only*	This property shall contain a human-readable message describing this condition.
MessageArgs [ ]	array (string)	*rea d-only*	This property shall contain an array of message arguments that are substituted for the arguments in the message when looked up in the message registry. It has the same semantics as the `MessageArgs` property in the Redfish `MessageRegistry` schema.
MessageId	string	*re ad-only* *re quired*	This property shall contain a `MessageId`, as defined in the 'MessageId format' clause of the Redfish Specification.
OriginOfCondition {	object		This property shall contain a link to the resource or object that originated the condition. This property shall not be present if the condition was caused by this resource.

Property	Type	Att ributes	Notes
@odata.id	string (URI)	*rea d-only*	The value of this property shall be the unique identifier for the resource and it shall be of the form defined in the Redfish specification.
}			
Resolution *(v1.14+)*	string	*rea d-only*	This property shall contain the resolution of the condition. Services should replace the resolution defined in the message registry with a more specific resolution.
ResolutionSteps *(v1.18+)* [{}]	array (object)		This property shall contain an array of recommended steps to resolve the condition. A client can stop executing the resolution steps once the condition is removed from the resource. For property details, see ResolutionStep v1.0.1).
Severity	string (enum)	*rea d-only*	This property shall contain the severity of the condition. Services can replace the value defined in the message registry with a value more applicable to the implementation. *For the possible property values, see Severity in Property details.*

Property	Type	Att ributes	Notes
Timestamp	string (da te-time)	*rea d-only*	This property shall indicate the time the condition occurred.
}]			
Health	string (enum)	*r ead-onl y {=html} (null)*	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. *For the possible property values, see Health in Property details.*
HealthRollup	string (enum)	*r ead-onl y {=html} (null)*	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. For additional property requirements, see the corresponding definition in the Redfish Data Model Specification. *For the possible property values, see HealthRollup in Property details.*

Property	Type	Att ributes	Notes
Oem {	object		This property shall contain the OEM extensions. All values for properties contained in this object shall conform to the Redfish S pecification-described requirements.
(pattern) {} }	object		Property names follow regular expression pattern "^[A-Za-z0-9_]+$"
State	string (enum)	*r ead-onl y {=html} (null)*	This property shall indicate the state of the resource. *For the possible property values, see State in Property details.*

9.5.12.3 Property details

9.5.12.3.1 Health   The defined property values are listed in Table 43. 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.

Table 43: Health property values

string	Description
Critical	A critical condition requires immediate attention.
OK	Normal.
Warning	A condition requires attention.

9.5.12.3.2 HealthRollup The defined property values are listed in Table 44. 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. For additional property requirements, see the corresponding definition in the Redfish Data Model Specification.

Table 44: HealthRollup property values

string	Description
Critical	A critical condition requires immediate attention.
OK	Normal.
Warning	A condition requires attention.

9.5.12.3.3 Severity The defined property values are listed in Table 45. This property shall contain the severity of the condition. Services can replace the value defined in the message registry with a value more applicable to the implementation.

Table 45: Severity property values

string	Description
Critical	A critical condition requires immediate attention.
OK	Normal.
Warning	A condition requires attention.

9.5.12.3.4 State The defined property values are listed in Table 46. This property shall indicate the state of the resource.

Table 46: State property values

string	Description
Absent	This value shall indicate that the function or device is absent as defined in the 'Absent resources' clause of the Redfish Specification.

string	Description
Deferring *(v1.2+)*	This value shall indicate the element does not process any commands but queues new requests.
Degraded *(v1.19+)*	This value shall indicate the resource is enabled but operating in a degraded mode.
Disabled	This value shall indicate that a function or resource is unavailable.
Enabled	This value shall indicate that a function or resource is capable of operating.
InTest	This value shall indicate that the component is undergoing testing or is in the process of capturing information for debugging.
Qualified *(v1.9+, deprecated v1.19)*	This value shall indicate the element is within the acceptable range of operation. *Deprecated in v1.19 and later. This value has been deprecated in favor of StandbySpare.*
Quiesced *(v1.2+)*	This value shall indicate the element is enabled but only processes a restricted set of commands.
StandbyOffline	This value shall indicate that a resource is temporarily unavailable but may become available through an external action.
StandbySpare	This value shall indicate that a resource is unavailable but may become available automatically as part of a failover, through an external action, or in response to the change in state of another device or resource.
Starting	This value shall indicate that the resource is attempting to transition to `Enabled`.
UnavailableOffline *(v1.1+)*	This value shall indicate that a function or resource is present but not able to be used.

string	Description
Updating *(v1.2+)*	This value shall indicate the element is updating. The element may become unavailable or operate at a degraded level of performance or functionality.

9.6 Swordfish Schema Types

9.6.1 CapacitySource 1.2.1

9.6.1.1 Description   This composition may be used to represent the source and type of storage capacity. At most one of the ProvidingDrives, ProvidingVolumes, ProvidingMemoryChunks, ProvidingMemory or ProvidingPools properties may have a value. If any of ProvidingDrives, ProvidingVolumes, ProvidingMemory or ProvidingPools reference more than one resource, allocation of capacity across those resources is implementation dependent.

9.6.1.2 URIs   /redfish/v1/Storage/*{StorageId}*/FileSystems/*{FileSystemId}*/CapacitySources/*{CapacitySourceId}*   /redfish/v1/Storage/*{StorageId}*/StoragePools/*{StoragePoolId}*/AllocatedVolumes/*{VolumeId}*/CapacitySources/*{CapacitySourceId}* /redfish/v1/Storage/*{StorageId}*/StoragePools/*{StoragePoolId}*/CapacitySources/*{CapacitySourceId}*   /redfish/v1/Storage/*{StorageId}*/Volumes/*{VolumeId}*/CapacitySources/*{CapacitySourceId}*   /redfish/v1/StorageServices/*{StorageServiceId}*/FileSystems/*{FileSystemId}*/CapacitySources/*{CapacitySourceId}*   /redfish/v1/StorageServices/*{StorageServiceId}*/StoragePools/*{StoragePoolId}*/CapacitySources/*{CapacitySourceId}*   /redfish/v1/StorageServices/*{StorageServiceId}*/Volumes/*{VolumeId}*/CapacitySources/*{CapacitySourceId}*   /redfish/v1/Systems/*{ComputerSystemId}*/Storage/*{StorageId}*/FileSystems/*{FileSystemId}*/CapacitySources/*{CapacitySourceId}*   /redfish/v1/Systems/*{ComputerSystemId}*/Storage/*{StorageId}*/StoragePools/*{StoragePoolId}*/CapacitySources/*{CapacitySourceId}*   /redfish/v1/Systems/*{ComputerSystemId}*/Storage/*{StorageId}*/Volumes/*{VolumeId}*/CapacitySources/*{CapacitySourceId}*

9.6.1.3 Properties   The properties defined for the CapacitySource 1.2.1 schema are summarized in Table 47.

Table 47: CapacitySource 1.2.1 properties

Property	Type	Att ributes	Notes
Actions *(v1.1.2+)* {}	object		The Actions property shall contain the available actions for this resource.
Description	string	*re ad-only (null)*	This property shall contain the description of this resource. The value shall conform with the 'Description' clause of the Redfish Specification.
Id	string	*re ad-only re quired*	This property shall contain the identifier for this resource. The value shall conform with the 'Id' clause of the Redfish Specification.
Name	string	*re ad-only re quired*	This property shall contain the name of this resource or array member. The value shall conform with the 'Name' clause of the Redfish Specification.
Oem {}	object		This property shall contain the OEM extensions. All values for properties that this object contains shall conform to the Redfish S pecification-described requirements. For property details, see Oem.

Property	Type	Att ributes	Notes
ProvidedCapacity {}	object		The value shall be the amount of space that has been provided from the ProvidingDrives, ProvidingVolumes, ProvidingMemory or ProvidingPools. For property details, see Capacity.
Pr ovidedClassOfService {	object		The value shall reference the provided ClassOfService from the ProvidingDrives, ProvidingVolumes, ProvidingMemoryChunks, ProvidingMemory or ProvidingPools. See the *ClassOfService* schema for details on this property.
@odata.id	string	*rea d-only*	Link to a ClassOfService resource. See the Links section and the *ClassOfService* schema for details.
}			
ProvidingDrives {	object		If present, the value shall be a reference to a contributing drive or drives.
@odata.id	string (URI)	*rea d-only*	The value of this property shall be the unique identifier for the resource and it shall be of the form defined in the Redfish specification.

Property	Type	Att ributes	Notes
}			
ProvidingMemory *(v1.1+)* {	object		If present, the value shall be a reference to the contributing memory.
@odata.id	string (URI)	*rea d-only*	The value of this property shall be the unique identifier for the resource and it shall be of the form defined in the Redfish specification.
}			
P rovidingMemoryChunks *(v1.1+)* {	object		If present, the value shall be a reference to the contributing memory chunks.
@odata.id	string (URI)	*rea d-only*	The value of this property shall be the unique identifier for the resource and it shall be of the form defined in the Redfish specification.
}			
ProvidingPools {	object		If present, the value shall be a reference to a contributing storage pool or storage pools. Contains a link to a resource.
@odata.id	string	*rea d-only*	Link to Collection of *StoragePool*. See the StoragePool schema for details.
}			

Property	Type	Att ributes	Notes
ProvidingVolumes {	object		If present, the value shall be a reference to a contributing volume or volumes. Contains a link to a resource.
@odata.id	string	*rea d-only*	Link to Collection of *Volume*. See the Volume schema for details.
}			

9.6.2 CapacitySourceCollection

9.6.2.1 URIs   /redfish/v1/Storage/{*StorageId*}/FileSystems/{*FileSystemId*}/CapacitySources   /redfish/v1/Storage/{*StorageId*}/StoragePools/{*StoragePoolId*}/CapacitySources   /redfish/v1/Storage/{*StorageId*}/Volumes/{*VolumeId*}/CapacitySources   / redfish/v1/StorageServices/{*StorageServiceId*}/FileSystems/{*FileSystemId*}/CapacitySources   /redfish/v1/StorageServices/{*StorageServiceId*}/StoragePools/{*StoragePoolId*}/CapacitySources   /redfish/v1/StorageServices/{*StorageServiceId*}/Volumes/ {*VolumeId*}/CapacitySources   /redfish/v1/Systems/{*ComputerSystemId*}/Storage/ {*StorageId*}/FileSystems/{*FileSystemId*}/CapacitySources /redfish/v1/Systems/{*ComputerSystemId*}/Storage/{*StorageId*}/StoragePools/{*StoragePoolId*}/CapacitySources /redfish/v1/Systems/{*ComputerSystemId*}/Storage/{*StorageId*}/Volumes/{*VolumeId*}/ CapacitySources

9.6.2.2 Properties   The properties defined for the CapacitySourceCollection schema are summarized in Table 48.

Table 48: CapacitySourceCollection properties

Property	Type	Att ributes	Notes
Description	string	*re ad-only (null)*	This property shall contain the description of this resource. The value shall conform with the 'Description' clause of the Redfish Specification.
Members [{	array		The value of each member entry shall reference a CapacitySource resource.
@odata.id	string	*rea d-only*	Link to a CapacitySource resource. See the Links section and the *CapacitySource* schema for details.
}]			
Me mbers@odata.nextLink	string (URI)	*rea d-only*	The value of this property shall be a URI to a resource, with the same @odata.type, containing the next set of partial members.
Name	string	*rea d-only*	This property shall contain the name of this resource or array member. The value shall conform with the 'Name' clause of the Redfish Specification.

Property	Type	Att ributes	Notes
Oem {}	object		This property shall contain the OEM extensions. All values for properties contained in this object shall conform to the Redfish S pecification-described requirements. For property details, see Oem.

9.6.3 ClassOfService 1.2.0

9.6.3.1 Description   This resource shall define a service option composed of one or more line of service entities. ITIL defines a service option as a choice of utility or warranty for a service.

9.6.3.2 URIs   /redfish/v1/StorageServices/{*StorageServiceId*}/ClassesOfService/ {*ClassOfServiceId*}   /redfish/v1/StorageServices/{*StorageServiceId*}/StoragePools/ {*StoragePoolId*}/ClassesOfService/{*ClassOfServiceId*}

9.6.3.3 Properties   The properties defined for the ClassOfService 1.2.0 schema are summarized in Table 49.

Table 49: ClassOfService 1.2.0 properties

Property	Type	Att ributes	Notes
Actions *(v1.1+)* {}	object		The Actions property shall contain the available actions for this resource.

Property	Type	Att ributes	Notes
C lassOfServiceVersion	string	*rea d-write (null)*	The version describing the creation or last modification of this service option specification. The string representing the version shall be in the form: M + '.' + N + '.' + U Where: M - The major version (in numeric form). N - The minor version (in numeric form). U - The update (e.g. errata or patch in numeric form).
DataProt ectionLinesOfService *(v1.1.1+)* [{	array		The value shall be a set of data protection service options. Within a class of service, one data protection service option shall be present for each replication session.
@odata.id	string	*read -write*	Link to a DataP rotectionLineOfService resource. See the Links section and the *DataPr otectionLineOfService* schema for details.
}]			
DataSe curityLinesOfService *(v1.1.1+)* [{	array		The value shall be a set of data security service options.

Property	Type	Att ributes	Notes
@odata.id	string	*read -write*	Link to a Dat aSecurityLineOfService resource. See the Links section and the *Data SecurityLineOfService* schema for details.
}]			
DataS torageLinesOfService *(v1.1.1+)* [{	array		The value shall be a set of data protection service options.
@odata.id	string	*read -write*	Link to a Da taStorageLineOfService resource. See the Links section and the *Dat aStorageLineOfService* schema for details.
}]			
Description	string	*re ad-only (null)*	This property shall contain the description of this resource. The value shall conform with the 'Description' clause of the Redfish Specification.
Id	string	*re ad-only re quired*	This property shall contain the identifier for this resource. The value shall conform with the 'Id' clause of the Redfish Specification.
Identifier {}	object		The value shall be unique within the managed ecosystem. For property details, see Identifier v1.19.1).

Property	Type	Att ributes	Notes
IOConnec tivityLinesOfService *(v1.1.1+)* [{	array		The value shall be a set of IO connectivity service options. Within a class of service, at most one IO connectivity service option may be present for a value of AccessProtocol.
@odata.id	string	*read -write*	Link to a IOCon nectivityLineOfService resource. See the Links section and the *IOConn ectivityLineOfService* schema for details.
}]			
IOPerfo rmanceLinesOfService *(v1.1.1+)* [{	array		The value shall be a set of IO performance service options.
@odata.id	string	*read -write*	Link to a IOPe rformanceLineOfService resource. See the Links section and the *IOPer formanceLineOfService* schema for details.
}]			
Name	string	*re ad-only re quired*	This property shall contain the name of this resource or array member. The value shall conform with the 'Name' clause of the Redfish Specification.

Property	Type	Att ributes	Notes
Oem {}	object		This property shall contain the OEM extensions. All values for properties that this object contains shall conform to the Redfish S pecification-described requirements. For property details, see Oem.

9.6.4 ClassOfServiceCollection

9.6.4.1 URIs   /redfish/v1/StorageServices/{*StorageServiceId*}/ClassesOfService
/redfish/v1/StorageServices/{*StorageServiceId*}/StoragePools/{*StoragePoolId*}/
ClassesOfService

9.6.4.2 Properties   The properties defined for the ClassOfServiceCollection schema are summarized in Table 50.

Table 50: ClassOfServiceCollection properties

Property	Type	Att ributes	Notes
Description	string	*re ad-only (null)*	This property shall contain the description of this resource. The value shall conform with the 'Description' clause of the Redfish Specification.
Members [{	array		The value of each member entry shall reference a ClassOfService or LineOfService resource.

Property	Type	Att ributes	Notes
@odata.id	string	*rea d-only*	Link to a LineOfService resource. See the Links section and the *LineOfService* schema for details.
}]			
Me mbers@odata.nextLink	string (URI)	*rea d-only*	The value of this property shall be a URI to a resource, with the same @odata.type, containing the next set of partial members.
Name	string	*rea d-only*	This property shall contain the name of this resource or array member. The value shall conform with the 'Name' clause of the Redfish Specification.
Oem {}	object		This property shall contain the OEM extensions. All values for properties contained in this object shall conform to the Redfish S pecification-described requirements. For property details, see Oem.

9.6.5 ConsistencyGroup 1.1.1

9.6.5.1 Description   A collection of volumes grouped together to ensure write order consistency across all those volumes. A management operation on a consistency group, such as configuring replication properties, applies to all the volumes within

the consistency group.

9.6.5.2 URIs /redfish/v1/Storage/*{StorageId}*/ConsistencyGroups/*{Consisten-cyGroupId}* /redfish/v1/StorageServices/*{StorageServiceId}*/ConsistencyGroups/ *{ConsistencyGroupId}* /redfish/v1/StorageServices/*{StorageServiceId}*/Volumes/ *{VolumeId}*/ConsistencyGroups/*{ConsistencyGroupId}* /redfish/v1/Systems/*{Comput-erSystemId}*/Storage/*{StorageId}*/ConsistencyGroups/*{ConsistencyGroupId}*

9.6.5.3 Properties The properties defined for the ConsistencyGroup 1.1.1 schema are summarized in Table 51.

Table 51: ConsistencyGroup 1.1.1 properties

Property	Type	Att ributes	Notes
Actions {	object		The Actions property shall contain the available actions for this resource.
#ConsistencyGroup .AssignReplicaTarget {}	object		This action shall be used to establish a replication relationship by assigning an existing consistency group to serve as a target replica for an existing source consistency group. *For more information, see the Actions section below.*

Property	Type	Att ributes	Notes
#ConsistencyGroup .CreateReplicaTarget {}	object		This action shall be used to create a new consistency group resource to provide expanded data protection through a replica relationship with the specified source consistency group. *For more information, see the Actions section below.*
# Consistency- Group.Remov eReplicaRelationship {}	object		This action shall be used to disable data synchronization between a source and target consistency group, remove the replication relationship, and optionally delete the target consistency group. *For more information, see the Actions section below.*
#ConsistencyGro up.ResumeReplication {}	object		This action shall be used to resume the active data synchronization between a source and target consistency group, without otherwise altering the replication relationship. *For more information, see the Actions section below.*

Property	Type	Att ributes	Notes
#Consi stency-Group.ReverseRep licationRelationship {}	object		This action shall be used to reverse the replication relationship between a source and target consistency group. *For more information, see the Actions section below.*
#ConsistencyGr oup.SplitReplication {}	object		This action shall be used to split the replication relationship and suspend data synchronization between a source and target consistency group. *For more information, see the Actions section below.*
#ConsistencyGrou p.SuspendReplication {}	object		This action shall be used to suspend active data synchronization between a source and target consistency group, without otherwise altering the replication relationship. *For more information, see the Actions section below.*
}			
ConsistencyMethod	string (enum)	*re ad-write\<br\> {=html} (null)*	The property shall set the consistency method used by this group. *For the possible property values, see ConsistencyMethod in Property details.*

Property	Type	Att ributes	Notes
ConsistencyType	string (enum)	*re ad-writ e\<br\> {=html} (null)*	This property shall set the consistency type used by this group. *For the possible property values, see ConsistencyType in Property details.*
Description	string	*re ad-only (null)*	This property shall contain the description of this resource. The value shall conform with the 'Description' clause of the Redfish Specification.
Id	string	*re ad-only re quired*	This property shall contain the identifier for this resource. The value shall conform with the 'Id' clause of the Redfish Specification.
IsConsistent	boolean	*re ad-only (null)*	The value of this property shall be set to true when the consistency group is in a consistent state.
Links {	object		This property shall contain links to other resources that are related to this resource.

Property	Type	Att ributes	Notes
Oem {}	object		This property shall contain the OEM extensions. All values for properties contained in this object shall conform to the Redfish S pecification-described requirements. For property details, see Oem.
}			
Name	string	*re ad-only re quired*	This property shall contain the name of this resource or array member. The value shall conform with the 'Name' clause of the Redfish Specification.
Oem {}	object		This property shall contain the OEM extensions. All values for properties that this object contains shall conform to the Redfish S pecification-described requirements. For property details, see Oem.
RemoteReplicaTargets *(v1.1+)* [ ]	array (string, null)	*rea d-only*	The value shall reference the URIs to the remote target replicas that are sourced by this replica. Remote indicates that the replica is managed by a separate Swordfish service instance.

Property	Type	Att ributes	Notes
ReplicaInfo {}	object		This property shall describe the replication relationship between this storage group and a corresponding source storage group. For property details, see ReplicaInfo v1.4.0).
ReplicaTargets [{	array		The value shall reference the target replicas that are sourced by this replica.
@odata.id	string (URI)	*rea d-only*	The value of this property shall be the unique identifier for the resource and it shall be of the form defined in the Redfish specification.
}]			
Status {}	object		The property shall contain the status of the ConsistencyGroup. For property details, see Status.
Volumes [{	array		An array of references to volumes managed by this storage group.
@odata.id	string	*read -write*	Link to a Volume resource. See the Links section and the *Volume* schema for details.
}]			

9.6.5.4 Actions

9.6.5.4.1 AssignReplicaTarget   Description

This action shall be used to establish a replication relationship by assigning an existing consistency group to serve as a target replica for an existing source consistency group.

Action URI

{Base URI of target resource}/Actions/ConsistencyGroup.AssignReplicaTarget

Action parameters

The parameters for the action which are included in the POST body to the URI shown in the 'target' property of the Action are summarized in Table 52.

Table 52: AssignReplicaTarget action parameters

Parameter Name	Type	Attributes	Notes
ReplicaType	string (enum)	*required*	This parameter shall contain the type of replica relationship to be created. *For the possible property values, see ReplicaType in Property details.*
R eplicaUpdateMode	string (enum)	*required*	This parameter shall specify the replica update mode. *For the possible property values, see Rep licaUpdateMode in Property details.*

Parameter Name	Type	Attributes	Notes
Target ConsistencyGroup	string	*required*	This parameter shall contain the Uri to the existing consistency group.

9.6.5.4.2 CreateReplicaTarget   Description

This action shall be used to create a new consistency group resource to provide expanded data protection through a replica relationship with the specified source consistency group.

Action URI

{Base URI of target resource}/Actions/ConsistencyGroup.CreateReplicaTarget

Action parameters

The parameters for the action which are included in the POST body to the URI shown in the 'target' property of the Action are summarized in Table 53.

Table 53: CreateReplicaTarget action parameters

Parameter Name	Type	Attributes	Notes
Cons istencyGroupName	string	*required*	This parameter shall contain the Name for the target consistency group.

Parameter Name	Type	Attributes	Notes
ReplicaType	string (enum)	*required*	This parameter shall contain the type of replica relationship to be created. *For the possible property values, see ReplicaType in Property details.*
R eplicaUpdateMode	string (enum)	*required*	This parameter shall specify the replica update mode. *For the possible property values, see Rep licaUpdateMode in Property details.*
T argetStoragePool	string	*required*	This parameter shall contain the Uri to the existing StoragePool in which to create the target consistency group.

9.6.5.4.3 RemoveReplicaRelationship   Description

This action shall be used to disable data synchronization between a source and target consistency group, remove the replication relationship, and optionally delete the target consistency group.

Action URI

{Base URI of target resource}/Actions/ConsistencyGroup.RemoveReplicaRelationship

Action parameters

The parameters for the action which are included in the POST body to the URI shown in the 'target' property of the Action are summarized in Table 54.

Table 54: RemoveReplicaRelationship action parameters

Parameter Name	Type	Attributes	Notes
DeleteTarget ConsistencyGroup	boolean	*optional*	This parameter shall indicate whether or not to delete the target consistency group as part of the operation. If not specified, the system should use its default behavior.
Target ConsistencyGroup	string	*required*	This parameter shall contain the Uri to the existing target consistency group.

9.6.5.4.4 ResumeReplication   Description

This action shall be used to resume the active data synchronization between a source and target consistency group, without otherwise altering the replication relationship.

Action URI

{Base URI of target resource}/Actions/ConsistencyGroup.ResumeReplication

Action parameters

The parameters for the action which are included in the POST body to the URI shown in the 'target' property of the Action are summarized in Table 55.

Table 55: ResumeReplication action parameters

Parameter Name	Type	Attributes	Notes
Target ConsistencyGroup	string	*required*	This parameter shall contain the Uri to the existing target consistency group.

9.6.5.4.5 ReverseReplicationRelationship   Description

This action shall be used to reverse the replication relationship between a source and target consistency group.

Action URI

{Base URI of target resource}/Actions/ConsistencyGroup.ReverseReplicationRelationship

Action parameters

The parameters for the action which are included in the POST body to the URI shown in the 'target' property of the Action are summarized in Table 56.

Table 56: ReverseReplicationRelationship action parameters

Parameter Name	Type	Attributes	Notes
Target ConsistencyGroup	string	*required*	This parameter shall contain the Uri to the existing target consistency group.

9.6.5.4.6 SplitReplication   Description

This action shall be used to split the replication relationship and suspend data synchronization between a source and target consistency group.

Action URI

{Base URI of target resource}/Actions/ConsistencyGroup.SplitReplication

Action parameters

The parameters for the action which are included in the POST body to the URI shown in the 'target' property of the Action are summarized in Table 57.

Table 57: SplitReplication action parameters

Parameter Name	Type	Attributes	Notes
Target ConsistencyGroup	string	*required*	This parameter shall contain the Uri to the existing target consistency group.

9.6.5.4.7 SuspendReplication   Description

This action shall be used to suspend active data synchronization between a source and target consistency group, without otherwise altering the replication relationship.

Action URI

{Base URI of target resource}/Actions/ConsistencyGroup.SuspendReplication

Action parameters

The parameters for the action which are included in the POST body to the URI shown in the 'target' property of the Action are summarized in Table 58.

Table 58: SuspendReplication action parameters

Parameter Name	Type	Attributes	Notes
Target ConsistencyGroup	string	*required*	This parameter shall contain the Uri to the existing target consistency group.

9.6.5.5 Property details

9.6.5.5.1 ConsistencyMethod The defined property values are listed in Table 59. The property shall set the consistency method used by this group.

Table 59: ConsistencyMethod property values

string	Description
HotStandby	Supports consistency method commonly orchestrated using application-specific code.
Other	Supports consistency method orchestrated using vendor-specific code.
VASA	Supports VMware consistency requirements, such as for VASA and VVOLs.
VDI	Supports Microsoft virtual backup device interface (VDI).
VSS	Supports Microsoft VSS.

9.6.5.5.2 ConsistencyType The defined property values are listed in Table 60. This property shall set the consistency type used by this group.

Table 60: ConsistencyType property values

string	Description
ApplicationConsistent	Orchestration exists to either flush or halt pending IO to ensure operations occur in a transactionally consistent manner.
CrashConsistent	Requested operations are either triggered or instituted without regard to pending IO.

9.6.5.5.3 ReplicaType　　The defined property values are listed in Table 61. This parameter shall contain the type of replica relationship to be created.

Table 61: ReplicaType property values

string	Description
Clone	This enumeration literal shall indicate that replication shall create a point in time, full copy the source.
Mirror	This enumeration literal shall indicate that replication shall create and maintain a copy of the source.
Snapshot	This enumeration literal shall indicate that replication shall create a point in time, virtual copy of the source.
TokenizedClone	This enumeration literal shall indicate that replication shall create a token based clone.

9.6.5.5.4 ReplicaUpdateMode　　The defined property values are listed in Table 62. This parameter shall specify the replica update mode.

Table 62: ReplicaUpdateMode property values

string	Description
Active	This enumeration literal shall indicate Active-Active (i.e. bidirectional) synchronous updates.
Adaptive	This enumeration literal shall indicate that an implementation may switch between synchronous and asynchronous modes.
Asynchronous	This enumeration literal shall indicate Asynchronous updates.
Synchronous	This enumeration literal shall indicate Synchronous updates.

9.6.6 ConsistencyGroupCollection

9.6.6.1 URIs   /redfish/v1/Storage/{*StorageId*}/ConsistencyGroups   /redfish/v1/ StorageServices/{*StorageServiceId*}/ConsistencyGroups   /redfish/v1/StorageSer- vices/{*StorageServiceId*}/Volumes/{*VolumeId*}/ConsistencyGroups   /redfish/v1/ Systems/{*ComputerSystemId*}/Storage/{*StorageId*}/ConsistencyGroups

9.6.6.2 Properties   The properties defined for the ConsistencyGroupCollection schema are summarized in Table 63.

Table 63: ConsistencyGroupCollection properties

Property	Type	Att ributes	Notes
Description	string	*re ad-only (null)*	This property shall contain the description of this resource. The value shall conform with the 'Description' clause of the Redfish Specification.

Property	Type	Att ributes	Notes
Members [{	array		The value of each member entry shall reference a ConsistencyGroup resource.
@odata.id	string	*rea d-only*	Link to a ConsistencyGroup resource. See the Links section and the *ConsistencyGroup* schema for details.
}]			
Me mbers@odata.nextLink	string (URI)	*rea d-only*	The value of this property shall be a URI to a resource, with the same @odata.type, containing the next set of partial members.
Name	string	*rea d-only*	This property shall contain the name of this resource or array member. The value shall conform with the 'Name' clause of the Redfish Specification.
Oem {}	object		This property shall contain the OEM extensions. All values for properties contained in this object shall conform to the Redfish S pecification-described requirements. For property details, see Oem.

9.6.7 DataProtectionLineOfService 1.3.0

9.6.7.1 Description   This service option describes a replica that protects data from loss. The requirements must be met collectively by the communication path and the replica.

9.6.7.2 URIs   /redfish/v1/StorageServices/{*StorageServiceId*}/ClassesOfService/ {*ClassOfServiceId*}/DataProtectionLinesOfService/{*DataProtectionLineOfServiceId*}   / redfish/v1/StorageServices/{*StorageServiceId*}/LinesOfService/DataProtectionLine-sOfService/{*DataProtectionLineOfServiceId*}

9.6.7.3 Properties   The properties defined for the DataProtectionLineOfService 1.3.0 schema are summarized in Table 64.

Table 64: DataProtectionLineOfService 1.3.0 properties

Property	Type	Att ributes	Notes
Actions *(v1.2+)* {	object		The Actions property shall contain the available actions for this resource.
# DataProtectionLineOfService.CreateReplicas {}	object		This action shall create an on-demand replica that conforms to the bound DataPr otectionLineOfService. *For more information, see the Actions section below.*
}			
Description	string	*re ad-only (null)*	This property shall contain the description of this resource. The value shall conform with the 'Description' clause of the Redfish Specification.

Property	Type	Att ributes	Notes
Id	string	*re ad-only re quired*	This property shall contain the identifier for this resource. The value shall conform with the 'Id' clause of the Redfish Specification.
IsIsolated	boolean	*rea d-write (null)*	True shall indicate that the replica is in a separate fault domain from its source. The default value of this property is false.
MinLifetime	string	*rea d-write (null)*	The value shall be an ISO 8601 duration that specifies the minimum required lifetime of the replica. Note: The maximum number of replicas can be determined using this value together with the replicaSchedule.
Name	string	*re ad-only re quired*	This property shall contain the name of this resource or array member. The value shall conform with the 'Name' clause of the Redfish Specification.

Property	Type	Att ributes	Notes
Oem {}	object		This property shall contain the OEM extensions. All values for properties that this object contains shall conform to the Redfish S pecification-described requirements. For property details, see Oem.
Recover yGeographicObjective	string (enum)	_re ad-writ e {=html} (null)_	The value specifies the geographic scope of the failure domain. _For the possible property values, see Recov eryGeographicObjective in Property details._
Recove ryPointObjectiveTime	string	_rea d-write (null)_	The value shall be an ISO 8601 duration that specifies the maximum time over which source data may be lost on failure. In the case that IsIsolated = false, failure of the domain is not a consideration.

Property	Type	Att ributes	Notes
R ecoveryTimeObjective	string (enum)	*re ad-write\<br\> {=html} (null)*	The value shall be an enumeration that indicates the maximum time required to access an alternate replica. In the case that IsIsolated = false, failure of the domain is not a consideration. *For the possible property values, see RecoveryTimeObjective in Property details.*
R eplicaAccessLocation {}	object		This value shall be used if the data access location of the replica is required to be at a specific location. Note 1: The location value may be granular. Note 2: A value may be required for some regulatory compliance. For property details, see Location v1.3.0).
R eplicaClassOfService {	object		The value shall reference the class of service that defines the required service levels of the replica. See the *ClassOfService* schema for details on this property.

Property	Type	Att ributes	Notes
@odata.id	string	*read -write*	Link to a ClassOfService resource. See the Links section and the *ClassOfService* schema for details.
}			
ReplicaType	string (enum)	*re ad-writ e\<br\> {=html} (null)*	The type of replica shall conform to this value. *For the possible property values, see ReplicaType in Property details.*
Schedule {}	object		If a replica is made periodically, the value shall define the schedule. For property details, see Schedule v1.2.5).

9.6.7.4 Actions

9.6.7.4.1 CreateReplicas   Description

This action shall create an on-demand replica that conforms to the bound DataPro-
tectionLineOfService.

Action URI

{Base URI of target resource}/Actions/DataProtectionLineOfService.CreateReplicas

Action parameters

The parameters for the action which are included in the POST body to the URI shown
in the 'target' property of the Action are summarized in Table 65.

Table 65: CreateReplicas action parameters

Parameter Name	Type	Attributes	Notes
Repl icaLineOfService {	object	*required*	The value shall reference the data protection line of service this operation is bound to.
@odata.id	string	*read-only*	Link to another DataProtectio nLineOfService resource.
}			
• *ReplicaRequests** [{	array	*optional*	Each value shall reference a source resource and provide a name for the replica.
ReplicaName *(v1.1+)*	string	*read-write (null)*	The value shall be the names of the replica.
ReplicaSource *(v1.1+)* {	object		The value shall reference a resource to be replicated.

Parameter Name	Type	Attributes	Notes
@odata.id	string (URI)	*read-only*	The value of this property shall be the unique identifier for the resource and it shall be of the form defined in the Redfish specification.

}

}]

9.6.7.5 Property details

9.6.7.5.1 RecoveryGeographicObjective   The defined property values are listed in Table 66. The value specifies the geographic scope of the failure domain.

Table 66: RecoveryGeographicObjective property values

string	Description
Datacenter	A facility that provides communication, power, or cooling infrastructure to a co-located set of servers, networking and storage.
Rack	A container within a datacenter that provides communication, power, or cooling to a set of components.
RackGroup	A set of racks that may share common communication, power, or cooling.
Region	A set of resources that are required to be either geographically or politically isolated from resources not in the resources.

string	Description
Row	A set of adjacent racks or rackgroups that may share common communication, power, or cooling.
Server	Components of a CPU/memory complex that share the same infrastructure.

9.6.7.5.2 RecoveryTimeObjective The defined property values are listed in Table 67. The value shall be an enumeration that indicates the maximum time required to access an alternate replica. In the case that IsIsolated = false, failure of the domain is not a consideration.

Table 67: RecoveryTimeObjective property values

string	Description
Nearline	Access to a replica shall be consistent with switching access to a different path through a different front-end interconnection infrastructure. Some inconsistency may occur. A restore step may be required before recovery can commence.
Offline	Access to a replica may take a significant amount of time. No direct connection to the replica is assumed. Some inconsistency loss may occur. A restore step is likely to be required.
OnlineActive	Access to synchronous replicas shall be instantaneous.
OnlinePassive	Access to a synchronous replica shall be consistent with switching access to a different path the same front-end interconnect. A restore step shall not be required.

9.6.7.5.3 ReplicaType The defined property values are listed in Table 68. The type of replica shall conform to this value.

Table 68: ReplicaType property values

string	Description
Clone	This enumeration literal shall indicate that replication shall create a point in time, full copy the source.
Mirror	This enumeration literal shall indicate that replication shall create and maintain a copy of the source.
Snapshot	This enumeration literal shall indicate that replication shall create a point in time, virtual copy of the source.
TokenizedClone	This enumeration literal shall indicate that replication shall create a token based clone.

9.6.8 DataProtectionLoSCapabilities 1.2.0

9.6.8.1 Description   The capabilities to protect data from loss by the use of a replica. The requirements shall be met collectively by the communication path and the replica. There should be one instance associated to a class of service for each replica. Each replica independently should have a class of service that describes its characteristics.

9.6.8.2 URIs   /redfish/v1/StorageServices/{*StorageServiceId*}/DataProtection-LoSCapabilities

9.6.8.3 Properties   The properties defined for the DataProtectionLoSCapabilities 1.2.0 schema are summarized in Table 69.

Table 69: DataProtectionLoSCapabilities 1.2.0 properties

Property	Type	Att ributes	Notes
Actions *(v1.1+)* {}	object		The Actions property shall contain the available actions for this resource.
Description	string	*re ad-only (null)*	This property shall contain the description of this resource. The value shall conform with the 'Description' clause of the Redfish Specification.
Id	string	*re ad-only re quired*	This property shall contain the identifier for this resource. The value shall conform with the 'Id' clause of the Redfish Specification.
Identifier {}	object		The value shall be unique within the managed ecosystem. For property details, see Identifier v1.19.1).
Links {	object		The value of this property shall contains links to other resources that are not contained in this resource.

Property	Type	Att ributes	Notes
Oem {}	object		This property shall contain the OEM extensions. All values for properties contained in this object shall conform to the Redfish S pecification-described requirements. For property details, see Oem.
Sup portedReplicaOptions [{	array		The collection shall contain known and supported replica Classes of Service.
@odata.id	string	*read -write*	Link to a ClassOfService resource. See the Links section and the *ClassOfService* schema for details.
}]			
}			
Name	string	*re ad-only re quired*	This property shall contain the name of this resource or array member. The value shall conform with the 'Name' clause of the Redfish Specification.

Property	Type	Att ributes	Notes
Oem {}	object		This property shall contain the OEM extensions. All values for properties that this object contains shall conform to the Redfish S pecification-described requirements. For property details, see Oem.
Sup portedLinesOfService [{	array		The collection shall contain known and supported DataPro tectionLinesOfService.
@odata.id	string	*read -write*	Link to a DataP rotectionLineOfService resource. See the Links section and the *DataPr otectionLineOfService* schema for details.
}]			
S upportedMinLifetimes [ ]	array (string, null)	*read -write*	The value of each entry shall be an ISO 8601 duration that specifies the minimum lifetime required for the replica.
SupportedRecovery GeographicObjectives [ ]	array (string (enum))	*re ad-writ e {=html} (null)*	The value of each entry shall specify a supported failure domain. *For the possible property values, see SupportedRecove ryGeographicObjectives in Property details.*

Property	Type	Att ributes	Notes
SupportedRecover yPointObjectiveTimes [ ]	array (string, null)	*read -write*	The value of each entry shall specify a supported ISO 8601 time interval defining the maximum source information that may be lost on failure. In the case that IsIsolated = false, failure of the domain is not a consideration.
SupportedRe coveryTimeObjectives [ ]	array (string (enum))	*re ad-writ e {=html} (null)*	The value of each entry shall specify an enumerated value that indicates a supported expectation for the time required to access an alternate replica. In the case that IsIsolated = false, failure of the domain is not a consideration. *For the possible property values, see Supported RecoveryTimeObjectives in Property details.*
S upportedReplicaTypes [ ]	array (string (enum))	*re ad-writ e {=html} (null)*	The value of each entry shall specify a supported replica type. *For the possible property values, see SupportedReplicaTypes in Property details.*

Property	Type	Att ributes	Notes
SupportsIsolated	boolean	*rea d-write (null)*	A value of true shall indicate that allocating a replica in a separate fault domain is supported. The default value for this property is false.

9.6.8.4 Property details

9.6.8.4.1 SupportedRecoveryGeographicObjectives The defined property values are listed in Table 70. The value of each entry shall specify a supported failure domain.

Table 70: SupportedRecoveryGeographicObjectives property values

string	Description
Datacenter	A facility that provides communication, power, or cooling infrastructure to a co-located set of servers, networking and storage.
Rack	A container within a datacenter that provides communication, power, or cooling to a set of components.
RackGroup	A set of racks that may share common communication, power, or cooling.
Region	A set of resources that are required to be either geographically or politically isolated from resources not in the resources.
Row	A set of adjacent racks or rackgroups that may share common communication, power, or cooling.
Server	Components of a CPU/memory complex that share the same infrastructure.

9.6.8.4.2 SupportedRecoveryTimeObjectives   The defined property values are listed in Table 71. The value of each entry shall specify an enumerated value that indicates a supported expectation for the time required to access an alternate replica. In the case that IsIsolated = false, failure of the domain is not a consideration.

Table 71: SupportedRecoveryTimeObjectives property values

string	Description
Nearline	Access to a replica shall be consistent with switching access to a different path through a different front-end interconnection infrastructure. Some inconsistency may occur. A restore step may be required before recovery can commence.
Offline	Access to a replica may take a significant amount of time. No direct connection to the replica is assumed. Some inconsistency loss may occur. A restore step is likely to be required.
OnlineActive	Access to synchronous replicas shall be instantaneous.
OnlinePassive	Access to a synchronous replica shall be consistent with switching access to a different path the same front-end interconnect. A restore step shall not be required.

9.6.8.4.3 SupportedReplicaTypes   The defined property values are listed in Table 72. The value of each entry shall specify a supported replica type.

Table 72: SupportedReplicaTypes property values

string	Description
Clone	This enumeration literal shall indicate that replication shall create a point in time, full copy the source.

string	Description
Mirror	This enumeration literal shall indicate that replication shall create and maintain a copy of the source.
Snapshot	This enumeration literal shall indicate that replication shall create a point in time, virtual copy of the source.
TokenizedClone	This enumeration literal shall indicate that replication shall create a token based clone.

9.6.9 DataSecurityLineOfService 1.1.1

9.6.9.1 Description   This structure shall be used to describe data security service level requirements.

9.6.9.2 URIs   /redfish/v1/StorageServices/{*StorageServiceId*}/ClassesOfService/ {*ClassOfServiceId*}/DataSecurityLinesOfService/{*DataSecurityLineOfServiceId*}   / redfish/v1/StorageServices/{*StorageServiceId*}/LinesOfService/DataSecurityLinesOf-Service/{*DataSecurityLineOfServiceId*}

9.6.9.3 Properties   The properties defined for the DataSecurityLineOfService 1.1.1 schema are summarized in Table 73.

Table 73: DataSecurityLineOfService 1.1.1 properties

Property	Type	Att ributes	Notes
Actions *(v1.1+)* {}	object		The Actions property shall contain the available actions for this resource.
Ant ivirusEngineProvider	string	*rea d-write (null)*	The value shall specify an AntiVirus provider.

Property	Type	Att ributes	Notes
A ntivirusScanPolicies [ ]	array (string (enum))	*re ad-writ e\<br\> {=html} (null)*	The enumeration literal shall specify the policy for triggering an AntiVirus scan. *For the possible property values, see AntivirusScanPolicies in Property details.*
Chann elEncryptionStrength	string (enum)	*re ad-writ e\<br\> {=html} (null)*	The enumeration literal shall specify a key size in a symmetric encryption algorithm for transport channel encryption. *For the possible property values, see Cha nnelEncryptionStrength in Property details.*
Da taSanitizationPolicy	string (enum)	*re ad-writ e\<br\> {=html} (null)*	The enumeration literal shall specify the data sanitization policy. *For the possible property values, see DataSanitizationPolicy in Property details.*
Description	string	*re ad-only (null)*	This property shall contain the description of this resource. The value shall conform with the 'Description' clause of the Redfish Specification.

Property	Type	Att ributes	Notes
Ho stAuthenticationType	string (enum)	*re ad-writ e {=html} (null)*	The enumeration literal shall specify the authentication type for hosts (servers) or initiator endpoints. *For the possible property values, see HostAuthenticationType in Property details.*
Id	string	*re ad-only re quired*	This property shall contain the identifier for this resource. The value shall conform with the 'Id' clause of the Redfish Specification.
Med iaEncryptionStrength	string (enum)	*re ad-writ e {=html} (null)*	The enumeration literal shall specify a key size in a symmetric encryption algorithm for media encryption. *For the possible property values, see M ediaEncryptionStrength in Property details.*
Name	string	*re ad-only re quired*	This property shall contain the name of this resource or array member. The value shall conform with the 'Name' clause of the Redfish Specification.

Property	Type	Att ributes	Notes
Oem {}	object		This property shall contain the OEM extensions. All values for properties that this object contains shall conform to the Redfish S pecification-described requirements. For property details, see Oem.
S ecureChannelProtocol	string (enum)	*re ad-write
 {=html} (null)*	The enumeration literal shall specify the protocol that provide encrypted communication. *For the possible property values, see SecureChannelProtocol in Property details.*
Us erAuthenticationType	string (enum)	*re ad-write
 {=html} (null)*	The enumeration literal shall specify the authentication type for users (or programs). *For the possible property values, see UserAuthenticationType in Property details.*

9.6.9.4 Property details

9.6.9.4.1 AntivirusScanPolicies   The defined property values are listed in Table 74. The enumeration literal shall specify the policy for triggering an AntiVirus scan.

Table 74: AntivirusScanPolicies property values

string	Description
None	This enumeration literal specifies No trigger.
OnFirstRead	This enumeration literal specifies to trigger on first read.
OnPatternUpdate	This enumeration literal specifies to trigger on antivirus pattern file update.
OnRename	This enumeration literal specifies to trigger on object rename.
OnUpdate	This enumeration literal specifies to trigger on object update.

9.6.9.4.2 ChannelEncryptionStrength   The defined property values are listed in Table 75. The enumeration literal shall specify a key size in a symmetric encryption algorithm for transport channel encryption.

Table 75: ChannelEncryptionStrength property values

string	Description
Bits_0	This enumeration literal specifies that there is no key.
Bits_112	This enumeration literal specifies a 3DES 112 bit key.
Bits_128	This enumeration literal specifies an AES 128 bit key.
Bits_192	This enumeration literal specifies an AES 192 bit key.
Bits_256	This enumeration literal specifies an AES 256 bit key.

9.6.9.4.3 DataSanitizationPolicy   The defined property values are listed in Table 76. The enumeration literal shall specify the data sanitization policy.

Table 76: DataSanitizationPolicy property values

string	Description
Clear	This enumeration literal specifies to sanitize data in all user-addressable storage locations for protection against simple non-invasive data recovery techniques.
CryptographicErase	This enumeration literal specifies to leverages the encryption of target data by enabling sanitization of the target data's encryption key. This leaves only the ciphertext remaining on the media, effectively sanitizing the data by preventing read-access. For more information, see NIST800-88 and ISO/IEC 27040.
None	This enumeration literal specifies no sanitization.

9.6.9.4.4 HostAuthenticationType The defined property values are listed in Table 77. The enumeration literal shall specify the authentication type for hosts (servers) or initiator endpoints.

Table 77: HostAuthenticationType property values

string	Description
None	This enumeration literal specifies No authentication.
Password	This enumeration literal specifies Password/shared-secret: Absent an distributed authentication infrastructure, this is what is typically done.

string	Description
PKI	This enumeration literal specifies a Public Key Infrastructure. Customers with the highest assurance requirements roll PKI out to hosts and users (it is more common for hosts than users. User PKI-based authentication has significant operational complications and administrative overheads, e.g., smart cards may be involved.
Ticket	This enumeration literal specifies Ticket-based (e.g., Kerberos): This is the most common class of authentication infrastructure used in enterprises. Kerberos is the best known example, and Windows usage of that via Active Directory is so widely deployed as to be a de facto standard. In other areas (e.g., academia) there are comparable ticket-based systems.

9.6.9.4.5 MediaEncryptionStrength   The defined property values are listed in Table 78. The enumeration literal shall specify a key size in a symmetric encryption algorithm for media encryption.

Table 78: MediaEncryptionStrength property values

string	Description
Bits_0	This enumeration literal specifies that there is no key.
Bits_112	This enumeration literal specifies a 3DES 112 bit key.
Bits_128	This enumeration literal specifies an AES 128 bit key.
Bits_192	This enumeration literal specifies an AES 192 bit key.
Bits_256	This enumeration literal specifies an AES 256 bit key.

9.6.9.4.6 SecureChannelProtocol   The defined property values are listed in Table 79. The enumeration literal shall specify the protocol that provide encrypted communication.

Table 79: SecureChannelProtocol property values

string	Description
IPsec	This enumeration literal specifies Internet Protocol Security (IPsec), as defined by IETF RFC 2401.
None	This enumeration literal specifies no encryption.
RPCSEC_GSS	This enumeration literal specifies RPC access to the Generic Security Services Application Programming Interface (GSS-API), as defined by IETF RPC 2203.
TLS	This enumeration literal specifies Transport Layer Security (TLS), as defined by IETF RFC 5246.

9.6.9.4.7 UserAuthenticationType   The defined property values are listed in Table 80. The enumeration literal shall specify the authentication type for users (or programs).

Table 80: UserAuthenticationType property values

string	Description
None	This enumeration literal specifies No authentication.
Password	This enumeration literal specifies Password/shared-secret: Absent an distributed authentication infrastructure, this is what is typically done.

string	Description
PKI	This enumeration literal specifies a Public Key Infrastructure. Customers with the highest assurance requirements roll PKI out to hosts and users (it is more common for hosts than users. User PKI-based authentication has significant operational complications and administrative overheads, e.g., smart cards may be involved.
Ticket	This enumeration literal specifies Ticket-based (e.g., Kerberos): This is the most common class of authentication infrastructure used in enterprises. Kerberos is the best known example, and Windows usage of that via Active Directory is so widely deployed as to be a de facto standard. In other areas (e.g., academia) there are comparable ticket-based systems.

9.6.10 DataSecurityLoSCapabilities 1.2.0

9.6.10.1 Description   This resource may be used to describe data security capabilities.

9.6.10.2 URIs   /redfish/v1/StorageServices/*{StorageServiceId}*/DataSecurity-LoSCapabilities

9.6.10.3 Properties   The properties defined for the DataSecurityLoSCapabilities 1.2.0 schema are summarized in Table 81.

Table 81: DataSecurityLoSCapabilities 1.2.0 properties

Property	Type	Att ributes	Notes
Actions *(v1.1+)* {}	object		The Actions property shall contain the available actions for this resource.
Description	string	*re ad-only (null)*	This property shall contain the description of this resource. The value shall conform with the 'Description' clause of the Redfish Specification.
Id	string	*re ad-only re quired*	This property shall contain the identifier for this resource. The value shall conform with the 'Id' clause of the Redfish Specification.
Identifier {}	object		The value identifies this resource. The value shall be unique within the managed ecosystem. For property details, see Identifier v1.19.1).
Name	string	*re ad-only re quired*	This property shall contain the name of this resource or array member. The value shall conform with the 'Name' clause of the Redfish Specification.

Property	Type	Att ributes	Notes
Oem {}	object		This property shall contain the OEM extensions. All values for properties that this object contains shall conform to the Redfish S pecification-described requirements. For property details, see Oem.
SupportedAnti virusEngineProviders [ ]	array (string, null)	*read -write*	The entry values shall specify supported AntiVirus providers.
SupportedA ntivirusScanPolicies [ ]	array (string (enum))	*re ad-writ e {=html} (null)*	The enumeration literal shall specify supported policies that trigger an AntiVirus scan. *For the possible property values, see Supporte dAntivirusScanPolicies in Property details.*
SupportedChanne lEncryptionStrengths [ ]	array (string (enum))	*re ad-writ e {=html} (null)*	The enumeration literal shall specify supported key sizes in a symmetric encryption algorithm (AES) for transport channel encryption. *For the possible property values, see SupportedChan nelEncryptionStrengths in Property details.*

Property	Type	Att ributes	Notes
SupportedData SanitizationPolicies [ ]	array (string (enum))	*re ad-writ e {=html} (null)*	The enumeration literal shall specify supported data sanitization policies. *For the possible property values, see SupportedDa taSanitizationPolicies in Property details.*
SupportedHos tAuthenticationTypes [ ]	array (string (enum))	*re ad-writ e {=html} (null)*	The enumeration literal shall specify supported authentication types for hosts (servers) or initiator endpoints. *For the possible property values, see SupportedH ostAuthenticationTypes in Property details.*
Sup portedLinesOfService [{	array		The collection shall contain supported DataSecurity service options.
@odata.id	string	*read -write*	Link to a Dat aSecurityLineOfService resource. See the Links section and the *Data SecurityLineOfService* schema for details.

}]

Property	Type	Att ributes	Notes
SupportedMedi aEncryptionStrengths [ ]	array (string (enum))	*re ad-writ e {=html} (null)*	The enumeration literal shall specify supported key sizes in a symmetric encryption algorithm (AES) for media encryption. *For the possible property values, see SupportedMe diaEncryptionStrengths in Property details.*
SupportedSe cureChannelProtocols [ ]	array (string (enum))	*re ad-writ e {=html} (null)*	The enumeration literal shall specify supported protocols that provide encrypted communication. *For the possible property values, see Supported SecureChannelProtocols in Property details.*
SupportedUse rAuthenticationTypes [ ]	array (string (enum))	*re ad-writ e {=html} (null)*	The enumeration literal shall specify supported authentication types for users (or programs). *For the possible property values, see SupportedU serAuthenticationTypes in Property details.*

9.6.10.4 Property details

9.6.10.4.1 SupportedAntivirusScanPolicies The defined property values are listed in Table 82. The enumeration literal shall specify supported policies that trigger an AntiVirus scan.

Table 82: SupportedAntivirusScanPolicies property values

string	Description
None	This enumeration literal specifies No trigger.
OnFirstRead	This enumeration literal specifies to trigger on first read.
OnPatternUpdate	This enumeration literal specifies to trigger on antivirus pattern file update.
OnRename	This enumeration literal specifies to trigger on object rename.
OnUpdate	This enumeration literal specifies to trigger on object update.

9.6.10.4.2 SupportedChannelEncryptionStrengths   The defined property values are listed in Table 83. The enumeration literal shall specify supported key sizes in a symmetric encryption algorithm (AES) for transport channel encryption.

Table 83: SupportedChannelEncryptionStrengths property values

string	Description
Bits_0	This enumeration literal specifies that there is no key.
Bits_112	This enumeration literal specifies a 3DES 112 bit key.
Bits_128	This enumeration literal specifies an AES 128 bit key.
Bits_192	This enumeration literal specifies an AES 192 bit key.
Bits_256	This enumeration literal specifies an AES 256 bit key.

9.6.10.4.3 SupportedDataSanitizationPolicies   The defined property values are listed in Table 84. The enumeration literal shall specify supported data sanitization policies.

Table 84: SupportedDataSanitizationPolicies property values

string	Description
Clear	This enumeration literal specifies to sanitize data in all user-addressable storage locations for protection against simple non-invasive data recovery techniques.
CryptographicErase	This enumeration literal specifies to leverages the encryption of target data by enabling sanitization of the target data's encryption key. This leaves only the ciphertext remaining on the media, effectively sanitizing the data by preventing read-access. For more information, see NIST800-88 and ISO/IEC 27040.
None	This enumeration literal specifies no sanitization.

9.6.10.4.4 SupportedHostAuthenticationTypes   The defined property values are listed in Table 85. The enumeration literal shall specify supported authentication types for hosts (servers) or initiator endpoints.

Table 85: SupportedHostAuthenticationTypes property values

string	Description
None	This enumeration literal specifies No authentication.
Password	This enumeration literal specifies Password/shared-secret: Absent an distributed authentication infrastructure, this is what is typically done.

string	Description
PKI	This enumeration literal specifies a Public Key Infrastructure. Customers with the highest assurance requirements roll PKI out to hosts and users (it is more common for hosts than users. User PKI-based authentication has significant operational complications and administrative overheads, e.g., smart cards may be involved.
Ticket	This enumeration literal specifies Ticket-based (e.g., Kerberos): This is the most common class of authentication infrastructure used in enterprises. Kerberos is the best known example, and Windows usage of that via Active Directory is so widely deployed as to be a de facto standard. In other areas (e.g., academia) there are comparable ticket-based systems.

9.6.10.4.5 SupportedMediaEncryptionStrengths   The defined property values are listed in Table 86. The enumeration literal shall specify supported key sizes in a symmetric encryption algorithm (AES) for media encryption.

Table 86: SupportedMediaEncryptionStrengths property values

string	Description
Bits_0	This enumeration literal specifies that there is no key.
Bits_112	This enumeration literal specifies a 3DES 112 bit key.
Bits_128	This enumeration literal specifies an AES 128 bit key.
Bits_192	This enumeration literal specifies an AES 192 bit key.
Bits_256	This enumeration literal specifies an AES 256 bit key.

9.6.10.4.6 SupportedSecureChannelProtocols   The defined property values are listed in Table 87. The enumeration literal shall specify supported protocols that provide encrypted communication.

Table 87: SupportedSecureChannelProtocols property values

string	Description
IPsec	This enumeration literal specifies Internet Protocol Security (IPsec), as defined by IETF RFC 2401.
None	This enumeration literal specifies no encryption.
RPCSEC_GSS	This enumeration literal specifies RPC access to the Generic Security Services Application Programming Interface (GSS-API), as defined by IETF RPC 2203.
TLS	This enumeration literal specifies Transport Layer Security (TLS), as defined by IETF RFC 5246.

9.6.10.4.7 SupportedUserAuthenticationTypes   The defined property values are listed in Table 88. The enumeration literal shall specify supported authentication types for users (or programs).

Table 88: SupportedUserAuthenticationTypes property values

string	Description
None	This enumeration literal specifies No authentication.
Password	This enumeration literal specifies Password/shared-secret: Absent an distributed authentication infrastructure, this is what is typically done.

string	Description
PKI	This enumeration literal specifies a Public Key Infrastructure. Customers with the highest assurance requirements roll PKI out to hosts and users (it is more common for hosts than users. User PKI-based authentication has significant operational complications and administrative overheads, e.g., smart cards may be involved.
Ticket	This enumeration literal specifies Ticket-based (e.g., Kerberos): This is the most common class of authentication infrastructure used in enterprises. Kerberos is the best known example, and Windows usage of that via Active Directory is so widely deployed as to be a de facto standard. In other areas (e.g., academia) there are comparable ticket-based systems.

9.6.11 DataStorageLineOfService 1.3.1

9.6.11.1 Description   This structure may be used to describe a service option covering storage provisioning and availability.

9.6.11.2 URIs   /redfish/v1/StorageServices/*{StorageServiceId}*/ClassesOfService/*{ClassOfServiceId}*/DataStorageLinesOfService/*{DataStorageLineOfServiceId}*   / redfish/v1/StorageServices/*{StorageServiceId}*/LinesOfService/DataStorageLinesOfService/*{DataStorageLineOfServiceId}*

9.6.11.3 Properties   The properties defined for the DataStorageLineOfService 1.3.1 schema are summarized in Table 89.

Table 89: DataStorageLineOfService 1.3.1 properties

Property	Type	Att ributes	Notes
AccessCapabilities *(v1.1+)* [ ]	array (string (enum))	*re ad-writ e {=html} (null)*	Each entry specifies a required storage access capability. *For the possible property values, see AccessCapabilities in Property details.*
Actions *(v1.3+)* {}	object		The Actions property shall contain the available actions for this resource.
Description	string	*re ad-only (null)*	This property shall contain the description of this resource. The value shall conform with the 'Description' clause of the Redfish Specification.
Id	string	*re ad-only re quired*	This property shall contain the identifier for this resource. The value shall conform with the 'Id' clause of the Redfish Specification.
IsSpaceEfficient	boolean	*rea d-write (null)*	A value of true shall indicate that the storage is compressed or deduplicated. The default value for this property is false.

Property	Type	Att ributes	Notes
Name	string	*re ad-only re quired*	This property shall contain the name of this resource or array member. The value shall conform with the 'Name' clause of the Redfish Specification.
Oem {}	object		This property shall contain the OEM extensions. All values for properties that this object contains shall conform to the Redfish S pecification-described requirements. For property details, see Oem.
ProvisioningPolicy	string (enum)	*re ad-writ e {=html} (null)*	The enumeration literal shall define the provisioning policy for storage. *For the possible property values, see ProvisioningPolicy in Property details.*

Property	Type	Att ributes	Notes
Recoverabl eCapacitySourceCount *(v1.2+)*	integer	*rea d-write (null)*	The value is minimum required number of available capacity source resources that shall be available in the event that an equivalent capacity source resource fails. It is assumed that drives and memory components can be replaced, repaired or otherwise added to increase an associated resource's Recoverab leCapacitySourceCount.
Re coveryTimeObjectives	string (enum)	*re ad-writ e\<br\> {=html} (null)*	The enumeration literal specifies the time after a disaster that the client shall regain conformant service level access to the primary store, typical values are 'immediate' or 'offline'. The expectation is that the services required to implement this capability are part of the advertising system. *For the possible property values, see RecoveryTimeObjectives in Property details.*

9.6.11.4 Property details

9.6.11.4.1 AccessCapabilities   The defined property values are listed in Table 90. Each entry specifies a required storage access capability.

Table 90: AccessCapabilities property values

string	Description
Append	This enumeration literal shall indicate that the storage may be written only to append.
Execute	This value shall indicate that Execute access is allowed by the file share.
Read	This enumeration literal shall indicate that the storage may be read.
Streaming	This enumeration literal shall indicate that the storage may be read sequentially.
Write	This enumeration literal shall indicate that the storage may be written multiple times.
WriteOnce	This enumeration literal shall indicate that the storage may be written only once.

9.6.11.4.2 ProvisioningPolicy   The defined property values are listed in Table 91. The enumeration literal shall define the provisioning policy for storage.

Table 91: ProvisioningPolicy property values

string	Description
Fixed	This enumeration literal specifies storage shall be fully allocated.
Thin	This enumeration literal specifies storage may be over allocated.

9.6.11.4.3 RecoveryTimeObjectives   The defined property values are listed in Table 92. The enumeration literal specifies the time after a disaster that the client shall regain conformant service level access to the primary store, typical values are 'immediate' or 'offline'. The expectation is that the services required to implement this capability are part of the advertising system.

Table 92: RecoveryTimeObjectives property values

string	Description
Nearline	Access to a replica shall be consistent with switching access to a different path through a different front-end interconnection infrastructure. Some inconsistency may occur. A restore step may be required before recovery can commence.
Offline	Access to a replica may take a significant amount of time. No direct connection to the replica is assumed. Some inconsistency loss may occur. A restore step is likely to be required.
OnlineActive	Access to synchronous replicas shall be instantaneous.
OnlinePassive	Access to a synchronous replica shall be consistent with switching access to a different path the same front-end interconnect. A restore step shall not be required.

9.6.12 DataStorageLoSCapabilities 1.2.2

9.6.12.1 Description   Each instance of DataStorageLoSCapabilities describes capabilities of the system to support various data storage service options.

9.6.12.2 URIs   /redfish/v1/StorageServices/{*StorageServiceId*}/DataStorageLoSCapabilities

9.6.12.3 Properties   The properties defined for the DataStorageLoSCapabilities 1.2.2 schema are summarized in Table 93.

Table 93: DataStorageLoSCapabilities 1.2.2 properties

Property	Type	Att ributes	Notes
Actions *(v1.1+)* {}	object		The Actions property shall contain the available actions for this resource.
Description	string	*re ad-only (null)*	This property shall contain the description of this resource. The value shall conform with the 'Description' clause of the Redfish Specification.
Id	string	*re ad-only re quired*	This property shall contain the identifier for this resource. The value shall conform with the 'Id' clause of the Redfish Specification.
Identifier {}	object		The value shall be unique within the managed ecosystem. For property details, see Identifier v1.19.1).
MaximumRecoverabl eCapacitySourceCount *(v1.2+)*	integer	*rea d-write (null)*	The maximum number of capacity source resources that can be supported for the purpose of recovery when in the event that an equivalent capacity source resource fails.

Property	Type	Att ributes	Notes
Name	string	*re ad-only re quired*	This property shall contain the name of this resource or array member. The value shall conform with the 'Name' clause of the Redfish Specification.
Oem {}	object		This property shall contain the OEM extensions. All values for properties that this object contains shall conform to the Redfish S pecification-described requirements. For property details, see Oem.
Support edAccessCapabilities [ ]	array (string (enum))	*re ad-writ e {=html} (null)*	Each entry specifies a storage access capability. *For the possible property values, see Suppo rtedAccessCapabilities in Property details.*
Sup portedLinesOfService [{	array		The collection shall contain known and supported Data StorageLinesOfService.
@odata.id	string	*read -write*	Link to a Da taStorageLineOfService resource. See the Links section and the *Dat aStorageLineOfService* schema for details.

}]

Property	Type	Att ributes	Notes
Supported ProvisioningPolicies [ ]	array (string (enum))	*re ad-writ e\<br\> {=html} (null)*	This collection specifies supported storage allocation policies. *For the possible property values, see Support edProvisioningPolicies in Property details.*
SupportedRe coveryTimeObjectives [ ]	array (string (enum))	*re ad-writ e\<br\> {=html} (null)*	This collection specifies supported expectations for time to access the primary store after recovery. *For the possible property values, see Supported RecoveryTimeObjectives in Property details.*
Sup portsSpaceEfficiency	boolean	*rea d-write (null)*	The value specifies whether storage compression or deduplication is supported. The default value for this property is false.

9.6.12.4 Property details

9.6.12.4.1 SupportedAccessCapabilities   The defined property values are listed in Table 94. Each entry specifies a storage access capability.

Table 94: SupportedAccessCapabilities property values

string	Description
Append	This enumeration literal shall indicate that the storage may be written only to append.
Execute	This value shall indicate that Execute access is allowed by the file share.
Read	This enumeration literal shall indicate that the storage may be read.
Streaming	This enumeration literal shall indicate that the storage may be read sequentially.
Write	This enumeration literal shall indicate that the storage may be written multiple times.
WriteOnce	This enumeration literal shall indicate that the storage may be written only once.

9.6.12.4.2 SupportedProvisioningPolicies The defined property values are listed in Table 95. This collection specifies supported storage allocation policies.

Table 95: SupportedProvisioningPolicies property values

string	Description
Fixed	This enumeration literal specifies storage shall be fully allocated.
Thin	This enumeration literal specifies storage may be over allocated.

9.6.12.4.3 SupportedRecoveryTimeObjectives The defined property values are listed in Table 96. This collection specifies supported expectations for time to access the primary store after recovery.

Table 96: SupportedRecoveryTimeObjectives property values

string	Description
Nearline	Access to a replica shall be consistent with switching access to a different path through a different front-end interconnection infrastructure. Some inconsistency may occur. A restore step may be required before recovery can commence.
Offline	Access to a replica may take a significant amount of time. No direct connection to the replica is assumed. Some inconsistency loss may occur. A restore step is likely to be required.
OnlineActive	Access to synchronous replicas shall be instantaneous.
OnlinePassive	Access to a synchronous replica shall be consistent with switching access to a different path the same front-end interconnect. A restore step shall not be required.

9.6.13 FeaturesRegistry 1.2.1

9.6.13.1 Description   This resource shall be used to represent a Feature registry for a Redfish implementation.

9.6.13.2 Properties   The properties defined for the FeaturesRegistry 1.2.1 schema are summarized in Table 97.

Table 97: FeaturesRegistry 1.2.1 properties

Property	Type	Att ributes	Notes
Actions {}	object		The Actions property shall contain the available actions for this resource.
Description	string	*re ad-only (null)*	This property shall contain the description of this resource. The value shall conform with the 'Description' clause of the Redfish Specification.
FeatureMappings *(v1.2+)* [{	array	*re quired*	The pattern property shall represent the suffix to be used in the FeatureId and shall be unique within this message registry. This may contain both standard and OEM-defined features.
Correspond ingProfileDefinition *(v1.2+)*	string	*re ad-only r equired (null)*	The value shall define a profile definition that contains the named profile declaration.
Description *(v1.2+)*	string	*re ad-only r equired (null)*	The value shall be a detailed description of the feature.
FeatureName *(v1.2+)*	string	*re ad-only r equired (null)*	The value shall be the unique name of the feature prefixed by the defining organization separated by a period (e .g. 'vendor.feature').

Property	Type	Att ributes	Notes
Resources *(v1.2+)* [{	array		The value shall contain an array of resources in the service containing the set that are related to support for the specified feature. For globally applied features, implementations should not implement the `Resources` property for the specified feature.
@odata.id	string (URI)	*rea d-only*	The value of this property shall be the unique identifier for the resource and it shall be of the form defined in the Redfish specification.
}]			
Version *(v1.2+)*	string	*re ad-only r equired (null)*	The value shall uniquely identify the version of the feature, using the major.minor.errata format.
}]			
Features *(deprecated v1.2)* [{	array		The pattern property shall represent the suffix to be used in the FeatureId and shall be unique within this message registry. *Deprecated in v1.2 and later. This property is deprecated in favor of the FeatureMappings property.*

Property	Type	Att ributes	Notes
Correspond ingProfileDefinition	string	*re ad-only r equired (null)*	If present, the value shall define a profile definition that contains the named profile declaration.
Description	string	*re ad-only r equired (null)*	The value shall be a detailed description of the feature.
FeatureName	string	*re ad-only r equired (null)*	The value shall be the unique name of the feature prefixed by the defining organization separated by a period (e .g. 'vendor.feature').
Version	string	*re ad-only r equired (null)*	The value shall uniquely identify the version of the feature, using the major.minor.errata format.
}]			
FeaturesUsed *(v1.2+)* [ ]	array (string, null)	*re ad-only re quired*	The property shall contain an array of all the standard feature names defined in the registry. OEM feature names shall be defined on the OEMFeaturesUsed property.
Id	string	*re ad-only re quired*	This property shall contain the identifier for this resource. The value shall conform with the 'Id' clause of the Redfish Specification.

Property	Type	Att ributes	Notes
Language	string	*re ad-only re quired*	The value of this property shall be a string consisting of an RFC 5646 language code.
Name	string	*re ad-only re quired*	This property shall contain the name of this resource or array member. The value shall conform with the 'Name' clause of the Redfish Specification.
Oem {}	object		This property shall contain the OEM extensions. All values for properties that this object contains shall conform to the Redfish S pecification-described requirements. For property details, see Oem.
OEMFeaturesUsed *(v1.2+)* [ ]	array (string, null)	*rea d-only*	The property shall contain an array of all the OEM feature names defined in the registry.
OwningEntity	string	*re ad-only re quired*	The value of this property shall be a string that represents the publisher of this registry.
RegistryPrefix	string	*re ad-only re quired*	The value of this property shall be the prefix used in IDs which uniquely identifies all of the Features in this registry as belonging to this registry.

Property	Type	Att ributes	Notes
RegistryVersion	string	*re ad-only re quired*	The value of this property shall be the version of this message registry. The format of this string shall be of the format majorversio n.minorversion.errata.

9.6.14 FileShare 1.3.0

9.6.14.1 Description This resource shall be used to represent a shared set of files with a common directory structure.

9.6.14.2 URIs /redfish/v1/Storage/*{StorageId}*/FileSystems/*{FileSystemsId}*/ ExportedFileShares/*{ExportedFileSharesId}* /redfish/v1/StorageServices/*{Storage-ServiceId}*/FileSystems/*{FileSystemsId}*/ExportedFileShares/*{ExportedFileSharesId}* / redfish/v1/Systems/*{ComputerSystemsId}*/Storage/*{StorageId}*/FileSystems/*{FileSys-temsId}*/ExportedFileShares/*{ExportedFileSharesId}*

9.6.14.3 Properties The properties defined for the FileShare 1.3.0 schema are summarized in Table 98.

Table 98: FileShare 1.3.0 properties

Property	Type	Att ributes	Notes
Actions *(v1.1+)* {}	object		The Actions property shall contain the available actions for this resource.

Property	Type	Att ributes	Notes
CASupported	boolean	*rea d-write (null)*	The value of this property shall indicate that Continuous Availability is supported. Client/Server mediated recovery from network and server failure with application transparency. This property shall be NULL unless the FileSharingProtocols property includes SMB. The default value for this property is false.
Defau ltAccessCapabilities [ ]	array (string (enum))	*r ead-onl y {=html} (null)*	The value of this property shall be an array containing entries for the default access capabilities for the file share. Each entry shall specify a default access privilege. The types of default access can include Read, Write, and/or Execute. *For the possible property values, see Def aultAccessCapabilities in Property details.*
Description	string	*re ad-only (null)*	This property shall contain the description of this resource. The value shall conform with the 'Description' clause of the Redfish Specification.

Property	Type	Att ributes	Notes
EthernetInterfaces {	object		The value shall be a link to an Ether netInterfaceCollection with members that provide access to the file share.
@odata.id	string (URI)	*rea d-only*	The value of this property shall be the unique identifier for the resource and it shall be of the form defined in the Redfish specification.
}			
ExecuteSupport	boolean	*re ad-only (null)*	The value of this property shall indicate whether Execute access is supported by the file share. The default value for this property is false.
FileSharePath	string	*re ad-only (null)*	The value of this property shall be a path (relative to the file system root) to the exported file or directory on the file system where this file share is hosted.

Property	Type	Att ributes	Notes
FileShareQuotaType	string (enum)	*re ad-writ e\<br\> {=html} (null)*	If FileShareQuotaType is present, a value of Soft shall specify that quotas are not enforced, and a value of Hard shall specify that writes shall fail if the space consumed would exceed the value of the Fi leShareTotalQuotaBytes property. *For the possible property values, see FileShareQuotaType in Property details.*
FileShar eRemainingQuotaBytes	integer (By)	*r ead-onl y\<br\> {=html} (null)*	If present, the value of this property shall indicate the remaining number of bytes that may be consumed by this file share.
File ShareTotalQuotaBytes	integer (By)	*re ad-writ e\<br\> {=html} (null)*	If present, the value of this property shall indicate the maximum number of bytes that may be consumed by this file share.

Property	Type	Att ributes	Notes
FileSharingProtocols [ ]	array (string (enum))	*r ead-onl y\<br\> {=html} (null)*	This property shall be an array containing entries for the file sharing protocols supported by this file share. Each entry shall specify a file sharing protocol supported by the file system. *For the possible property values, see FileSharingProtocols in Property details.*
Id	string	*re ad-only re quired*	This property shall contain the identifier for this resource. The value shall conform with the 'Id' clause of the Redfish Specification.
Links {	object		The Links property, as described by the Redfish Specification, shall contain references to resources that are related to, but not contained by (subordinate to), this resource.
ClassOfService {	object		This value shall be a link to the ClassOfService for this file share. See the *ClassOfService* schema for details on this property.
@odata.id	string	*rea d-only*	Link to a ClassOfService resource. See the Links section and the *ClassOfService* schema for details.

Property	Type	Att ributes	Notes
}			
FileSystem {	object		The value shall be a link to the file system containing the file share. See the *FileSystem* schema for details on this property.
@odata.id	string	*rea d-only*	Link to a FileSystem resource. See the Links section and the *FileSystem* schema for details.
}			
Oem {}	object		This property shall contain the OEM extensions. All values for properties contained in this object shall conform to the Redfish S pecification-described requirements. For property details, see Oem.
}			

Property	Type	Att ributes	Notes
LowSpaceWarn ingThresholdPercents [ ]	array (%) (integer, null)	*read -write*	This property shall be an array containing entries for the percentages of file share capacity at which low space warning events are be issued. A LOW_S PACE_THRESHOLD_WARNING event shall be triggered each time the remaining file share capacity value becomes less than one of the values in the array. The following shall be true: Across all CapacitySources entries, percent = (SUM(AllocatedBytes) - SUM(ConsumedBytes))/SUM(AllocatedBytes).
Name	string	*re ad-only re quired*	This property shall contain the name of this resource or array member. The value shall conform with the 'Name' clause of the Redfish Specification.
Oem {}	object		This property shall contain the OEM extensions. All values for properties that this object contains shall conform to the Redfish S pecification-described requirements. For property details, see Oem.

Property	Type	Att ributes	Notes
Rema iningCapacityPercent *(v1.1+)*	integer	*re ad-only (null)*	If present, this value shall return {[(SUM(AllocatedBytes) - SU M(ConsumedBytes)]/SUM (AllocatedBytes)}*100 represented as an integer value.
ReplicationEnabled *(v1.3+)*	boolean	*rea d-write (null)*	The property shall indicate whether or not replication is enabled on the file share. This property shall be consistent with the state reflected at the storage pool level.
RootAccess	boolean	*re ad-only (null)*	The value of this property shall indicate whether Root access is allowed by the file share. The default value for this property is false.
Status {}	object		This value of this property shall indicate the status of the file share. For property details, see Status.
WritePolicy	string (enum)	*r ead-onl y {=html} (null)*	The value of this property shall define how writes are replicated to the shared source. *For the possible property values, see WritePolicy in Property details.*

9.6.14.4 Property details

9.6.14.4.1 DefaultAccessCapabilities   The defined property values are listed in Table 99. The value of this property shall be an array containing entries for the default access capabilities for the file share. Each entry shall specify a default access privilege. The types of default access can include Read, Write, and/or Execute.

Table 99: DefaultAccessCapabilities property values

string	Description
Append	This enumeration literal shall indicate that the storage may be written only to append.
Execute	This value shall indicate that Execute access is allowed by the file share.
Read	This enumeration literal shall indicate that the storage may be read.
Streaming	This enumeration literal shall indicate that the storage may be read sequentially.
Write	This enumeration literal shall indicate that the storage may be written multiple times.
WriteOnce	This enumeration literal shall indicate that the storage may be written only once.

9.6.14.4.2 FileShareQuotaType   The defined property values are listed in Table 100. If FileShareQuotaType is present, a value of Soft shall specify that quotas are not enforced, and a value of Hard shall specify that writes shall fail if the space consumed would exceed the value of the FileShareTotalQuotaBytes property.

Table 100: FileShareQuotaType property values

string	Description
Hard	This value shall indicate that quotas are enabled and enforced.

string	Description
Soft	This value shall indicate that quotas are enabled but not enforced.

9.6.14.4.3　FileSharingProtocols　　The defined property values are listed in Table 101. This property shall be an array containing entries for the file sharing protocols supported by this file share. Each entry shall specify a file sharing protocol supported by the file system.

Table 101: FileSharingProtocols property values

string	Description
NFSv3	This value shall indicate that NFSv3, as defined in RFC 1813, is supported by the file system.
NFSv4_0	This value shall indicate that NFSv4, as defined in RFC 7530, is supported by the file system.
NFSv4_1	This value shall indicate that NFSv4.1, as defined in RFC 5661, is supported by the file system.
SMBv2_0	This value shall indicate that Server Message Block version 2.0 is supported by the file system.
SMBv2_1	This value shall indicate that Server Message Block version 2.1 is supported by the file system.
SMBv3_0	This value shall indicate that Server Message Block version 3.0 is supported by the file system.
SMBv3_0_2	This value shall indicate that Server Message Block version 3.0.2 is supported by the file system.
SMBv3_1_1	This value shall indicate that Server Message Block version 3.1.1 is supported by the file system.

9.6.14.4.4　WritePolicy　　The defined property values are listed in Table 102. The value of this property shall define how writes are replicated to the shared source.

Table 102: WritePolicy property values

string	Description
Active	This enumeration literal shall indicate Active-Active (i.e. bidirectional) synchronous updates.
Adaptive	This enumeration literal shall indicate that an implementation may switch between synchronous and asynchronous modes.
Asynchronous	This enumeration literal shall indicate Asynchronous updates.
Synchronous	This enumeration literal shall indicate Synchronous updates.

9.6.15 FileShareCollection

9.6.15.1 URIs   /redfish/v1/Storage/*{StorageId}*/FileSystems/*{FileSystemsId}*/ExportedFileShares   /redfish/v1/StorageServices/*{StorageServiceId}*/FileSystems/*{FileSystemsId}*/ExportedFileShares

9.6.15.2 Properties   The properties defined for the FileShareCollection schema are summarized in Table 103.

Table 103: FileShareCollection properties

Property	Type	Att ributes	Notes
Description	string	*re ad-only (null)*	This property shall contain the description of this resource. The value shall conform with the 'Description' clause of the Redfish Specification.

Property	Type	Att ributes	Notes
Members [{	array		This property shall contain references to the members of this FileSystem collection.
@odata.id	string	*rea d-only*	Link to a FileShare resource. See the Links section and the *FileShare* schema for details.
}]			
Me mbers@odata.nextLink	string (URI)	*rea d-only*	The value of this property shall be a URI to a resource, with the same @odata.type, containing the next set of partial members.
Name	string	*rea d-only*	This property shall contain the name of this resource or array member. The value shall conform with the 'Name' clause of the Redfish Specification.
Oem {}	object		This property shall contain the OEM extensions. All values for properties contained in this object shall conform to the Redfish S pecification-described requirements. For property details, see Oem.

9.6.16 FileSystem 1.4.1

9.6.16.1 Description   This resource shall be used to represent an instance of a hierarchical namespace of files.

9.6.16.2 URIs   /redfish/v1/Storage/*{StorageId}*/FileSystems/*{FileSystemId}*   /
redfish/v1/StorageServices/*{StorageServiceId}*/FileSystems/*{FileSystemId}*

9.6.16.3 Properties   The properties defined for the FileSystem 1.4.1 schema are summarized in Table 104.

Table 104: FileSystem 1.4.1 properties

Property	Type	Att ributes	Notes
AccessCapabilities [ ]	array (string (enum))	*re ad-write {=html} (null)*	This property shall be an array containing entries for the supported IO access capabilities. Each entry shall specify a current storage access capability. *For the possible property values, see AccessCapabilities in Property details.*
Actions *(v1.1+)* {}	object		The Actions property shall contain the available actions for this resource.
BlockSizeBytes	integer (By)	*r ead-only {=html} (null)*	The value of this property shall be the block size of the file system in bytes.

Property	Type	Att ributes	Notes
Capacity {}	object		The value of this property shall be the capacity allocated to the file system in bytes. For property details, see Capacity v1.0.0).
CapacitySources [{	array		This property shall be an array containing entries for all the capacity sources for the file system. Each entry shall provide capacity allocation information from a named resource.
@odata.id	string	*read -write*	Link to a CapacitySource resource. See the Links section and the *CapacitySource* schema for details.
}]			
CasePreserved	boolean	*rea d-write (null)*	This property shall indicate that the case of file names is preserved by the file system. A value of True shall indicate that case of file names shall be preserved.
CaseSensitive	boolean	*rea d-write (null)*	This property shall indicate that case sensitive file names are supported by the file system. A value of True shall indicate that file names are case sensitive.

Property	Type	Att ributes	Notes
CharacterCodeSet [ ]	array (string (enum))	*re ad-writ e
 {=html} (null)*	This property shall be an array containing entries for the character sets or encodings supported by the file system. Each entry shall specify a character set encoding supported by the file system. *For the possible property values, see CharacterCodeSet in Property details.*
ClusterSizeBytes	integer (By)	*re ad-writ e
 {=html} (null)*	This value shall specify the minimum file allocation size imposed by the file system. This minimum allocation size shall be the smallest amount of storage allocated to a file by the file system. Under stress conditions, the file system may allocate storage in amounts smaller than this value.
Description	string	*re ad-only (null)*	This property shall contain the description of this resource. The value shall conform with the 'Description' clause of the Redfish Specification.

Property	Type	Att ributes	Notes
ExportedShares {	object		This property shall be an array of exported file shares of this file system. Each entry shall define an exported file share of this file system. Contains a link to a resource.
@odata.id	string	*read -write*	Link to Collection of *FileShare*. See the FileShare schema for details.
}			
Id	string	*re ad-only re quired*	This property shall contain the identifier for this resource. The value shall conform with the 'Id' clause of the Redfish Specification.
Identifiers *(v1.1.1+)* [{ }]	array (object)		This property shall contain a list of all known durable names for this file system. For property details, see Identifier v1.19.1).
ImportedShares *(v1.0.1+)* [{	array		The value shall be an array of imported file shares.
ImportedShare		*read -write*	
}]			

Property	Type	Att ributes	Notes
IOStatistics *(v1.2+, deprecated v1.4)* {}	object		The value shall represent IO statistics for this FileSystem. For property details, see IOStatistics. *Deprecated in v1.4 and later. This property is deprecated in favor of the IOStatistics property in FileSystemMetrics.*
Links {	object		This property shall contain links to other resources that are related to this resource.
ClassOfService {	object		This value shall be a link to the ClassOfService for this file system. See the *ClassOfService* schema for details on this property.
@odata.id	string	*rea d-only*	Link to a ClassOfService resource. See the Links section and the *ClassOfService* schema for details.
}			
Oem {}	object		This property shall contain the OEM extensions. All values for properties contained in this object shall conform to the Redfish S pecification-described requirements. For property details, see Oem.

Property	Type	Att ributes	Notes
ReplicaCollection [{	array		This property shall be an array of links to replicas for this file system. Each entry shall be a link to a replica for this file system.
@odata.id	string	*rea d-only*	Link to another FileSystem resource.
}]			
SpareResourceSets *(v1.2+)* [{	array		Each referenced SpareResourceSet shall contain resources that may be utilized to replace the capacity provided by a failed resource having a compatible type.
@odata.id	string	*read -write*	Link to a SpareResourceSet resource. See the Links section and the *SpareResourceSet* schema for details.
}]			
}			

Property	Type	Att ributes	Notes
LowSpaceWarn ingThresholdPercents [ ]	array (%) (integer, null)	*read -write*	This property shall be an array containing entries for the percentages of file system capacity at which low space warning events are be issued. A LOW_S PACE_THRESHOLD_WARNING event shall be triggered each time the remaining file system capacity value becomes less than one of the values in the array. The following shall be true: Across all CapacitySources entries, percent = (SUM(AllocatedBytes) - SUM(ConsumedBytes))/SUM(AllocatedBytes).
Ma xFileNameLengthBytes	integer (By)	*re ad-writ e {=html} (null)*	If specified, this value shall specify the maximum length of a file name within the file system.
Metrics *(v1.4+)* {	object	* (null)*	This property shall contain a link to a resource of type FileSystemMetrics that specifies the metrics for this file system. IO metrics are reported in the IOStatistics property. See the *FileSystemMetrics* schema for details on this property.

Property	Type	Att ributes	Notes
@odata.id	string	*rea d-only*	Link to a FileSystemMetrics resource. See the Links section and the *FileSystemMetrics* schema for details.
}			
Name	string	*re ad-only* *re quired*	This property shall contain the name of this resource or array member. The value shall conform with the 'Name' clause of the Redfish Specification.
Oem {}	object		This property shall contain the OEM extensions. All values for properties that this object contains shall conform to the Redfish S pecification-described requirements. For property details, see Oem.
Recoverabl eCapacitySourceCount *(v1.2+)*	integer	*rea d-write (null)*	The value is the number of available capacity source resources currently available in the event that an equivalent capacity source resource fails.

Property	Type	Att ributes	Notes
RemainingCapacity {}	object		The value of this property shall be the remaining capacity allocated to the file system in bytes. For property details, see Capacity v1.0.0).
Rema iningCapacityPercent *(v1.1+)*	integer	*re ad-only (null)*	If present, this value shall return {[(SUM(AllocatedBytes) - SU M(ConsumedBytes)]/SUM (AllocatedBytes)}*100 represented as an integer value.
ReplicaInfo {	object		If this file system is a replica, this value shall describe its replication attributes. This value shall not be present if this file system is not a replica. A file system may be both a source and a replica. See the *StorageReplicaInfo* schema for details on this property.
@odata.id	string	*rea d-only*	Link to a ReplicaInfo resource. See the Links section and the *StorageReplicaInfo* schema for details.
}			

Property	Type	Att ributes	Notes
ReplicaTargets *(v1.2.1+)* [{	array		The value shall reference the target replicas that are sourced by this replica.
@odata.id	string (URI)	*rea d-only*	The value of this property shall be the unique identifier for the resource and it shall be of the form defined in the Redfish specification.
}]			
ReplicationEnabled *(v1.3+)*	boolean	*rea d-write (null)*	The property shall indicate whether or not replication is enabled on the file system. This property shall be consistent with the state reflected at the storage pool level.

9.6.16.4 Property details

9.6.16.4.1 AccessCapabilities   The defined property values are listed in Table 105. This property shall be an array containing entries for the supported IO access capabilities. Each entry shall specify a current storage access capability.

Table 105: AccessCapabilities property values

string	Description
Append	This enumeration literal shall indicate that the storage may be written only to append.
Execute	This value shall indicate that Execute access is allowed by the file share.

string	Description
Read	This enumeration literal shall indicate that the storage may be read.
Streaming	This enumeration literal shall indicate that the storage may be read sequentially.
Write	This enumeration literal shall indicate that the storage may be written multiple times.
WriteOnce	This enumeration literal shall indicate that the storage may be written only once.

9.6.16.4.2 CharacterCodeSet   The defined property values are listed in Table 106. This property shall be an array containing entries for the character sets or encodings supported by the file system. Each entry shall specify a character set encoding supported by the file system.

Table 106: CharacterCodeSet property values

string	Description
ASCII	This value shall indicate that the ASCII character encoding is supported by the file system.
ExtendedUNIXCode	This value shall indicate that Extended Unix Code character encoding is supported by the file system.
ISO2022	This value shall indicate that ISO-2022 character encoding is supported by the file system.
ISO8859_1	This value shall indicate that ISO-8859-1 character encoding is supported by the file system.
UCS_2	This value shall indicate that the UCS-2 character encoding is supported by the file system.
Unicode	This value shall indicate that Unicode character encoding is supported by the file system.

string	Description
UTF_16	This value shall indicate that the UTF-16 character encoding is supported by the file system.
UTF_8	This value shall indicate that the UTF-8 character encoding is supported by the file system.

9.6.17 FileSystemCollection

9.6.17.1 URIs   /redfish/v1/Storage/*{StorageId}*/FileSystems   /redfish/v1/Storage-Services/*{StorageServiceId}*/FileSystems

9.6.17.2 Properties   The properties defined for the FileSystemCollection schema are summarized in Table 107.

Table 107: FileSystemCollection properties

Property	Type	Att ributes	Notes
Description	string	*re ad-only (null)*	This property shall contain the description of this resource. The value shall conform with the 'Description' clause of the Redfish Specification.
Members [{	array		This property shall contain references to the members of this FileSystem collection.
@odata.id	string	*rea d-only*	Link to a FileSystem resource. See the Links section and the *FileSystem* schema for details.
}]			

Property	Type	Att ributes	Notes
Me mbers@odata.nextLink	string (URI)	*rea d-only*	The value of this property shall be a URI to a resource, with the same @odata.type, containing the next set of partial members.
Name	string	*rea d-only*	This property shall contain the name of this resource or array member. The value shall conform with the 'Name' clause of the Redfish Specification.
Oem {}	object		This property shall contain the OEM extensions. All values for properties contained in this object shall conform to the Redfish S pecification-described requirements. For property details, see Oem.

9.6.18 FileSystemMetrics 1.0.1

9.6.18.1 Description   The FileSystemMetrics schema shall contain the usage and health statistics for a file system in a Redfish implementation.

9.6.18.2 URIs   /redfish/v1/Storage/{*StorageId*}/FileSystems/{*FileSystemId*}/Met-rics   /redfish/v1/StorageServices/{*StorageServiceId*}/FileSystems/{*FileSystemId*}/Metrics

9.6.18.3 Properties   The properties defined for the FileSystemMetrics 1.0.1 schema are summarized in Table 108.

Table 108: FileSystemMetrics 1.0.1 properties

Property	Type	Att ributes	Notes
Actions {}	object		This property shall contain the available actions for this resource.
Description	string	*re ad-only (null)*	This property shall contain the description of this resource. The value shall conform with the 'Description' clause of the Redfish Specification.
Id	string	*re ad-only re quired*	This property shall contain the identifier for this resource. The value shall conform with the 'Id' clause of the Redfish Specification.
IOStatistics {}	object		The value shall represent IO statistics for this file system. For property details, see IOStatistics.
Name	string	*re ad-only re quired*	This property shall contain the name of this resource or array member. The value shall conform with the 'Name' clause of the Redfish Specification.

Property	Type	Att ributes	Notes
Oem {}	object		This property shall contain the OEM extensions. All values for properties that this object contains shall conform to the Redfish S pecification-described requirements. For property details, see Oem.

9.6.19 HostedStorageServices

9.6.19.1 URIs /redfish/v1/Systems/*{ComputerSystemId}*/HostedServices

9.6.19.2 Properties The properties defined for the HostedStorageServices schema are summarized in Table 109.

Table 109: HostedStorageServices properties

Property	Type	Att ributes	Notes
Description	string	*re ad-only (null)*	This property shall contain the description of this resource. The value shall conform with the 'Description' clause of the Redfish Specification.
Members [{	array		The value of each member entry shall reference a StorageService resource.

Property	Type	Att ributes	Notes
@odata.id	string	*rea d-only*	Link to a StorageService resource. See the Links section and the *StorageService* schema for details.
}]			
Me mbers@odata.nextLink	string (URI)	*rea d-only*	The value of this property shall be a URI to a resource, with the same @odata.type, containing the next set of partial members.
Name	string	*rea d-only*	This property shall contain the name of this resource or array member. The value shall conform with the 'Name' clause of the Redfish Specification.
Oem {}	object		This property shall contain the OEM extensions. All values for properties contained in this object shall conform to the Redfish S pecification-described requirements. For property details, see Oem.

9.6.20 IOConnectivityLineOfService 1.2.1

9.6.20.1 Description   An IO connectivity service option may be used to specify the characteristics of storage connectivity.

9.6.20.2 URIs /redfish/v1/StorageServices/*{StorageServiceId}*/ClassesOfService/
{ClassOfServiceId}/IOConnectivityLinesOfService/*{IOConnectivityLineOfServiceId}* /
redfish/v1/StorageServices/*{StorageServiceId}*/LinesOfService/IOConnectivityLine-
sOfService/*{IOConnectivityLineOfServiceId}*

9.6.20.3 Properties The properties defined for the IOConnectivityLineOfService
1.2.1 schema are summarized in Table 110.

Table 110: IOConnectivityLineOfService 1.2.1 properties

Property	Type	Att ributes	Notes
AccessProtocols [ ]	array (string (enum))	*re ad-writ e
 {=html} (null)*	The Enumeration Literal shall specify the Access protocol for this service option. NOTE: If multiple protocols are specified, the corresponding MaxSupportedIOPS governs the max achieved across all protocol uses. This may be less than the sum of the individual max values, which may be specified by individual Line of Service entries. *For the possible property values, see AccessProtocols in Property details.*
Actions *(v1.2+)* {}	object		The Actions property shall contain the available actions for this resource.

Property	Type	Att ributes	Notes
Description	string	*re ad-only (null)*	This property shall contain the description of this resource. The value shall conform with the 'Description' clause of the Redfish Specification.
Id	string	*re ad-only re quired*	This property shall contain the identifier for this resource. The value shall conform with the 'Id' clause of the Redfish Specification.
MaxBytesPerSecond *(v1.1+)*	integer (By/s)	*re ad-writ e {=html} (null)*	The value shall be the maximum bytes per second that a connection can support.
MaxIOPS *(v1.1+)*	integer ([IO]/s)	*re ad-writ e {=html} (null)*	The value shall be the maximum IOs per second that the connection shall allow for the selected access protocol.
Name	string	*re ad-only re quired*	This property shall contain the name of this resource or array member. The value shall conform with the 'Name' clause of the Redfish Specification.

Property	Type	Att ributes	Notes
Oem {}	object		This property shall contain the OEM extensions. All values for properties that this object contains shall conform to the Redfish S pecification-described requirements. For property details, see Oem.

9.6.20.4 Property details

9.6.20.4.1 AccessProtocols   The defined property values are listed in Table 111. The Enumeration Literal shall specify the Access protocol for this service option. NOTE: If multiple protocols are specified, the corresponding MaxSupportedIOPS governs the max achieved across all protocol uses. This may be less than the sum of the individual max values, which may be specified by individual Line of Service entries.

Table 111: AccessProtocols property values

string	Description
AHCI	This value shall indicate conformance to the Intel Advanced Host Controller Interface (AHCI) Specification.
CXL	This value shall indicate conformance to the Compute Express Link Specification.
DisplayPort	This value shall indicate conformance to the VESA DisplayPort Specification.
DVI	This value shall indicate conformance to the Digital Display Working Group DVI-A, DVI-D, or DVI-I Specification.

string	Description
eMMC	This value shall indicate conformance to the JEDEC JESD84-B51A specification.
Ethernet	This value shall indicate conformance to the IEEE 802.3 Ethernet specification.
FC	This value shall indicate conformance to the T11 Fibre Channel Physical and Signaling Interface Specification.
FCoE	This value shall indicate conformance to the T11 FC-BB-5 Specification.
FCP	This value shall indicate conformance to the INCITS 481: Information Technology - Fibre Channel Protocol for SCSI.
FICON	This value shall indicate conformance to the ANSI FC-SB-3 Single-Byte Command Code Sets-3 Mapping Protocol for the Fibre Channel (FC) protocol. Fibre Connection (FICON) is the IBM-proprietary name for this protocol.
FTP	This value shall indicate conformance to the RFC114-defined File Transfer Protocol (FTP).
GenZ	This value shall indicate conformance to the Gen-Z Core Specification.
HDMI	This value shall indicate conformance to the HDMI Forum HDMI Specification.
HTTP	This value shall indicate conformance to the Hypertext Transport Protocol (HTTP) as defined by RFC3010 or RFC5661.
HTTPS	This value shall indicate conformance to the Hypertext Transfer Protocol Secure (HTTPS) as defined by RFC2068 or RFC2616, which uses Transport Layer Security (TLS) as defined by RFC5246 or RFC6176.
I2C	This value shall indicate conformance to the NXP Semiconductors I2C-bus Specification.

string	Description
InfiniBand	This value shall indicate conformance to the InfiniBand Architecture Specification-defined InfiniBand protocol.
iSCSI	This value shall indicate conformance to the IETF Internet Small Computer Systems Interface (iSCSI) Specification.
iWARP	This value shall indicate conformance to the RFC5042-defined Internet Wide Area RDMA Protocol (iWARP) that uses the transport layer mechanisms as defined by RFC5043 or RFC5044.
MultiProtocol	This value shall indicate conformance to multiple protocols.
NFSv3	This value shall indicate conformance to the RFC1813-defined Network File System (NFS) protocol.
NFSv4	Network File System (NFS) version 4.
NVLink	This value shall indicate conformance to the NVIDIA NVLink protocol.
NVMe	This value shall indicate conformance to the Non-Volatile Memory Host Controller Interface Specification.
NVMeOverFabrics	This value shall indicate conformance to the NVM Express over Fabrics Specification.
OEM	This value shall indicate conformance to an OEM-specific architecture, and the OEM section may include additional information.
PCIe	This value shall indicate conformance to the PCI-SIG PCI Express Base Specification.
QPI	This value shall indicate conformance to the Intel QuickPath Interconnect (QPI) protocol.
RoCE	This value shall indicate conformance to the InfiniBand Architecture Specification-defined RDMA over Converged Ethernet Protocol.

string	Description
RoCEv2	This value shall indicate conformance to the InfiniBand Architecture Specification-defined RDMA over Converged Ethernet Protocol version 2.
SAS	This value shall indicate conformance to the T10 SAS Protocol Layer Specification.
SATA	This value shall indicate conformance to the Serial ATA International Organization Serial ATA Specification.
SFTP	This value shall indicate conformance to the RFC114-defined SSH File Transfer Protocol (SFTP) that uses Transport Layer Security (TLS) as defined by RFC5246 or RFC6176.
SMB	This value shall indicate conformance to the Server Message Block (SMB), or Common Internet File System (CIFS), protocol.
TCP	This value shall indicate conformance to the IETF-defined Transmission Control Protocol (TCP). For example, RFC7414 defines the roadmap of the TCP specification.
TFTP	This value shall indicate conformance to the IETF-defined Trivial File Transfer Protocol (TFTP). For example, RFC1350 defines the core TFTP version 2 specification.
UDP	This value shall indicate conformance to the IETF-defined User Datagram Protocol (UDP). For example, RFC768 defines the core UDP specification.
UHCI	This value shall indicate conformance to the Intel Universal Host Controller Interface (UHCI) Specification, Enhanced Host Controller Interface Specification, or the Extensible Host Controller Interface Specification.

string	Description
UPI	This value shall indicate conformance to the Intel UltraPath Interconnect (UPI) protocol.
USB	This value shall indicate conformance to the USB Implementers Forum Universal Serial Bus Specification.
VGA	This value shall indicate conformance to the VESA SVGA Specification.

9.6.21 IOConnectivityLoSCapabilities 1.2.0

9.6.21.1 Description   Each instance of IOConnectivityLoSCapabilities describes capabilities of the system to support various IO Connectivity service options.

9.6.21.2 URIs   /redfish/v1/StorageServices/{*StorageServiceId*}/IOConnectivity-LoSCapabilities

9.6.21.3 Properties   The properties defined for the IOConnectivityLoSCapabilities 1.2.0 schema are summarized in Table 112.

Table 112: IOConnectivityLoSCapabilities 1.2.0 properties

Property	Type	Att ributes	Notes
Actions *(v1.1+)* {}	object		The Actions property shall contain the available actions for this resource.
Description	string	*re ad-only (null)*	This property shall contain the description of this resource. The value shall conform with the 'Description' clause of the Redfish Specification.

Property	Type	Att ributes	Notes
Id	string	*re ad-only* *re quired*	This property shall contain the identifier for this resource. The value shall conform with the 'Id' clause of the Redfish Specification.
Identifier {}	object		The value identifies this resource. The value shall be unique within the managed ecosystem. For property details, see Identifier v1.19.1).
MaxSup portedBytesPerSecond	integer (By/s)	*re ad-writ e {=html} (null)*	The value shall be the maximum bytes per second that a connection can support.
MaxSupportedIOPS *(v1.1+)*	integer ([IO]/s)	*re ad-writ e {=html} (null)*	The value shall be the maximum IOPS that a connection can support.
Name	string	*re ad-only* *re quired*	This property shall contain the name of this resource or array member. The value shall conform with the 'Name' clause of the Redfish Specification.

Property	Type	Att ributes	Notes
Oem {}	object		This property shall contain the OEM extensions. All values for properties that this object contains shall conform to the Redfish S pecification-described requirements. For property details, see Oem.
Supp ortedAccessProtocols [ ]	array (string (enum))	*re ad-writ e\<br\> {=html} (null)*	Access protocols supported by this service option. NOTE: SMB+NFS* requires that SMB and at least one of NFSv3 or NFXv4 are also selected, (i.e. {'SMB', 'NFSv4', 'SMB+NFS*}). For the possible property values, see Su pportedAccessProtocols in Property details.*
Sup portedLinesOfService [{	array		The collection shall contain known and supported IOConne ctivityLinesOfService.
@odata.id	string	*read -write*	Link to a IOCon nectivityLineOfService resource. See the Links section and the *IOConn ectivityLineOfService* schema for details.
}]			

9.6.21.4 Property details

9.6.21.4.1 SupportedAccessProtocols   The defined property values are listed in Table 113. Access protocols supported by this service option. NOTE: SMB+NFS* requires that SMB and at least one of NFSv3 or NFXv4 are also selected, (i.e. {'SMB', 'NFSv4', 'SMB+NFS*'}).

Table 113: SupportedAccessProtocols property values

string	Description
AHCI	This value shall indicate conformance to the Intel Advanced Host Controller Interface (AHCI) Specification.
CXL	This value shall indicate conformance to the Compute Express Link Specification.
DisplayPort	This value shall indicate conformance to the VESA DisplayPort Specification.
DVI	This value shall indicate conformance to the Digital Display Working Group DVI-A, DVI-D, or DVI-I Specification.
eMMC	This value shall indicate conformance to the JEDEC JESD84-B51A specification.
Ethernet	This value shall indicate conformance to the IEEE 802.3 Ethernet specification.
FC	This value shall indicate conformance to the T11 Fibre Channel Physical and Signaling Interface Specification.
FCoE	This value shall indicate conformance to the T11 FC-BB-5 Specification.
FCP	This value shall indicate conformance to the INCITS 481: Information Technology - Fibre Channel Protocol for SCSI.

string	Description
FICON	This value shall indicate conformance to the ANSI FC-SB-3 Single-Byte Command Code Sets-3 Mapping Protocol for the Fibre Channel (FC) protocol. Fibre Connection (FICON) is the IBM-proprietary name for this protocol.
FTP	This value shall indicate conformance to the RFC114-defined File Transfer Protocol (FTP).
GenZ	This value shall indicate conformance to the Gen-Z Core Specification.
HDMI	This value shall indicate conformance to the HDMI Forum HDMI Specification.
HTTP	This value shall indicate conformance to the Hypertext Transport Protocol (HTTP) as defined by RFC3010 or RFC5661.
HTTPS	This value shall indicate conformance to the Hypertext Transfer Protocol Secure (HTTPS) as defined by RFC2068 or RFC2616, which uses Transport Layer Security (TLS) as defined by RFC5246 or RFC6176.
I2C	This value shall indicate conformance to the NXP Semiconductors I2C-bus Specification.
InfiniBand	This value shall indicate conformance to the InfiniBand Architecture Specification-defined InfiniBand protocol.
iSCSI	This value shall indicate conformance to the IETF Internet Small Computer Systems Interface (iSCSI) Specification.
iWARP	This value shall indicate conformance to the RFC5042-defined Internet Wide Area RDMA Protocol (iWARP) that uses the transport layer mechanisms as defined by RFC5043 or RFC5044.
MultiProtocol	This value shall indicate conformance to multiple protocols.

string	Description
NFSv3	This value shall indicate conformance to the RFC1813-defined Network File System (NFS) protocol.
NFSv4	Network File System (NFS) version 4.
NVLink	This value shall indicate conformance to the NVIDIA NVLink protocol.
NVMe	This value shall indicate conformance to the Non-Volatile Memory Host Controller Interface Specification.
NVMeOverFabrics	This value shall indicate conformance to the NVM Express over Fabrics Specification.
OEM	This value shall indicate conformance to an OEM-specific architecture, and the OEM section may include additional information.
PCIe	This value shall indicate conformance to the PCI-SIG PCI Express Base Specification.
QPI	This value shall indicate conformance to the Intel QuickPath Interconnect (QPI) protocol.
RoCE	This value shall indicate conformance to the InfiniBand Architecture Specification-defined RDMA over Converged Ethernet Protocol.
RoCEv2	This value shall indicate conformance to the InfiniBand Architecture Specification-defined RDMA over Converged Ethernet Protocol version 2.
SAS	This value shall indicate conformance to the T10 SAS Protocol Layer Specification.
SATA	This value shall indicate conformance to the Serial ATA International Organization Serial ATA Specification.

string	Description
SFTP	This value shall indicate conformance to the RFC114-defined SSH File Transfer Protocol (SFTP) that uses Transport Layer Security (TLS) as defined by RFC5246 or RFC6176.
SMB	This value shall indicate conformance to the Server Message Block (SMB), or Common Internet File System (CIFS), protocol.
TCP	This value shall indicate conformance to the IETF-defined Transmission Control Protocol (TCP). For example, RFC7414 defines the roadmap of the TCP specification.
TFTP	This value shall indicate conformance to the IETF-defined Trivial File Transfer Protocol (TFTP). For example, RFC1350 defines the core TFTP version 2 specification.
UDP	This value shall indicate conformance to the IETF-defined User Datagram Protocol (UDP). For example, RFC768 defines the core UDP specification.
UHCI	This value shall indicate conformance to the Intel Universal Host Controller Interface (UHCI) Specification, Enhanced Host Controller Interface Specification, or the Extensible Host Controller Interface Specification.
UPI	This value shall indicate conformance to the Intel UltraPath Interconnect (UPI) protocol.
USB	This value shall indicate conformance to the USB Implementers Forum Universal Serial Bus Specification.
VGA	This value shall indicate conformance to the VESA SVGA Specification.

9.6.22 IOPerformanceLineOfService 1.1.1

9.6.22.1 Description   This structure may be used to define a service option related to IO performance.

9.6.22.2 URIs   /redfish/v1/StorageServices/*{StorageServiceId}*/ClassesOfService/ *{ClassOfServiceId}*/IOPerformanceLinesOfService/*{IOPerformanceLineOfServiceId}*   / redfish/v1/StorageServices/*{StorageServiceId}*/LinesOfService/IOPerformanceLine- sOfService/*{IOPerformanceLineOfServiceId}*

9.6.22.3 Properties   The properties defined for the IOPerformanceLineOfService 1.1.1 schema are summarized in Table 114.

Table 114: IOPerformanceLineOfService 1.1.1 properties

Property	Type	Att ributes	Notes
Actions *(v1.1+)* {}	object		The Actions property shall contain the available actions for this resource.
AverageIOOperatio nLatencyMicroseconds	integer (us)	*re ad-writ e {=html} (null)*	The value shall be the expected average IO latency in microseconds calculated over sample periods (see SamplePeriodSeconds).
Description	string	*re ad-only (null)*	This property shall contain the description of this resource. The value shall conform with the 'Description' clause of the Redfish Specification.

Property	Type	Att ributes	Notes
Id	string	*re ad-only re quired*	This property shall contain the identifier for this resource. The value shall conform with the 'Id' clause of the Redfish Specification.
IOOperatio nsPerSecondIsLimited	boolean	*rea d-write (null)*	If true, the system should not allow IOPS to exceed MaxIoOperatio nsPerSecondPerTerabyte * VolumeSize. Otherwise, the system shall not enforce a limit. The default value for this property is false.
IOWorkload {}	object		The value shall be a description of the expected workload. The workload provides the context in which the values of MaxIOOperatio nsPerSecondPerTerabyte and AverageIOOperat ionLatencyMicroseconds are expected to be achievable. For property details, see IOWorkload v1.0.0).

Property	Type	Att ributes	Notes
MaxIOOperations PerSecondPerTerabyte	integer (1/s/TBy)	*re ad-writ e {=html} (null)*	The value shall be the amount of IOPS a volume of a given committed size in Terabytes can support. This IOPS density value is useful as a metric that is independent of capacity. Cost is a function of this value and the AverageIOOperati onLatencyMicroseconds.
Name	string	*re ad-only re quired*	This property shall contain the name of this resource or array member. The value shall conform with the 'Name' clause of the Redfish Specification.
Oem {}	object		This property shall contain the OEM extensions. All values for properties that this object contains shall conform to the Redfish S pecification-described requirements. For property details, see Oem.
SamplePeriod	string	*rea d-write (null)*	The value shall be an ISO 8601 duration specifying the sampling period over which average values are calculated.

9.6.23 IOPerformanceLoSCapabilities 1.3.0

9.6.23.1 Description   Each instance of IOPerformanceLoSCapabilities shall describe the capabilities of the system to support various IO performance service options.

9.6.23.2 URIs   /redfish/v1/StorageServices/{*StorageServiceId*}/IOPerformanceLoSCapabilities

9.6.23.3 Properties   The properties defined for the IOPerformanceLoSCapabilities 1.3.0 schema are summarized in Table 115.

Table 115: IOPerformanceLoSCapabilities 1.3.0 properties

Property	Type	Att ributes	Notes
Actions *(v1.1+)* {}	object		The Actions property shall contain the available actions for this resource.
Description	string	*re ad-only (null)*	This property shall contain the description of this resource. The value shall conform with the 'Description' clause of the Redfish Specification.
Id	string	*re ad-only re quired*	This property shall contain the identifier for this resource. The value shall conform with the 'Id' clause of the Redfish Specification.
Identifier {}	object		The value shall be unique within the managed ecosystem. For property details, see Identifier v1.19.1).

Property	Type	Att ributes	Notes
I OLimitingIsSupported	boolean	*rea d-write (null)*	If true, the system should limit IOPS to MaxIOOperatio nsPerSecondPerTerabyte * (Volume Size in Terabytes). Otherwise, the system shall not inforce a limit. The default value for this property is false.
MaxSamplePeriod	string (s)	*re ad-writ e {=html} (null)*	The value shall be an ISO 8601 duration specifying the maximum sampling period over which average values are calculated.
MinSamplePeriod	string (s)	*re ad-writ e {=html} (null)*	The value shall be an ISO 8601 duration specifying the minimum sampling period over which average values are calculated.
MinSupportedIoOpera- tio nLatencyMicroseconds	integer (us)	*re ad-writ e {=html} (null)*	The value shall be the minimum supported average IO latency in microseconds calculated over the SamplePeriod.
Name	string	*re ad-only re quired*	This property shall contain the name of this resource or array member. The value shall conform with the 'Name' clause of the Redfish Specification.

Property	Type	Att ributes	Notes
Oem {}	object		This property shall contain the OEM extensions. All values for properties that this object contains shall conform to the Redfish S pecification-described requirements. For property details, see Oem.
SupportedIOWorkloads [{ }]	array (object)	* (null)*	The value shall be a collection of supported workloads. For property details, see IOWorkload.
Sup portedLinesOfService [{	array		The value shall be a collection supported IO performance service options.
@odata.id	string	*read -write*	Link to a IOPe rformanceLineOfService resource. See the Links section and the *IOPer formanceLineOfService* schema for details.
}]			

9.6.24 LineOfService 1.1.0

9.6.24.1 Description   This service option is the abstract base class for other Clas-sOfService and concrete lines of service.

9.6.24.2 Properties   The properties defined for the LineOfService 1.1.0 schema are summarized in Table 116.

Table 116: LineOfService 1.1.0 properties

Property	Type	Att ributes	Notes
Description	string	*re ad-only (null)*	This property shall contain the description of this resource. The value shall conform with the 'Description' clause of the Redfish Specification.
Id	string	*re ad-only re quired*	This property shall contain the identifier for this resource. The value shall conform with the 'Id' clause of the Redfish Specification.
Name	string	*re ad-only re quired*	This property shall contain the name of this resource or array member. The value shall conform with the 'Name' clause of the Redfish Specification.
Oem {}	object		This property shall contain the OEM extensions. All values for properties that this object contains shall conform to the Redfish S pecification-described requirements. For property details, see Oem.

9.6.25 LineOfServiceCollection

9.6.25.1 URIs　/redfish/v1/StorageServices/{*StorageServiceId*}/ClassesOfService/ {*ClassOfServiceId*}/DataProtectionLinesOfService　　　　/redfish/v1/StorageServices/

{StorageServiceId}/ClassesOfService/*{ClassOfServiceId}*/DataSecurityLinesOfService
/redfish/v1/StorageServices/*{StorageServiceId}*/ClassesOfService/*{ClassOfServiceId}*/DataStorageLinesOfService   /redfish/v1/StorageServices/*{StorageServiceId}*/
ClassesOfService/*{ClassOfServiceId}*/IOConnectivityLinesOfService /redfish/v1/StorageServices/*{StorageServiceId}*/ClassesOfService/*{ClassOfServiceId}*/IOPerformance-
LinesOfService   /redfish/v1/StorageServices/*{StorageServiceId}*/LinesOfService   /
redfish/v1/StorageServices/*{StorageServiceId}*/LinesOfService/DataProtection-
LinesOfService   /redfish/v1/StorageServices/*{StorageServiceId}*/LinesOfService/
DataSecurityLinesOfService   /redfish/v1/StorageServices/*{StorageServiceId}*/LinesOfService/DataStorageLinesOfService   /redfish/v1/StorageServices/*{StorageServiceId}*/LinesOfService/IOConnectivityLinesOfService   /redfish/v1/StorageServices/
{StorageServiceId}/LinesOfService/IOPerformanceLinesOfService

9.6.25.2 Properties   The properties defined for the LineOfServiceCollection
schema are summarized in Table 117.

Table 117: LineOfServiceCollection properties

Property	Type	Att ributes	Notes
Description	string	*re ad-only (null)*	This property shall contain the description of this resource. The value shall conform with the 'Description' clause of the Redfish Specification.
Members [{	array		The value of each member entry shall reference a LineOfService resource.
@odata.id	string	*rea d-only*	Link to a LineOfService resource. See the Links section and the *LineOfService* schema for details.
}]			

Property	Type	Att ributes	Notes
Me mbers@odata.nextLink	string (URI)	*rea d-only*	The value of this property shall be a URI to a resource, with the same @odata.type, containing the next set of partial members.
Name	string	*rea d-only*	This property shall contain the name of this resource or array member. The value shall conform with the 'Name' clause of the Redfish Specification.
Oem {}	object		This property shall contain the OEM extensions. All values for properties contained in this object shall conform to the Redfish S pecification-described requirements. For property details, see Oem.

9.6.26 NVMeDomain 1.2.0

9.6.26.1 Description   Properties for the Domain.

9.6.26.2 URIs   /redfish/v1/NVMeDomains/*{NVMeDomainId}*

9.6.26.3 Properties   The properties defined for the NVMeDomain 1.2.0 schema are summarized in Table 118.

Table 118: NVMeDomain 1.2.0 properties

Property	Type	Att ributes	Notes
Actions {}	object		This property shall contain the available actions for this resource.
ANAGroupId *(v1.2+)*	number	*re ad-only (null)*	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.
Ava ilableFirmwareImages [{	array		A collection of available firmware images.
@odata.id	string	*rea d-only*	Link to a NVMeFirmwareImage resource. See the Links section and the *NVMeFirmwareImage* schema for details.
}]			
Description	string	*re ad-only (null)*	This property shall contain the description of this resource. The value shall conform with the 'Description' clause of the Redfish Specification.
DomainContents *(v1.2+)* {	object	* (null)*	This property shall contain the members of the domain.

Property	Type	Att ributes	Notes
Controllers *(v1.2+)* [{	array		Contains the current controllers that are part of this domain. These can be IO, Admin, or discovery controllers.
@odata.id	string (URI)	*rea d-only*	The value of this property shall be the unique identifier for the resource and it shall be of the form defined in the Redfish specification.
}]			
Namespaces *(v1.2+)* [{	array		Contains the current namespaces that are part of this domain. These can be IO, Admin, or discovery controllers.
@odata.id	string	*rea d-only*	Link to a Volume resource. See the Links section and the *Volume* schema for details.
}] }			
DomainMembers [{	array		The members of the domain.
@odata.id	string (URI)	*rea d-only*	The value of this property shall be the unique identifier for the resource and it shall be of the form defined in the Redfish specification.
}]			

Property	Type	Att ributes	Notes
FirmwareImages *(v1.2+)* [{	array		This property shall contain an array of pointers to available firmware images.
@odata.id	string (URI)	*rea d-only*	The value of this property shall be the unique identifier for the resource and it shall be of the form defined in the Redfish specification.
}]			
Id	string	*re ad-only* *re quired*	This property shall contain the identifier for this resource. The value shall conform with the 'Id' clause of the Redfish Specification.
Links {	object		This property shall contain links to resources that are related to but are not contained by or subordinate to this resource.
AssociatedDomains [{	array		This property shall contain an array of links to resources of type NVMeDomain that represent associated domains.
@odata.id	string	*rea d-only*	Link to another NVMeDomain resource.
}]			

Property	Type	Att ributes	Notes
Oem {} }	object		This property shall contain the OEM extensions. All values for properties contained in this object shall conform to the Redfish S pecification-described requirements. For property details, see Oem.
MaximumCapacityPe rEnduranceGroupBytes	integer (By)	*r ead-onl y\<br\> {=html} (null)*	This property shall contain the maximum capacity per endurance group in bytes of this NVMe Domain.
MaxNamespacesSu pportedPerController *(v1.2+)*	number	*re ad-only (null)*	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.
Name	string	*re ad-only re quired*	This property shall contain the name of this resource or array member. The value shall conform with the 'Name' clause of the Redfish Specification.

Property	Type	Att ributes	Notes
Oem {}	object		This property shall contain the OEM extensions. All values for properties that this object contains shall conform to the Redfish S pecification-described requirements. For property details, see Oem.
Status {}	object		This property shall contain any status or health properties of the resource. For property details, see Status.
Tota lDomainCapacityBytes	integer (By)	*r ead-onl y {=html} (null)*	This property shall contain the total capacity in bytes of this NVMe Domain.
Unallocate dDomainCapacityBytes	integer (By)	*r ead-onl y {=html} (null)*	This property shall contain the total unallocated capacity in bytes of this NVMe Domain.

9.6.27 NVMeDomainCollection

9.6.27.1 URIs   /redfish/v1/NVMeDomains

9.6.27.2 Properties   The properties defined for the NVMeDomainCollection schema are summarized in Table 119.

Table 119: NVMeDomainCollection properties

Property	Type	Att ributes	Notes
Description	string	*re ad-only (null)*	This property shall contain the description of this resource. The value shall conform with the 'Description' clause of the Redfish Specification.
Members [{	array		The value of each member entry shall reference a NVMeDomain resource.
@odata.id	string	*rea d-only*	Link to a NVMeDomain resource. See the Links section and the *NVMeDomain* schema for details.
}]			
Me mbers@odata.nextLink	string (URI)	*rea d-only*	The value of this property shall be a URI to a resource, with the same @odata.type, containing the next set of partial members.
Name	string	*rea d-only*	This property shall contain the name of this resource or array member. The value shall conform with the 'Name' clause of the Redfish Specification.

Property	Type	Att ributes	Notes
Oem {}	object		This property shall contain the OEM extensions. All values for properties contained in this object shall conform to the Redfish S pecification-described requirements. For property details, see Oem.

9.6.28 NVMeFirmwareImage 1.2.0 (deprecated)

This schema has been deprecated and use in new implementations is discouraged except to retain compatibility with existing products. This schema has been deprecated in favor of the SoftwareInventory schema.

9.6.28.1 Description　　NVMe Domain firmware image information.

9.6.28.2 URIs　　/redfish/v1/NVMeDomains/*{DomainId}*/AvailableFirmwareImages/ *{FirmwareImageId}*

9.6.28.3 Properties　　The properties defined for the NVMeFirmwareImage 1.2.0 (deprecated) schema are summarized in Table 120.

Table 120: NVMeFirmwareImage 1.2.0 (deprecated) properties

Property	Type	Att ributes	Notes
Actions {}	object		This property shall contain the available actions for this resource.

Property	Type	Att ributes	Notes
Description	string	*re ad-only (null)*	This property shall contain the description of this resource. The value shall conform with the 'Description' clause of the Redfish Specification.
FirmwareVersion	string	*re ad-only (null)*	This property shall contain the firmware version of the available NVMe firmware image.
Id	string	*re ad-only re quired*	This property shall contain the identifier for this resource. The value shall conform with the 'Id' clause of the Redfish Specification.
Name	string	*re ad-only re quired*	This property shall contain the name of this resource or array member. The value shall conform with the 'Name' clause of the Redfish Specification.
NVMeDeviceType	string (enum)	*r ead-onl y
 {=html} (null)*	This property shall specify the type of NVMe device for this NVMe firmware image. *For the possible property values, see NVMeDeviceType in Property details.*

Property	Type	Att ributes	Notes
Oem {}	object		This property shall contain the OEM extensions. All values for properties that this object contains shall conform to the Redfish S pecification-described requirements. For property details, see Oem.
Vendor	string	*re ad-only (null)*	This property shall include the name of the manufacturer or vendor associate with this NVMe firmware image.

9.6.28.4 Property details

9.6.28.4.1 NVMeDeviceType   The defined property values are listed in Table 121. This property shall specify the type of NVMe device for this NVMe firmware image.

Table 121: NVMeDeviceType property values

string	Description
Drive	Specifies an device type of Drive, indicating a NVMe device that presents as an NVMe SSD device.
FabricAttachArray	Specifies an NVMe device type of FabricAttachArray, indicating a NVMe device that presents an NVMe front-end that abstracts the back end storage, typically with multiple options for availability and protection.

string	Description
JBOF	Specifies an device type of JBOF, indicating a NVMe device that presents as an NVMe smart enclosure for NVMe devices, typically NVMe Drives.

9.6.29 SpareResourceSet 1.0.2

9.6.29.1 Description   The values define a set of spares of a particular type.

9.6.29.2 Properties   The properties defined for the SpareResourceSet 1.0.2 schema are summarized in Table 122.

Table 122: SpareResourceSet 1.0.2 properties

Property	Type	Att ributes	Notes
Actions *(v1.0.1+)* {}	object		The Actions property shall contain the available actions for this resource.
Description	string	*re ad-only (null)*	This property shall contain the description of this resource. The value shall conform with the 'Description' clause of the Redfish Specification.
Id	string	*re ad-only re quired*	This property shall contain the identifier for this resource. The value shall conform with the 'Id' clause of the Redfish Specification.

Property	Type	Att ributes	Notes
Links {	object		This structure shall contain references to resources that are not contained within this resource.
Oem {}	object		This property shall contain the OEM extensions. All values for properties contained in this object shall conform to the Redfish S pecification-described requirements. For property details, see Oem.
OnHandSpares [{	array		The type of resources in the set.
@odata.id	string (URI)	*rea d-only*	The value of this property shall be the unique identifier for the resource and it shall be of the form defined in the Redfish specification.
}]			
ReplacementSpareSets [{	array		Other spare sets that can be utilized to replenish this spare set.
@odata.id	string	*rea d-only*	Link to another SpareResourceSet resource.
}]			
}			

Property	Type	Att ributes	Notes
Name	string	*re ad-only re quired*	This property shall contain the name of this resource or array member. The value shall conform with the 'Name' clause of the Redfish Specification.
Oem {}	object		This property shall contain the OEM extensions. All values for properties that this object contains shall conform to the Redfish S pecification-described requirements. For property details, see Oem.
OnHandLocation {}	object		The location where this set of spares is kept. For property details, see Location v1.5.0).
OnLine	boolean	*rea d-write (null)*	This set shall be available online.
ResourceType	string	*rea d-write (null)*	The type of resources in the set.
TimeToProvision	string (d uration)	*re ad-writ e {=html} (null)*	Amount of time needed to make an on-hand resource available as a spare. Pattern: ^P(\d+D)?(T(\d+H)?(\d +M)?(\d+(.\d+)?S)?)?$

Property	Type	Att ributes	Notes
TimeToReplenish	string (d uration)	*re ad-writ e\<br\> {=html} (null)*	Amount of time needed to replenish consumed on-hand resources. Pattern: `^P(\d+D)?(T(\d+H)?(\d +M)?(\d+(.\d+)?S)?)?$`

9.6.30 StorageGroup 1.6.0

9.6.30.1 Description   The primary purposes of the collection shall be to govern access to the storage by clients or to add service requirements for the members of the collection. Access to the collected storage by a specified set of hosts shall be made available or unavailable atomically. Requirements specified by the class of service shall be satisfied by each collected element to which they apply. The storage group may contain: block, file, or object storage; local storage system access points through which the collection is made available; and hosts, or host access points to which the collection is made available.

9.6.30.2 URIs   /redfish/v1/Storage/*{StorageId}*/StorageGroups/*{StorageGroupId}* (deprecated)   /redfish/v1/Storage/*{StorageId}*/Volumes/*{VolumeId}*/StorageGroups/ *{StorageGroupId}*   (deprecated)   /redfish/v1/StorageServices/*{StorageServiceId}*/ StorageGroups/*{StorageGroupId}*   (deprecated)   /redfish/v1/StorageServices/*{StorageServiceId}*/Volumes/*{VolumeId}*/StorageGroups/*{StorageGroupId}* (deprecated)

9.6.30.3 Properties   The properties defined for the StorageGroup 1.6.0 schema are summarized in Table 123.

Table 123: StorageGroup 1.6.0 properties

Property	Type	Att ributes	Notes
AccessState	string (enum)	*re ad-writ e {=html} (null)*	The value of this property shall describe the access characteristics of this storage group. All associated logical units through all aggregated ports shall share this access state. *For the possible property values, see AccessState in Property details.*
Actions {	object		The Actions property shall contain the available actions for this resource.
#Storag eGroup.ExposeVolumes {}	object		Exposes the storage of this group via the target endpoints named in the ServerEndpointGroups to the initiator endpoints named in the ClientEndpointGroups. The property VolumesAreExposed shall be set to true when this action is completed. *For more information, see the Actions section below.*

Property	Type	Att ributes	Notes
#Stor ageGroup.HideVolumes {} }	object		Hide the storage of this group from the initiator endpoints named in the ClientEndpointGroups. The property VolumesAreExposed shall be set to false when this action is completed. *For more information, see the Actions section below.*
AuthenticationMethod *(v1.2+)*	string (enum)	*re ad-writ e\<br\> {=html} (null)*	The value of this property must be what kind of authentication that the endpoints in this StorageGroup understands. *For the possible property values, see AuthenticationMethod in Property details.*

Property	Type	Att ributes	Notes
ChapInfo *(v1.2+)* [{	array		The value of this property must reflect the authentication used by this specific endpoint. If this endpoint represents an initiator, and AuthenticationMethod is CHAP or MutualCHAP, the Credentials fields CHAPUsername and CHAPSecret must be used. If this endpoint represents a target endpoint and AuthenticationMethod is MutualCHAP, then MutualCHAPUsername and MutualCHAPSecret must be used.
CHAPPassword *(v1.3+)*	string	*rea d-write (null)*	The value of this property shall be the password when CHAP authentication is specified.
CHAPUser *(v1.3+)*	string	*rea d-write (null)*	The value of this property shall be the username when CHAP authentication is specified.
I nitiatorCHAPPassword *(v1.2+)*	string	*rea d-write (null)*	The value of this property shall be the shared secret for Mutual (2-way)CHAP authentication.

Property	Type	Att ributes	Notes
InitiatorCHAPUser *(v1.2+)*	string	*rea d-write (null)*	If present, this property is the initiator CHAP username for Mutual (2-way) authentication. For example, with an iSCSI scenario, use the initiator iQN.
TargetCHAPPassword *(v1.3+)*	string	*rea d-write (null)*	The value of this property shall be the CHAP Secret for 2-way CHAP authentication.
TargetCHAPUser *(v1.2+)*	string	*rea d-write (null)*	The value of this property shall be the Target CHAP Username for Mutual (2-way) CHAP authentication. For example, with an iSCSI scenario, use the target iQN.
TargetPassword *(v1.2+, deprecated v1.3)*	string	*rea d-write (null)*	The value of this property shall be the CHAP Secret for 2-way CHAP authentication. *Deprecated in v1.3 and later. This property is deprecated in favor of TargetCHAPPassword.*

}]

Property	Type	Att ributes	Notes
ClientEndpointGroups [{	array		An array of references to groups of client-side endpoints that may be used to make requests to the storage exposed by this StorageGroup. If null, the implementation may allow access to the storage via any client-side endpoint. If empty, the implementation shall not allow access to the storage via any client-side endpoint.
@odata.id	string (URI)	*rea d-only*	The value of this property shall be the unique identifier for the resource and it shall be of the form defined in the Redfish specification.
}]			
Description	string	*re ad-only (null)*	This property shall contain the description of this resource. The value shall conform with the 'Description' clause of the Redfish Specification.

Property	Type	Att ributes	Notes
DHChapInfo *(v1.3+)* [{	array		The value of this property must reflect the authentication used by this specific endpoint when the authentication type is specificed as DHCHAP. If this endpoint represents an initiator, and AuthenticationMethod is DHCHAP, the Credentials fields LocalDHCHAPAuthSecret and PeerDHCHAPAuthSecret must be used.
L ocalDHCHAPAuthSecret *(v1.3+)*	string	*rea d-write (null)*	This property shall be the local DHCHAP auth secret for DHCHAP authentication.
PeerDHCHAPAuthSecret *(v1.3+)*	string	*rea d-write (null)*	The value of this property shall be the peer DHCHAP auth secret for DHCHAP authentication.
}]			
Id	string	*re ad-only re quired*	This property shall contain the identifier for this resource. The value shall conform with the 'Id' clause of the Redfish Specification.

Property	Type	Att ributes	Notes
Identifier {}	object		The value shall be unique within the managed ecosystem. For property details, see Identifier v1.19.1).
Links {	object		This property shall contain links to other resources that are related to this resource.
ChildStorageGroups [{	array		An array of references to StorageGroups are incorporated into this StorageGroup.
@odata.id	string	*read -write*	Link to another StorageGroup resource.
}]			
ClassOfService {	object		The ClassOfService that all storage in this StorageGroup conforms to. See the *ClassOfService* schema for details on this property.
@odata.id	string	*read -write*	Link to a ClassOfService resource. See the Links section and the *ClassOfService* schema for details.
}			

Property	Type	Att ributes	Notes
Oem {}	object		This property shall contain the OEM extensions. All values for properties contained in this object shall conform to the Redfish S pecification-described requirements. For property details, see Oem.
• *ParentStorageGroups** [{	array		An array of references to StorageGroups that incorporate this StorageGroup.
@odata.id	string	*rea d-only*	Link to another StorageGroup resource.
}]			
}			
MappedVolumes *(v1.1+)* [{	array		An array of mapped volumes managed by this storage group.
AccessCapability *(v1.4+)*	string (enum)	*re ad-writ e\<br\> {=html} (null)*	Each entry shall specify the storage access capability for this mapped volume. *For the possible property values, see AccessCapability in Property details.*
LogicalUnitNumber	string	*rea d-write (null)*	If present, the value is a SCSI Logical Unit Number for the Volume.

Property	Type	Att ributes	Notes
Volume {	object		The value shall reference a mapped Volume. See the *Volume* schema for details on this property.
@odata.id	string	*read -write*	Link to a Volume resource. See the Links section and the *Volume* schema for details.
}			
}]			
MembersAreConsistent	boolean	*rea d-write (null)*	The value of this property shall be set to true if all members are in a consistent state. The default value for this property is false.
Name	string	*re ad-only re quired*	This property shall contain the name of this resource or array member. The value shall conform with the 'Name' clause of the Redfish Specification.
Oem {}	object		This property shall contain the OEM extensions. All values for properties that this object contains shall conform to the Redfish S pecification-described requirements. For property details, see Oem.

Property	Type	Att ributes	Notes
ReplicaInfo {	object		This property shall describe the replication relationship between this storage group and a corresponding source storage group. See the *StorageReplicaInfo* schema for details on this property.
@odata.id	string	*rea d-only*	Link to a ReplicaInfo resource. See the Links section and the *StorageReplicaInfo* schema for details.
}			
ReplicaTargets *(v1.1.1+)* [{	array		The value shall reference the target replicas that are sourced by this replica.
@odata.id	string (URI)	*rea d-only*	The value of this property shall be the unique identifier for the resource and it shall be of the form defined in the Redfish specification.
}]			

Property	Type	Att ributes	Notes
ServerEndpointGroups [{	array		An array of references to groups of server-side endpoints that may be used to make requests to the storage exposed by this storage group. If null, the implementation may allow access to the storage via any server-side endpoint. If empty, the implementation shall not allow access to the storage via any server-side endpoint.
@odata.id	string (URI)	*rea d-only*	The value of this property shall be the unique identifier for the resource and it shall be of the form defined in the Redfish specification.
}]			
Status {}	object		The property shall contain the status of the StorageGroup. For property details, see Status.
Volumes [{	array		An array of references to volumes managed by this storage group.
@odata.id	string	*read -write*	Link to a Volume resource. See the Links section and the *Volume* schema for details.

Property	Type	Att ributes	Notes
}]			
VolumesAreExposed	boolean	*rea d-write (null)*	The value of this property shall be set to true if storage volumes are exposed to the paths defined by the client and server endpoints. The default value for this property is false.

9.6.30.4 Actions

9.6.30.4.1 ExposeVolumes   Description

Exposes the storage of this group via the target endpoints named in the ServerEnd-pointGroups to the initiator endpoints named in the ClientEndpointGroups. The property VolumesAreExposed shall be set to true when this action is completed.

Action URI

{Base URI of target resource}/Actions/StorageGroup.ExposeVolumes

Action parameters

This action takes no parameters.

9.6.30.4.2 HideVolumes   Description

Hide the storage of this group from the initiator endpoints named in the ClientEnd-pointGroups. The property VolumesAreExposed shall be set to false when this action is completed.

Action URI

{Base URI of target resource}/Actions/StorageGroup.HideVolumes

Action parameters

This action takes no parameters.

9.6.30.5 Property details

9.6.30.5.1 AccessCapability   The defined property values are listed in Table 124. Each entry shall specify the storage access capability for this mapped volume.

Table 124: AccessCapability property values

string	Description
Read	Endpoints are allowed to perform reads from the specified resource.
ReadWrite	Endpoints are allowed to perform reads from and writes to the specified resource.

9.6.30.5.2 AccessState   The defined property values are listed in Table 125. The value of this property shall describe the access characteristics of this storage group. All associated logical units through all aggregated ports shall share this access state.

Table 125: AccessState property values

string	Description
NonOptimized	This value shall indicate each endpoint is in an active and non-optimized state.
Optimized	This value shall indicate each endpoint is in an active and optimized state.
Standby	This value shall indicate each endpoint is in a standby state.
Transitioning	This value shall indicate each endpoint is transitioning to a new state.
Unavailable	This value shall indicate each endpoint is in an unavailable state.

9.6.30.5.3 AuthenticationMethod   The defined property values are listed in Table 126. The value of this property must be what kind of authentication that the endpoints in this StorageGroup understands.

Table 126: AuthenticationMethod property values

string	Description
CHAP	iSCSI Challenge Handshake Authentication Protocol (CHAP) authentication is used. For ChapInfo, the CHAPUser and CHAPPassword properties shall be used when type CHAP is selected.
DHCHAP	Diffie-Hellman Challenge Handshake Authentication Protocol (DHCHAP) is an authentication protocol used in Fibre Channel. When MutualCHAP is selected, DHChapInfo shall be used instead of CHAPInfo, and the LocalDHCHAPAuthSecret and PeerDHCHAPAuthSecret properties shall be used.
MutualCHAP	iSCSI Mutual Challenge Handshake Authentication Protocol (CHAP) authentication is used. For ChapInfo, the InitiatorCHAPUser, InitiatorCHAPPassword, TargetCHAPUser, and TargetCHAPPassword properties shall be used when type MutualCHAP is selected.
None	No authentication is used.

9.6.31 StorageGroupCollection

9.6.31.1 URIs   /redfish/v1/Storage/*{StorageId}*/StorageGroups   /redfish/v1/Storage/*{StorageId}*/Volumes/*{VolumeId}*/StorageGroups   /redfish/v1/StorageServices/*{StorageServiceId}*/StorageGroups   /redfish/v1/StorageServices/*{StorageServiceId}*/Volumes/*{VolumeId}*/StorageGroups

9.6.31.2 Properties   The properties defined for the StorageGroupCollection schema are summarized in Table 127.

Table 127: StorageGroupCollection properties

Property	Type	Att ributes	Notes
Description	string	*re ad-only (null)*	This property shall contain the description of this resource. The value shall conform with the 'Description' clause of the Redfish Specification.
Members [{	array		The value of each member entry shall reference a StorageGroup resource.
@odata.id	string	*rea d-only*	Link to a StorageGroup resource. See the Links section and the *StorageGroup* schema for details.
}]			
Me mbers@odata.nextLink	string (URI)	*rea d-only*	The value of this property shall be a URI to a resource, with the same @odata.type, containing the next set of partial members.
Name	string	*rea d-only*	This property shall contain the name of this resource or array member. The value shall conform with the 'Name' clause of the Redfish Specification.

Property	Type	Att ributes	Notes
Oem {}	object		This property shall contain the OEM extensions. All values for properties contained in this object shall conform to the Redfish S pecification-described requirements. For property details, see Oem.

9.6.32 StoragePool 1.9.1

9.6.32.1 Description   A container of data storage capable of providing capacity conforming to one of its supported classes of service. The storage pool does not support IO to its data storage.

9.6.32.2 URIs   /redfish/v1/Storage/{*StorageId*}/FileSystems/{*FileSystemId*}/ CapacitySources/{*CapacitySourceId*}/ProvidingPools/{*StoragePoolId*}   /redfish/ v1/Storage/{*StorageId*}/StoragePools/{*StoragePoolId*}   /redfish/v1/Storage/{*StorageId*}/StoragePools/{*StoragePoolId*}/AllocatedPools/{*AllocatedPoolId*}   /redfish/ v1/Storage/{*StorageId*}/StoragePools/{*StoragePoolId*}/CapacitySources/{*CapacitySourceId*}/ProvidingPools/{*ProvidingPoolId*}   /redfish/v1/Storage/{*StorageId*}/ Volumes/{*VolumeId*}/AllocatedPools/{*StoragePoolId*}   /redfish/v1/Storage/{*StorageId*}/Volumes/{*VolumeId*}/CapacitySources/{*CapacitySourceId*}/ProvidingPools/ {*StoragePoolId*} /redfish/v1/StorageServices/{*StorageServiceId*}/FileSystems/{*FileSystemId*}/CapacitySources/{*CapacitySourceId*}/ProvidingPools/{*StoragePoolId*}   / redfish/v1/StorageServices/{*StorageServiceId*}/StoragePools/{*StoragePoolId*}   / redfish/v1/StorageServices/{*StorageServiceId*}/StoragePools/{*StoragePoolId*}/AllocatedPools/{*AllocatedPoolId*}   /redfish/v1/StorageServices/{*StorageServiceId*}/ StoragePools/{*StoragePoolId*}/CapacitySources/{*CapacitySourceId*}/ProvidingPools/{*ProvidingPoolId*}   /redfish/v1/StorageServices/{*StorageServiceId*}/Volumes/ {*VolumeId*}/AllocatedPools/{*StoragePoolId*} /redfish/v1/StorageServices/{*StorageServiceId*}/Volumes/{*VolumeId*}/CapacitySources/{*CapacitySourceId*}/ProvidingPools/ {*StoragePoolId*}   /redfish/v1/Systems/{*ComputerSystemId*}/Storage/{*StorageId*}/

FileSystems/*{FileSystemId}*/CapacitySources/*{CapacitySourceId}*/ProvidingPools/
{StoragePoolId}   /redfish/v1/Systems/*{ComputerSystemId}*/Storage/*{StorageId}*/
StoragePools/*{StoragePoolId}*   /redfish/v1/Systems/*{ComputerSystemId}*/Storage/
{StorageId}/StoragePools/*{StoragePoolId}*/AllocatedPools/*{AllocatedPoolId}* /redfish/
v1/Systems/*{ComputerSystemId}*/Storage/*{StorageId}*/StoragePools/*{Storage-
PoolId}*/CapacitySources/*{CapacitySourceId}*/ProvidingPools/*{ProvidingPoolId}*   /
redfish/v1/Systems/*{ComputerSystemId}*/Storage/*{StorageId}*/Volumes/*{VolumeId}*/
AllocatedPools/*{StoragePoolId}*   /redfish/v1/Systems/*{ComputerSystemId}*/Storage/
{StorageId}/Volumes/*{VolumeId}*/CapacitySources/*{CapacitySourceId}*/Providing-
Pools/*{StoragePoolId}*

9.6.32.3 Properties   The properties defined for the StoragePool 1.9.1 schema are
summarized in Table 128.

Table 128: StoragePool 1.9.1 properties

Property	Type	Att ributes	Notes
Actions *(v1.3+)* {	object		The Actions property shall contain the available actions for this resource.
#S toragePool.AddDrives {}	object		This action shall be used to add a drive, or set of drives, to an underlying capacity source for the storage pool. *For more information, see the Actions section below.*

Property	Type	Att ributes	Notes
#Stor agePool.RemoveDrives {}	object		This action shall be used to remove a drive from the StoragePool. This action is targeted at a graceful drive removal process, such as initiating a drive cleanup and data reallocation before drive removal from the pool. The implementation may impose restrictions on the number of drives removed simultaneously. *For more information, see the Actions section below.*
#StoragePool .SetCompressionState {}	object		This action shall be used to set the compression state of the storage pool. This may be both a highly impactful, as well as a long running operation. *For more information, see the Actions section below.*
#StoragePool.S etDeduplicationState {}	object		This action shall be used to set the dedupe state of the storage pool. This may be both a highly impactful, as well as a long running operation. *For more information, see the Actions section below.*

Property	Type	Att ributes	Notes
#StoragePoo l.SetEncryptionState {}	object		This action shall be used to set the encryption state of the storage pool. This may be both a highly impactful, as well as a long running operation. *For more information, see the Actions section below.*
}			
AllocatedPools {	object		The value of this property shall contain a reference to the collection of storage pools allocated from this storage pool. Contains a link to a resource.
@odata.id	string	*rea d-only*	Link to Collection of *StoragePool*. See the StoragePool schema for details.
}			
AllocatedVolumes {	object		The value of this property shall contain a reference to the collection of volumes allocated from this storage pool. Contains a link to a resource.
@odata.id	string	*rea d-only*	Link to Collection of *Volume*. See the Volume schema for details.
}			

Property	Type	Att ributes	Notes
BlockSizeBytes	integer (By)	*r ead-onl y {=html} (null)*	Maximum size in bytes of the blocks which form this Volume. If the block size is variable, then the maximum block size in bytes should be specified. If the block size is unknown or if a block concept is not valid (for example, with Memory), enter a 1.
Capacity {}	object		The value of this property shall provide an information about the actual utilization of the capacity within this storage pool. For property details, see Capacity v1.0.0).
CapacitySources [{	array		Fully or partially consumed storage from a source resource. Each entry shall provide capacity allocation data from a named source resource.
@odata.id	string	*read -write*	Link to a CapacitySource resource. See the Links section and the *CapacitySource* schema for details.
}]			

Property	Type	Att ributes	Notes
ClassesOfService {	object		This property shall contain references to all classes of service supported by this storage pool. Capacity allocated from this storage pool shall conform to one of the referenced classes of service. Contains a link to a resource.
@odata.id	string	*read -write*	Link to Collection of *LineOfService*. See the LineOfService schema for details.
}			
Compressed *(v1.3+, deprecated v1.6)*	boolean	*rea d-write (null)*	This property shall contain a boolean indicator if the StoragePool is currently utilizing compression or not. *Deprecated in v1.6 and later. This property has been deprecated in favor of the IsCompressed and Defa ultCompressionBehavior properties.*
CompressionEnabled *(v1.6+)*	boolean	*rea d-write (null)*	The property shall indicate whether or not compression is enabled on the storage pool.

Property	Type	Att ributes	Notes
Deduplicated *(v1.3+, deprecated v1.6)*	boolean	*rea d-write (null)*	This property shall contain a boolean indicator if the StoragePool is currently utilizing deduplication or not. *Deprecated in v1.6 and later. This property has been deprecated in favor of the IsDeduplicated and DefaultDedupeBehavior properties.*
DeduplicationEnabled *(v1.6+)*	boolean	*rea d-write (null)*	The property shall indicate whether or not deduplication is enabled on the storage pool.
D efaultClassOfService *(v1.2+)* {	object		If present, this property shall reference the default class of service for entities allocated from this storage pool. If the ClassesOfService collection is not empty, then the value of this property shall be one of its entries. If not present, the default class of service of the containing StorageService entity shall be used. See the *ClassOfService* schema for details on this property.

Property	Type	Att ributes	Notes
@odata.id	string	*read -write*	Link to a ClassOfService resource. See the Links section and the *ClassOfService* schema for details.
}			
Defaul tCompressionBehavior *(v1.6+)*	boolean	*rea d-write (null)*	If implemented, this property shall indicate the default dedupe behavior applied to the child resource (E.g., volume or storage pool) created out of the storage pool if the 'Compressed' property is not set on the create request.
DefaultD eduplicationBehavior *(v1.6+)*	boolean	*rea d-write (null)*	If implemented, this property shall indicate the default deduplication behavior applied to the child resource (E.g., volume or storage pool) created out of the storage pool if the 'Deduplicated' property is not set on the create request.

Property	Type	Att ributes	Notes
Defau ltEncryptionBehavior *(v1.6+)*	boolean	*rea d-write (null)*	If implemented, this property shall indicate the default dedupe behavior applied to the child resource (E.g., volume or storage pool) created out of the storage pool if the 'Encrypted' property is not set on the create request.
Description	string	*re ad-only (null)*	This property shall contain the description of this resource. The value shall conform with the 'Description' clause of the Redfish Specification.
Encrypted *(v1.3+, deprecated v1.6)*	boolean	*rea d-write (null)*	This property shall contain a boolean indicator if the StoragePool is currently utilizing encryption or not. *Deprecated in v1.6 and later. This property has been deprecated in favor of the IsEncrypted and Def aultEncryptionBehavior properties.*
EncryptionEnabled *(v1.6+)*	boolean	*rea d-write (null)*	The property shall indicate whether or not encryption is enabled on the storage pool.

Property	Type	Att ributes	Notes
Id	string	*re ad-only* *re quired*	This property shall contain the identifier for this resource. The value shall conform with the 'Id' clause of the Redfish Specification.
Identifier {}	object		The value identifies this resource. The value shall be unique within the managed ecosystem. For property details, see Identifier v1.19.1).
IOStatistics *(v1.2+, deprecated v1.9)* {}	object		The value shall represent IO statistics for this StoragePool. For property details, see IOStatistics. *Deprecated in v1.9 and later. This property has been deprecated in favor of the IOStatistics property in Metrics.*
Links {	object		The Links property, as described by the Redfish Specification, shall contain references to resources that are related to, but not contained by (subordinate to), this resource.

Property	Type	Att ributes	Notes
DedicatedSpareDrives *(v1.2+)* [{	array		The value of this property shall be a reference to the resources that this StoragePool is associated with and shall reference resources of type Drive. This property shall only contain references to Drive entities which are currently assigned as a dedicated spare and are able to support this StoragePool.
@odata.id	string (URI)	*rea d-only*	The value of this property shall be the unique identifier for the resource and it shall be of the form defined in the Redfish specification.

}]

Property	Type	Att ributes	Notes
D efaultClassOfService {	object		If present, this property shall reference the default class of service for entities allocated from this storage pool. If the ClassesOfService collection is not empty, then the value of this property shall be one of its entries. If not present, the default class of service of the containing StorageService entity shall be used. See the *ClassOfService* schema for details on this property.
@odata.id	string	*read -write*	Link to a ClassOfService resource. See the Links section and the *ClassOfService* schema for details.
}			
Oem {}	object		This property shall contain the OEM extensions. All values for properties contained in this object shall conform to the Redfish S pecification-described requirements. For property details, see Oem.

Property	Type	Att ributes	Notes
O wningStorageResource *(v1.4+)* {	object		This shall be a pointer to the Storage resource that owns or contains this StoragePool.
@odata.id	string (URI)	*rea d-only*	The value of this property shall be the unique identifier for the resource and it shall be of the form defined in the Redfish specification.
}			
SpareResourceSets *(v1.2+)* [{	array		Each referenced SpareResourceSet shall contain resources that may be utilized to replace the capacity provided by a failed resource having a compatible type.
@odata.id	string	*read -write*	Link to a SpareResourceSet resource. See the Links section and the *SpareResourceSet* schema for details.
}]			
}			

Property	Type	Att ributes	Notes
LowSpaceWarn ingThresholdPercents [ ]	array (%) (integer, null)	*read -write*	Each time the following value is less than one of the values in the array the LOW_S PACE_THRESHOLD_WARNING event shall be triggered: Across all CapacitySources entries, percent = (SUM(AllocatedBytes) - SUM(ConsumedBytes))/SUM(AllocatedBytes).
MaxBlockSizeBytes *(v1.1.1+)*	integer (By)	*r ead-onl y\<b r\> {=html} (null)*	If present, the value is the maximum block size of an allocated resource. If the block size is unknown or if a block concept is not valid (for example, with Memory), this property shall be NULL.
Metrics *(v1.9+)* {	object	* (null)*	This property shall contain a link to a resource of type StoragePoolMetrics that specifies the metrics for this storage pool. IO metrics are reported in the IOStatistics property. See the *StoragePoolMetrics* schema for details on this property.

Property	Type	Att ributes	Notes
@odata.id	string	*rea d-only*	Link to a StoragePoolMetrics resource. See the Links section and the *StoragePoolMetrics* schema for details.
}			
Name	string	*re ad-only* *re quired*	This property shall contain the name of this resource or array member. The value shall conform with the 'Name' clause of the Redfish Specification.
NVMeEndu ranceGroupProperties *(v1.4+)* {	object	* (null)*	This property shall contain properties to use when StoragePool is used to describe an NVMe Endurance Group.
EndGrpLifetime *(v1.4+)* {	object	* (null)*	This property shall contain any Endurance Group Lifetime properties.

Property	Type	Att ributes	Notes
DataUnitsRead *(v1.4+)*	integer	*re ad-only (null)*	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.
DataUnitsWritten *(v1.4+)*	integer	*re ad-only (null)*	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.

Property	Type	Att ributes	Notes
EnduranceEstimate *(v1.4+)*	integer	*re ad-only (null)*	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.
ErrorInfo rmationLogEntryCount *(v1.4+)*	integer	*re ad-only (null)*	This property shall contain the number of error information log entries over the life of the controller for the endurance group.
HostReadCommand-Count *(v1.4+)*	integer	*re ad-only (null)*	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.

Property	Type	Att ributes	Notes
H ostWriteCommand- Count *(v1.4+)*	integer	*re ad-only (null)*	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.
MediaAndDat aIntegrityErrorCount *(v1.4+)*	integer	*re ad-only (null)*	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.

Property	Type	Att ributes	Notes
MediaUnitsWritten (v1.4+)	integer	*re ad-only (null)*	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.
PercentUsed (v1.4+)	integer	*re ad-only (null)*	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.

Property	Type	Att ributes	Notes
}			
Predicted MediaLifeLeftPercent *(v1.4+)*	number (%)	*r ead-onl y {=html} (null)*	This property shall contain an indicator of the percentage of life remaining in the drive's media.
}			
NVMeProperties *(v1.6+)* {	object	* (null)*	The property shall indicate the type of storage pool.
NVMePoolType *(v1.6+)*	string (enum)	*r ead-onl y {=html} (null)*	This property shall indicate whether the StoragePool is used as an EnduranceGroup or an NVMSet. *For the possible property values, see NVMePoolType in Property details.*
}			
NVMeSetProperties *(v1.4+)* {	object	* (null)*	This property shall contain properties to use when StoragePool is used to describe an NVMe Set.
Endu ranceGroupIdentifier *(v1.4+)*	string	*re ad-only (null)*	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. Pattern: `^0[xX ](([a−fA−F]|[0−9])*)$`

Property	Type	Att ributes	Notes
O ptimalWriteSizeBytes *(v1.4+)*	integer (By)	*r ead-onl y {=html} (null)*	This property shall contain the Optimal Write Size in Bytes for this NVMe Set.
Random4kRe adTypicalNanoSeconds *(v1.4+)*	integer	*re ad-only (null)*	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.
SetIdentifier *(v1.4+)*	string	*re ad-only (null)*	This property shall contain a 16-bit hex value that contains the NVMe Set group identifier. The NVM Set identifier is unique within a subsystem. Reserved values include 0. Pattern: `^0[xX ](([a-fA-F]|[0-9])*)$`
UnallocatedNVMNa mespaceCapacityBytes *(v1.4+)*	integer (By)	*r ead-onl y {=html} (null)*	This property shall contain the unallocated capacity of the NVMe Set in bytes.

}

Property	Type	Att ributes	Notes
Oem {}	object		This property shall contain the OEM extensions. All values for properties that this object contains shall conform to the Redfish S pecification-described requirements. For property details, see Oem.
PoolType *(v1.6+, deprecated v1.7)* [ ]	array (string (enum))	*r ead-onl y {=html} (null)*	The property shall indicate the type of storage pool. *For the possible property values, see PoolType in Property details. Deprecated in v1.7 and later. This property has been deprecated in favor of the SupportedPoolTypes property.*
Recoverabl eCapacitySourceCount *(v1.2+)*	integer	*rea d-write (null)*	The value is the number of available capacity source resources currently available in the event that an equivalent capacity source resource fails.

Property	Type	Att ributes	Notes
Rema iningCapacityPercent *(v1.1+)*	integer	*re ad-only (null)*	If present, this value shall return {[(SUM(AllocatedBytes) - SU M(ConsumedBytes)]/SUM (AllocatedBytes)}*100 represented as an integer value.
ReplicationEnabled *(v1.8+)*	boolean	*rea d-write (null)*	The property shall indicate whether or not replication is enabled on the storage pool. If enabled for pool, replication can still be disabled on individual resources (e.g., volumes) within the pool.
Status {}	object		The property shall contain the status of the StoragePool. For property details, see Status.
SupportedPoolTypes *(v1.7+)* [ ]	array (string (enum))	*r ead-onl y {=html} (null)*	This collection shall contain all the PoolType values supported by the storage pool. *For the possible property values, see SupportedPoolTypes in Property details.*

Property	Type	Att ributes	Notes
Supported ProvisioningPolicies *(v1.3+)* [ ]	array (string (enum))	*re ad-write {=html} (null)*	This collection shall specify all supported storage allocation policies for the Storage Pool. *For the possible property values, see Support edProvisioningPolicies in Property details.*
SupportedRAIDTypes *(v1.3+)* [ ]	array (string (enum))	*r ead-only {=html} (null)*	This collection shall contain all the RAIDType values supported by the storage pool. *For the possible property values, see SupportedRAIDTypes in Property details.*

9.6.32.4 Actions

9.6.32.4.1 AddDrives   Description

This action shall be used to add a drive, or set of drives, to an underlying capacity source for the storage pool.

Action URI

{Base URI of target resource}/Actions/StoragePool.AddDrives

Action parameters

The parameters for the action which are included in the POST body to the URI shown in the 'target' property of the Action are summarized in Table 129.

Table 129: AddDrives action parameters

Parameter Name	Type	Attributes	Notes
CapacitySource {	object	*optional*	This parameter shall contain the target capacity source for the drive(s). This property does not need to be specified if the storage pool only contains one capacity source, or if the implementation is capable of automatically selecting the appropriate capacity source. See the *CapacitySource* schema for details on this property.
@odata.id	string	*read-only*	Link to a CapacitySource resource. See the Links section and the *CapacitySource* schema for details.
}			

Parameter Name	Type	Attributes	Notes
Drives [{	array	*required*	This parameter shall contain the Uri to the existing drive or drives to be added to a capacity source of the storage pool. The implementation may impose restrictions on the number of drives added s imultaneously.
@odata.id	string (URI)	*read-only*	The value of this property shall be the unique identifier for the resource and it shall be of the form defined in the Redfish specification.
}]			

9.6.32.4.2 RemoveDrives   Description

This action shall be used to remove a drive from the StoragePool. This action is targeted at a graceful drive removal process, such as initiating a drive cleanup and data reallocation before drive removal from the pool. The implementation may impose restrictions on the number of drives removed simultaneously.

Action URI

{Base URI of target resource}/Actions/StoragePool.RemoveDrives

Action parameters

The parameters for the action which are included in the POST body to the URI shown in the 'target' property of the Action are summarized in Table 130.

Table 130: RemoveDrives action parameters

Parameter Name	Type	Attributes	Notes
Drives [{	array	*required*	This parameter shall contain the Uri to the drive or drives to be removed from the underlying capacity source.
@odata.id	string (URI)	*read-only*	The value of this property shall be the unique identifier for the resource and it shall be of the form defined in the Redfish specification.
}]			

9.6.32.4.3 SetCompressionState   Description

This action shall be used to set the compression state of the storage pool. This may be both a highly impactful, as well as a long running operation.

Action URI

{Base URI of target resource}/Actions/StoragePool.SetCompressionState

Action parameters

The parameters for the action which are included in the POST body to the URI shown in the 'target' property of the Action are summarized in Table 131.

Table 131: SetCompressionState action parameters

Parameter Name	Type	Attributes	Notes
Enable	boolean	*required*	This property shall indicate the desired compression state of the storage pool.

9.6.32.4.4 SetDeduplicationState   Description

This action shall be used to set the dedupe state of the storage pool. This may be both a highly impactful, as well as a long running operation.

Action URI

{Base URI of target resource}/Actions/StoragePool.SetDeduplicationState

Action parameters

The parameters for the action which are included in the POST body to the URI shown in the 'target' property of the Action are summarized in Table 132.

Table 132: SetDeduplicationState action parameters

Parameter Name	Type	Attributes	Notes
Enable	boolean	*required*	This property shall indicate the desired deduplication state of the storage pool.

9.6.32.4.5 SetEncryptionState   Description

This action shall be used to set the encryption state of the storage pool. This may be both a highly impactful, as well as a long running operation.

Action URI

{Base URI of target resource}/Actions/StoragePool.SetEncryptionState

Action parameters

The parameters for the action which are included in the POST body to the URI shown in the 'target' property of the Action are summarized in Table 133.

Table 133: SetEncryptionState action parameters

Parameter Name	Type	Attributes	Notes
Enable	boolean	*required*	This property shall indicate the desired encryption state of the storage pool.

9.6.32.5 Property details

9.6.32.5.1 NVMePoolType   The defined property values are listed in Table 134. This property shall indicate whether the StoragePool is used as an EnduranceGroup or an NVMSet.

Table 134: NVMePoolType property values

string	Description
EnduranceGroup	This type shall be used to specify a pool of type EnduranceGroup, used by NVMe devices.
NVMSet	This type shall be used to specify a pool of type NVMSet, used by NVMe devices.

9.6.32.5.2 PoolType   The defined property values are listed in Table 135. The property shall indicate the type of storage pool.

Table 135: PoolType property values

string	Description
Block	This type shall be used to specify a pool of type block. This is used when the pool serves block storage.
File	This type shall be used to specify a pool of type file. This setting is used when the pool serves file storage.
Object	This type shall be used to specify a pool of type object.
Pool	This type shall be used to specify a pool of type pool. This setting is used to indicate a 'pool of pools' hierarchy.

9.6.32.5.3 SupportedPoolTypes   The defined property values are listed in Table 136. This collection shall contain all the PoolType values supported by the storage pool.

Table 136: SupportedPoolTypes property values

string	Description
Block	This type shall be used to specify a pool of type block. This is used when the pool serves block storage.
File	This type shall be used to specify a pool of type file. This setting is used when the pool serves file storage.
Object	This type shall be used to specify a pool of type object.
Pool	This type shall be used to specify a pool of type pool. This setting is used to indicate a 'pool of pools' hierarchy.

9.6.32.5.4 SupportedProvisioningPolicies The defined property values are listed in Table 137. This collection shall specify all supported storage allocation policies for the Storage Pool.

Table 137: SupportedProvisioningPolicies property values

string	Description
Fixed	This enumeration literal specifies storage shall be fully allocated.
Thin	This enumeration literal specifies storage may be over allocated.

9.6.32.5.5 SupportedRAIDTypes The defined property values are listed in Table 138. This collection shall contain all the RAIDType values supported by the storage pool.

Table 138: SupportedRAIDTypes property values

string	Description
None	A placement policy with no redundancy at the device level.
RAID0	A placement policy where consecutive logical blocks of data are uniformly distributed across a set of independent storage devices without offering any form of redundancy. This is commonly referred to as data striping. This form of RAID will encounter data loss with the failure of any storage device in the set.
RAID00	A placement policy that creates a RAID 0 stripe set over two or more RAID 0 sets. This is commonly referred to as RAID 0+0. This form of data layout is not fault tolerant; if any storage device fails there will be data loss.

string	Description
RAID01	A data placement policy that creates a mirrored device (RAID 1) over a set of striped devices (RAID 0). This is commonly referred to as RAID 0+1 or RAID 0/1. Data stored using this form of RAID is able to survive a single RAID 0 data set failure without data loss.
RAID1	A placement policy where each logical block of data is stored on more than one independent storage device. This is commonly referred to as mirroring. Data stored using this form of RAID is able to survive a single storage device failure without data loss.
RAID10	A placement policy that creates a striped device (RAID 0) over a set of mirrored devices (RAID 1). This is commonly referred to as RAID 1/0. Data stored using this form of RAID is able to survive storage device failures in each RAID 1 set without data loss.
RAID10E	A placement policy that uses a RAID 0 stripe set over two or more RAID 10 sets. This is commonly referred to as Enhanced RAID 10. Data stored using this form of RAID is able to survive a single device failure within each nested RAID 1 set without data loss.
RAID10Triple	A placement policy that uses a striped device (RAID 0) over a set of triple mirrored devices (RAID 1Triple). This form of RAID can survive up to two failures in each triple mirror set without data loss.

string	Description
RAID1E	A placement policy that uses a form of mirroring implemented over a set of independent storage devices where logical blocks are duplicated on a pair of independent storage devices so that data is uniformly distributed across the storage devices. This is commonly referred to as RAID 1 Enhanced. Data stored using this form of RAID is able to survive a single storage device failure without data loss.
RAID1Triple	A placement policy where each logical block of data is mirrored three times across a set of three independent storage devices. This is commonly referred to as three-way mirroring. This form of RAID can survive two device failures without data loss.
RAID3	A placement policy using parity-based protection where logical bytes of data are uniformly distributed across a set of independent storage devices and where the parity is stored on a dedicated independent storage device. Data stored using this form of RAID is able to survive a single storage device failure without data loss. If the storage devices use rotating media, they are assumed to be rotationally synchronized, and the data stripe size should be no larger than the exported block size.
RAID4	A placement policy using parity-based protection where logical blocks of data are uniformly distributed across a set of independent storage devices and where the parity is stored on a dedicated independent storage device. Data stored using this form of RAID is able to survive a single storage device failure without data loss.

string	Description
RAID5	A placement policy using parity-based protection for storing stripes of 'n' logical blocks of data and one logical block of parity across a set of 'n+1' independent storage devices where the parity and data blocks are interleaved across the storage devices. Data stored using this form of RAID is able to survive a single storage device failure without data loss.
RAID50	A placement policy that uses a RAID 0 stripe set over two or more RAID 5 sets of independent storage devices. Data stored using this form of RAID is able to survive a single storage device failure within each RAID 5 set without data loss.
RAID6	A placement policy using parity-based protection for storing stripes of 'n' logical blocks of data and two logical blocks of independent parity across a set of 'n+2' independent storage devices where the parity and data blocks are interleaved across the storage devices. Data stored using this form of RAID is able to survive any two independent storage device failures without data loss.
RAID60	A placement policy that uses a RAID 0 stripe set over two or more RAID 6 sets of independent storage devices. Data stored using this form of RAID is able to survive two device failures within each RAID 6 set without data loss.

string	Description
RAID6TP	A placement policy that uses parity-based protection for storing stripes of 'n' logical blocks of data and three logical blocks of independent parity across a set of 'n+3' independent storage devices where the parity and data blocks are interleaved across the storage devices. This is commonly referred to as Triple Parity RAID. Data stored using this form of RAID is able to survive any three independent storage device failures without data loss.

9.6.33 StoragePoolCollection

9.6.33.1 URIs   /redfish/v1/Storage/*{StorageId}*/FileSystems/*{FileSystemId}*/
CapacitySources/*{CapacitySourceId}*/ProvidingPools   /redfish/v1/Storage/*{StorageId}*/StoragePools /redfish/v1/Storage/*{StorageId}*/StoragePools/*{StoragePoolId}*/
AllocatedPools   /redfish/v1/Storage/*{StorageId}*/StoragePools/*{StoragePoolId}*/
CapacitySources/*{CapacitySourceId}*/ProvidingPools   /redfish/v1/Storage/*{StorageId}*/Volumes/*{VolumeId}*/AllocatedPools   /redfish/v1/Storage/*{StorageId}*/Volumes/*{VolumeId}*/CapacitySources/*{CapacitySourceId}*/ProvidingPools   /redfish/v1/
StorageServices/*{StorageServiceId}*/FileSystems/*{FileSystemId}*/CapacitySources/
{CapacitySourceId}/ProvidingPools   /redfish/v1/StorageServices/*{StorageServiceId}*/StoragePools   /redfish/v1/StorageServices/*{StorageServiceId}*/StoragePools/
{StoragePoolId}/AllocatedPools   /redfish/v1/StorageServices/*{StorageServiceId}*/
StoragePools/*{StoragePoolId}*/CapacitySources/*{CapacitySourceId}*/Providing-
Pools   /redfish/v1/StorageServices/*{StorageServiceId}*/Volumes/*{VolumeId}*/Allo-
catedPools   /redfish/v1/StorageServices/*{StorageServiceId}*/Volumes/*{VolumeId}*/
CapacitySources/*{CapacitySourceId}*/ProvidingPools   /redfish/v1/Systems/*{ComputerSystemId}*/Storage/*{StorageId}*/FileSystems/*{FileSystemId}*/CapacitySources/
{CapacitySourceId}/ProvidingPools   /redfish/v1/Systems/*{ComputerSystemId}*/
Storage/*{StorageId}*/StoragePools   /redfish/v1/Systems/*{ComputerSystemId}*/
Storage/*{StorageId}*/StoragePools/*{StoragePoolId}*/AllocatedPools   /redfish/v1/
Systems/*{ComputerSystemId}*/Storage/*{StorageId}*/StoragePools/*{StoragePoolId}*/
CapacitySources/*{CapacitySourceId}*/ProvidingPools   /redfish/v1/Systems/*{ComputerSystemId}*/Storage/*{StorageId}*/Volumes/*{VolumeId}*/AllocatedPools   /redfish/

v1/Systems/*{ComputerSystemId}*/Storage/*{StorageId}*/Volumes/*{VolumeId}*/Capaci-
tySources/*{CapacitySourceId}*/ProvidingPools

9.6.33.2 Properties The properties defined for the StoragePoolCollection
schema are summarized in Table 139.

Table 139: StoragePoolCollection properties

Property	Type	Att ributes	Notes
Description	string	*re ad-only (null)*	This property shall contain the description of this resource. The value shall conform with the 'Description' clause of the Redfish Specification.
Members [{	array		The value of each member entry shall reference a StoragePool resource.
@odata.id	string	*rea d-only*	Link to a StoragePool resource. See the Links section and the *StoragePool* schema for details.
}]			
Me mbers@odata.nextLink	string (URI)	*rea d-only*	The value of this property shall be a URI to a resource, with the same @odata.type, containing the next set of partial members.

Property	Type	Att ributes	Notes
Name	string	*rea d-only*	This property shall contain the name of this resource or array member. The value shall conform with the 'Name' clause of the Redfish Specification.
Oem {}	object		This property shall contain the OEM extensions. All values for properties contained in this object shall conform to the Redfish S pecification-described requirements. For property details, see Oem.

9.6.34 StoragePoolMetrics 1.0.1

9.6.34.1 Description   The StoragePoolMetrics schema shall contain the usage and health statistics for a storage pool in a Redfish implementation.

9.6.34.2 URIs   /redfish/v1/Storage/*{StorageId}*/FileSystems/*{FileSystemId}*/Ca-
pacitySources/*{CapacitySourceId}*/ProvidingPools/*{StoragePoolId}*/Metrics /redfish/
v1/Storage/*{StorageId}*/StoragePools/*{StoragePoolId}*/AllocatedPools/*{Allocated-
PoolId}*/Metrics   /redfish/v1/Storage/*{StorageId}*/StoragePools/*{StoragePoolId}*/
CapacitySources/*{CapacitySourceId}*/ProvidingPools/*{ProvidingPoolId}*/Metrics   /
redfish/v1/Storage/*{StorageId}*/StoragePools/*{StoragePoolId}*/Metrics   /redfish/v1/
Storage/*{StorageId}*/Volumes/*{VolumeId}*/AllocatedPools/*{StoragePoolId}*/Metrics
/redfish/v1/Storage/*{StorageId}*/Volumes/*{VolumeId}*/CapacitySources/*{Capaci-
tySourceId}*/ProvidingPools/*{StoragePoolId}*/Metrics   /redfish/v1/StorageServices/
{StorageServiceId}/FileSystems/*{FileSystemId}*/CapacitySources/*{CapacitySourceId}*/
ProvidingPools/*{StoragePoolId}*/Metrics   /redfish/v1/StorageServices/*{StorageSer-
viceId}*/StoragePools/*{StoragePoolId}*/AllocatedPools/*{AllocatedPoolId}*/Metrics
/redfish/v1/StorageServices/*{StorageServiceId}*/StoragePools/*{StoragePoolId}*/

CapacitySources/*{CapacitySourceId}*/ProvidingPools/*{ProvidingPoolId}*/Metrics /redfish/v1/StorageServices/*{StorageServiceId}*/StoragePools/*{StoragePoolId}*/Metrics / redfish/v1/StorageServices/*{StorageServiceId}*/Volumes/*{VolumeId}*/AllocatedPools/ *{StoragePoolId}*/Metrics /redfish/v1/StorageServices/*{StorageServiceId}*/Volumes/ *{VolumeId}*/CapacitySources/*{CapacitySourceId}*/ProvidingPools/*{StoragePoolId}*/ Metrics /redfish/v1/Systems/*{ComputerSystemId}*/Storage/*{StorageId}*/FileSystems/*{FileSystemId}*/CapacitySources/*{CapacitySourceId}*/ProvidingPools/*{StoragePoolId}*/Metrics /redfish/v1/Systems/*{ComputerSystemId}*/Storage/*{StorageId}*/ StoragePools/*{StoragePoolId}*/AllocatedPools/*{AllocatedPoolId}*/Metrics /redfish/v1/ Systems/*{ComputerSystemId}*/Storage/*{StorageId}*/StoragePools/*{StoragePoolId}*/ CapacitySources/*{CapacitySourceId}*/ProvidingPools/*{ProvidingPoolId}*/Metrics / redfish/v1/Systems/*{ComputerSystemId}*/Storage/*{StorageId}*/StoragePools/*{StoragePoolId}*/Metrics /redfish/v1/Systems/*{ComputerSystemId}*/Storage/*{StorageId}*/ Volumes/*{VolumeId}*/AllocatedPools/*{StoragePoolId}*/Metrics /redfish/v1/Systems/ *{ComputerSystemId}*/Storage/*{StorageId}*/Volumes/*{VolumeId}*/CapacitySources/ *{CapacitySourceId}*/ProvidingPools/*{StoragePoolId}*/Metrics

9.6.34.3 Properties The properties defined for the StoragePoolMetrics 1.0.1 schema are summarized in Table 140.

Table 140: StoragePoolMetrics 1.0.1 properties

Property	Type	Att ributes	Notes
Actions {}	object		This property shall contain the available actions for this resource.
Consis tencyCheckErrorCount	number	*re ad-only (null)*	This property shall contain the number of consistency check errors over the lifetime of the storage pool.
Correct ableIOReadErrorCount	integer	*re ad-only (null)*	This property shall contain the number of the correctable read errors for the lifetime of the storage pool.

Property	Type	Att ributes	Notes
Correcta bleIOWriteErrorCount	integer	*re ad-only (null)*	This property shall contain the number of the correctable write errors for the lifetime of the storage pool.
Description	string	*re ad-only (null)*	This property shall contain the description of this resource. The value shall conform with the 'Description' clause of the Redfish Specification.
Id	string	*re ad-only re quired*	This property shall contain the identifier for this resource. The value shall conform with the 'Id' clause of the Redfish Specification.
IOStatistics {}	object		The value shall represent IO statistics for this storage pool. For property details, see IOStatistics.
Name	string	*re ad-only re quired*	This property shall contain the name of this resource or array member. The value shall conform with the 'Name' clause of the Redfish Specification.

Property	Type	Att ributes	Notes
Oem {}	object		This property shall contain the OEM extensions. All values for properties that this object contains shall conform to the Redfish S pecification-described requirements. For property details, see Oem.
RebuildErrorCount	number	*re ad-only (null)*	This property shall contain the number of rebuild errors over the lifetime of the storage pool.
StateChangeCount	number	*re ad-only (null)*	This property shall contain the number of state changes (changes in Status.State) for this storage pool.
Uncorrect ableIOReadErrorCount	integer	*re ad-only (null)*	This property shall contain the number of the uncorrectable read errors for the lifetime of the storage pool.
Uncorrecta bleIOWriteErrorCount	integer	*re ad-only (null)*	This property shall contain the number of the uncorrectable write errors for the lifetime of the storage pool.

9.6.35 StorageReplicaInfo 1.4.0

9.6.35.1 Description   This entity shall define the characteristics of a replica.

9.6.35.2 Properties   The properties defined for the StorageReplicaInfo 1.4.0 schema are summarized in Table 141.

Table 141: StorageReplicaInfo 1.4.0 properties

Property	Type	Att ributes	Notes
Actions *(v1.2+)* {}	object		The Actions property shall contain the available actions for this resource.
Description	string	*re ad-only (null)*	This property shall contain the description of this resource. The value shall conform with the 'Description' clause of the Redfish Specification.
Id	string	*re ad-only re quired*	This property shall contain the identifier for this resource. The value shall conform with the 'Id' clause of the Redfish Specification.
Name	string	*re ad-only re quired*	This property shall contain the name of this resource or array member. The value shall conform with the 'Name' clause of the Redfish Specification.

Property	Type	Att ributes	Notes
Oem {}	object		This property shall contain the OEM extensions. All values for properties that this object contains shall conform to the Redfish S pecification-described requirements. For property details, see Oem.

9.6.36 StorageService 1.7.0

9.6.36.1 Description   Collection of resources that the system can make available to one or more host systems. The collection can contain: block, file, or object storage; local system access points through which the collection is made available; hosts, or host access points to which the collection is made available.

9.6.36.2 URIs   /redfish/v1/StorageServices/*{StorageServiceId}*   /redfish/v1/Systems/*{ComputerSystemId}*/StorageServices/*{StorageServiceId}*

9.6.36.3 Properties   The properties defined for the StorageService 1.7.0 schema are summarized in Table 142.

Table 142: *StorageService 1.7.0 properties*

Property	Type	Att ributes	Notes
Actions {	object		The Actions property shall contain the available actions for this resource.

Property	Type	Att ributes	Notes
#StorageServ ice.SetEncryptionKey {} }	object		This defines the name of the custom action supported on this resource. *For more information, see the Actions section below.*
ClassesOfService {	object		The value of each entry in the array shall reference a ClassOfService supported by this service. Contains a link to a resource.
@odata.id }	string	*read -write*	Link to Collection of *LineOfService*. See the LineOfService schema for details.
ClientEndpointGroups {}	object		The value of each entry in the array shall reference an EndpointGroup.
Connections *(v1.6+)* {}	object		The value of this property shall contain references to all Connections that include this volume.
ConsistencyGroups *(v1.3+)* {	object		The value of each entry in the array shall reference a ConsistencyGroup. Contains a link to a resource.
@odata.id	string	*read -write*	Link to Collection of *ConsistencyGroup*. See the ConsistencyGroup schema for details.

Property	Type	Att ributes	Notes
}			
DataProte ctionLoSCapabilities *(v1.2+)* {	object		The value shall reference the data protection capabilities of this service. See the *DataProt ectionLoSCapabilities* schema for details on this property.
@odata.id	string	*read -write*	Link to a DataPro tectionLoSCapabilities resource. See the Links section and the *DataProt ectionLoSCapabilities* schema for details.
}			
DataSec urityLoSCapabilities *(v1.2+)* {	object		The value shall reference the data security capabilities of this service. See the *DataSe curityLoSCapabilities* schema for details on this property.
@odata.id	string	*read -write*	Link to a DataS ecurityLoSCapabilities resource. See the Links section and the *DataSe curityLoSCapabilities* schema for details.
}			

Property	Type	Att ributes	Notes
DataSt orageLoSCapabilities *(v1.2+)* {	object		The value shall reference the data storage capabilities of this service. See the *DataS torageLoSCapabilities* schema for details on this property.
@odata.id	string	*read -write*	Link to a Data StorageLoSCapabilities resource. See the Links section and the *DataS torageLoSCapabilities* schema for details.
}			
D efaultClassOfService *(v1.2+)* {	object		If present, this property shall reference the default class of service for entities allocated by this storage service. This default may be overridden by the DefaultClassOfService property values within contained StoragePools. See the *ClassOfService* schema for details on this property.
@odata.id	string	*read -write*	Link to a ClassOfService resource. See the Links section and the *ClassOfService* schema for details.
}			

Property	Type	Att ributes	Notes
Description	string	*re ad-only (null)*	This property shall contain the description of this resource. The value shall conform with the 'Description' clause of the Redfish Specification.
Drives {}	object		A collection that indicates all the drives managed by this storage service.
EndpointGroups {}	object		The value of each entry in the array shall reference an EndpointGroup.
Endpoints {}	object		The value of each entry in the array shall reference an Endpoint managed by this service.
FileSystems {	object		An array of references to FileSystems managed by this storage service. Contains a link to a resource.
@odata.id	string	*read -write*	Link to Collection of *FileSystem*. See the FileSystem schema for details.
}			
Id	string	*re ad-only re quired*	This property shall contain the identifier for this resource. The value shall conform with the 'Id' clause of the Redfish Specification.

Property	Type	Att ributes	Notes
Identifier {}	object		The value identifies this resource. The value shall be unique within the managed ecosystem. For property details, see Identifier v1.19.1).
IOConnect ivityLoSCapabilities *(v1.2+)* {	object		The value shall reference the IO connectivity capabilities of this service. See the *IOConnec tivityLoSCapabilities* schema for details on this property.
@odata.id	string	*read -write*	Link to a IOConne ctivityLoSCapabilities resource. See the Links section and the *IOConnec tivityLoSCapabilities* schema for details.
}			
IOPerfor manceLoSCapabilities *(v1.2+)* {	object		The value shall reference the IO performance capabilities of this service. See the *IOPerfo rmanceLoSCapabilities* schema for details on this property.
@odata.id	string	*read -write*	Link to a IOPerf ormanceLoSCapabilities resource. See the Links section and the *IOPerfo rmanceLoSCapabilities* schema for details.
}			

Property	Type	Att ributes	Notes
IOStatistics *(v1.2+, deprecated v1.7)* {}	object		The value shall represent IO statistics for this StorageService. For property details, see IOStatistics. *Deprecated in v1.7 and later. This property is deprecated in favor of the IOStatistics property in S torageServiceMetrics.*
LinesOfService *(v1.4+)* [{	array		The value of each entry shall reference a LineOfService collection defined for this service.
@odata.id	string	*read -write*	Link to Collection of *LineOfService*. See the LineOfService schema for details.
}]			
Links {	object		This property shall contain links to other resources that are related to this resource.
DataProte ctionLoSCapabilities {	object		The value shall reference the data protection capabilities of this service. See the *DataProt ectionLoSCapabilities* schema for details on this property.

Property	Type	Att ributes	Notes
@odata.id	string	*read -write*	Link to a DataPro tectionLoSCapabilities resource. See the Links section and the *DataProt ectionLoSCapabilities* schema for details.
}			
DataSec urityLoSCapabilities {	object		The value shall reference the data security capabilities of this service. See the *DataSe curityLoSCapabilities* schema for details on this property.
@odata.id	string	*read -write*	Link to a DataS ecurityLoSCapabilities resource. See the Links section and the *DataSe curityLoSCapabilities* schema for details.
}			
DataSt orageLoSCapabilities {	object		The value shall reference the data storage capabilities of this service. See the *DataS torageLoSCapabilities* schema for details on this property.
@odata.id	string	*read -write*	Link to a Data StorageLoSCapabilities resource. See the Links section and the *DataS torageLoSCapabilities* schema for details.

Property	Type	Att ributes	Notes
}			
D efaultClassOfService {	object		If present, this property shall reference the default class of service for entities allocated by this storage service. This default may be overridden by the DefaultClassOfService property values within contained StoragePools. See the *ClassOfService* schema for details on this property.
@odata.id	string	*read -write*	Link to a ClassOfService resource. See the Links section and the *ClassOfService* schema for details.
}			
HostingSystem {}	object		The value shall reference the ComputerSystem or StorageController that hosts this service.
IOConnect ivityLoSCapabilities {	object		The value shall reference the IO connectivity capabilities of this service. See the *IOConnec tivityLoSCapabilities* schema for details on this property.

Property	Type	Att ributes	Notes
@odata.id	string	*read -write*	Link to a IOConne ctivityLoSCapabilities resource. See the Links section and the *IOConnec tivityLoSCapabilities* schema for details.
}			
IOPerfor manceLoSCapabilities {	object		The value shall reference the IO performance capabilities of this service. See the *IOPerfo rmanceLoSCapabilities* schema for details on this property.
@odata.id	string	*read -write*	Link to a IOPerf ormanceLoSCapabilities resource. See the Links section and the *IOPerfo rmanceLoSCapabilities* schema for details.
}			
Oem {}	object		This property shall contain the OEM extensions. All values for properties contained in this object shall conform to the Redfish S pecification-described requirements. For property details, see Oem.
}			

Property	Type	Att ributes	Notes
Metrics *(v1.7+)* {	object	* (null)*	This property shall contain a link to a resource of type StorageServiceMetrics that specifies the metrics for this storage service. IO metrics are reported in the IOStatistics property. See the *StorageServiceMetrics* schema for details on this property.
@odata.id	string	*rea d-only*	Link to a StorageServiceMetrics resource. See the Links section and the *StorageServiceMetrics* schema for details.
}			
Name	string	*re ad-only* *re quired*	This property shall contain the name of this resource or array member. The value shall conform with the 'Name' clause of the Redfish Specification.
Oem {}	object		This property shall contain the OEM extensions. All values for properties that this object contains shall conform to the Redfish S pecification-described requirements. For property details, see Oem.

Property	Type	Att ributes	Notes
Redundancy [{	array		This collection shall contain the redundancy information for the storage subsystem.
@odata.id	string (URI)	*rea d-only*	The value of this property shall be the unique identifier for the resource and it shall be of the form defined in the Redfish specification.
}]			
ServerEndpointGroups {}	object		The value of each entry in the array shall reference a EndpointGroup.
SpareResourceSets *(v1.2+)* [{	array		Each contained SpareResourceSet shall contain resources that may be utilized to replace the capacity provided by a failed resource having a compatible type.
@odata.id	string	*read -write*	Link to a SpareResourceSet resource. See the Links section and the *SpareResourceSet* schema for details.
}]			
Status {}	object		The property shall contain the status of the StorageService. For property details, see Status.

Property	Type	Att ributes	Notes
StorageGroups *(deprecated v1.6)* {	object		The value of each entry in the array shall reference a StorageGroup. Contains a link to a resource. *Deprecated in v1.6 and later. This property is deprecated in favor of the Connections property.*
@odata.id	string	*rea d-only*	Link to Collection of *StorageGroup*. See the StorageGroup schema for details.
}			
StoragePools {	object		An array of references to StoragePools. Contains a link to a resource.
@odata.id	string	*rea d-only*	Link to Collection of *StoragePool*. See the StoragePool schema for details.
}			
StorageSubsystems *(v1.0.1+)* [{	array		The value shall be a link to a collection of type StorageCollection having members that represent storage subsystems managed by this storage service.
@odata.id	string (URI)	*rea d-only*	The value of this property shall be the unique identifier for the resource and it shall be of the form defined in the Redfish specification.

Property	Type	Att ributes	Notes
}]			
Volumes {	object		An array of references to Volumes managed by this storage service. Contains a link to a resource.
@odata.id	string	*read -write*	Link to Collection of *Volume*. See the Volume schema for details.
}			

9.6.36.4 Actions

9.6.36.4.1 SetEncryptionKey   Description

This defines the name of the custom action supported on this resource.

Action URI

{Base URI of target resource}/Actions/StorageService.SetEncryptionKey

Action parameters

The parameters for the action which are included in the POST body to the URI shown in the 'target' property of the Action are summarized in Table 143.

Table 143: SetEncryptionKey action parameters

Parameter Name	Type	Attributes	Notes
EncryptionKey	string	*optional*	This defines the property name for the action.

9.6.36.5 Property details

9.6.36.5.1 idRef

• st ri *@*g (od UR I) at	• The value of this property shall be the unique identifier for re the resource and it shall be of the form defined in the ad Redfish specification.
a. id **	- o nl y*

9.6.37 StorageServiceCollection

9.6.37.1 URIs   /redfish/v1/StorageServices   /redfish/v1/Systems/*{ComputerSys-temId}*/StorageServices

9.6.37.2 Properties   The properties defined for the StorageServiceCollection schema are summarized in Table 145.

Table 145: *StorageServiceCollection properties*

Property	Type	Att ributes	Notes
Description	string	*re ad-only (null)*	This property shall contain the description of this resource. The value shall conform with the 'Description' clause of the Redfish Specification.
Members [{	array		The value of each member entry shall reference a StorageService resource.

Property	Type	Att ributes	Notes
@odata.id	string	*rea d-only*	Link to a StorageService resource. See the Links section and the *StorageService* schema for details.
}]			
Me mbers@odata.nextLink	string (URI)	*rea d-only*	The value of this property shall be a URI to a resource, with the same @odata.type, containing the next set of partial members.
Name	string	*rea d-only*	This property shall contain the name of this resource or array member. The value shall conform with the 'Name' clause of the Redfish Specification.
Oem {}	object		This property shall contain the OEM extensions. All values for properties contained in this object shall conform to the Redfish S pecification-described requirements. For property details, see Oem.

9.6.38 StorageServiceMetrics 1.0.1

9.6.38.1 Description   The StorageServiceMetrics schema shall contain the usage and health statistics for a storage service in a Redfish implementation.

9.6.38.2 URIs   /redfish/v1/StorageServices/{*StorageServiceId*}/Metrics   /redfish/ v1/Systems/{*ComputerSystemId*}/StorageServices/{*StorageServiceId*}/Metrics

9.6.38.3 Properties   The properties defined for the StorageServiceMetrics 1.0.1 schema are summarized in Table 146.

Table 146: StorageServiceMetrics 1.0.1 properties

Property	Type	Att ributes	Notes
Actions {}	object		This property shall contain the available actions for this resource.
Description	string	*re ad-only (null)*	This property shall contain the description of this resource. The value shall conform with the 'Description' clause of the Redfish Specification.
Id	string	*re ad-only re quired*	This property shall contain the identifier for this resource. The value shall conform with the 'Id' clause of the Redfish Specification.
IOStatistics {}	object		The value shall represent IO statistics for this storage service. For property details, see IOStatistics.
Name	string	*re ad-only re quired*	This property shall contain the name of this resource or array member. The value shall conform with the 'Name' clause of the Redfish Specification.

Property	Type	Att ributes	Notes
Oem {}	object		This property shall contain the OEM extensions. All values for properties that this object contains shall conform to the Redfish S pecification-described requirements. For property details, see Oem.

9.6.39 StorageSystemCollection

9.6.39.1 URIs   /redfish/v1/StorageSystems

9.6.39.2 Properties   The properties defined for the StorageSystemCollection schema are summarized in Table 147.

Table 147: StorageSystemCollection properties

Property	Type	Att ributes	Notes
Description	string	*re ad-only (null)*	This property shall contain the description of this resource. The value shall conform with the 'Description' clause of the Redfish Specification.
Members [{	array		The value of each member entry shall reference a ComputerSystem resource that shall have a HostingRoles entry with a value of 'StorageServer'.

Property	Type	Att ributes	Notes
@odata.id	string (URI)	*rea d-only*	The value of this property shall be the unique identifier for the resource and it shall be of the form defined in the Redfish specification.
}]			
Me mbers@odata.nextLink	string (URI)	*rea d-only*	The value of this property shall be a URI to a resource, with the same @odata.type, containing the next set of partial members.
Name	string	*rea d-only*	This property shall contain the name of this resource or array member. The value shall conform with the 'Name' clause of the Redfish Specification.
Oem {}	object		This property shall contain the OEM extensions. All values for properties contained in this object shall conform to the Redfish S pecification-described requirements. For property details, see Oem.

9.6.40 Volume 1.10.1

9.6.40.1 Description   This resource shall be used to represent a volume, virtual disk, logical disk, LUN, or other logical storage for a Redfish implementation.

9.6.40.2 URIs /redfish/v1/CompositionService/ResourceBlocks/{*Resource-BlockId*}/Storage/{*StorageId*}/Volumes/{*VolumeId*} /redfish/v1/CompositionService/ResourceBlocks/{*ResourceBlockId*}/Systems/{*ComputerSystemId*}/Storage/{*StorageId*}/Volumes/{*VolumeId*} /redfish/v1/ResourceBlocks/{*ResourceBlockId*}/Storage/{*StorageId*}/Volumes/{*VolumeId*} /redfish/v1/ResourceBlocks/{*ResourceBlockId*}/Systems/{*ComputerSystemId*}/Storage/{*StorageId*}/Volumes/{*VolumeId*} /redfish/v1/Storage/{*StorageId*}/ConsistencyGroups/{*ConsistencyGroupId*}/Volumes/{*VolumeId*} /redfish/v1/Storage/{*StorageId*}/FileSystems/{*FileSystemId*}/CapacitySources/{*CapacitySourceId*}/ProvidingVolumes/{*VolumeId*} /redfish/v1/Storage/{*StorageId*}/StoragePools/{*StoragePoolId*}/AllocatedVolumes/{*VolumeId*} /redfish/v1/Storage/{*StorageId*}/StoragePools/{*StoragePoolId*}/CapacitySources/{*CapacitySourceId*}/ProvidingVolumes/{*VolumeId*} /redfish/v1/Storage/{*StorageId*}/Volumes/{*VolumeId*} /redfish/v1/StorageServices/{*StorageServiceId*}/ConsistencyGroups/{*ConsistencyGroupId*}/Volumes/{*VolumeId*} /redfish/v1/StorageServices/{*StorageServiceId*}/FileSystems/{*FileSystemId*}/CapacitySources/{*CapacitySourceId*}/ProvidingVolumes/{*VolumeId*} /redfish/v1/StorageServices/{*StorageServiceId*}/StoragePools/{*StoragePoolId*}/AllocatedVolumes/{*VolumeId*} /redfish/v1/StorageServices/{*StorageServiceId*}/StoragePools/{*StoragePoolId*}/CapacitySources/{*CapacitySourceId*}/ProvidingVolumes/{*VolumeId*} /redfish/v1/StorageServices/{*StorageServiceId*}/Volumes/{*VolumeId*} /redfish/v1/StorageServices/{*StorageServiceId*}/Volumes/{*VolumeId*}/CapacitySources/{*CapacitySourceId*}/ProvidingVolumes/{*ProvidingVolumeId*} /redfish/v1/Systems/{*ComputerSystemId*}/Storage/{*StorageId*}/ConsistencyGroups/{*ConsistencyGroupId*}/Volumes/{*VolumeId*} /redfish/v1/Systems/{*ComputerSystemId*}/Storage/{*StorageId*}/FileSystems/{*FileSystemId*}/CapacitySources/{*CapacitySourceId*}/ProvidingVolumes/{*VolumeId*} /redfish/v1/Systems/{*ComputerSystemId*}/Storage/{*StorageId*}/StoragePools/{*StoragePoolId*}/AllocatedVolumes/{*VolumeId*} /redfish/v1/Systems/{*ComputerSystemId*}/Storage/{*StorageId*}/StoragePools/{*StoragePoolId*}/CapacitySources/{*CapacitySourceId*}/ProvidingVolumes/{*VolumeId*} /redfish/v1/Systems/{*ComputerSystemId*}/Storage/{*StorageId*}/Volumes/{*VolumeId*}

9.6.40.3 Properties The properties defined for the Volume 1.10.1 schema are summarized in Table 148.

Table 148: Volume 1.10.1 properties

Property	Type	Att ributes	Notes
AccessCapabilities (v1.1+) [ ]	array (string (enum))	*re ad-writ e {=html} (null)*	Each entry shall specify a current storage access capability. *For the possible property values, see AccessCapabilities in Property details.*
Actions {	object		The Actions property shall contain the available actions for this resource.
#Volume .AssignReplicaTarget (v1.4+) {}	object		This action shall be used to establish a replication relationship by assigning an existing volume to serve as a target replica for an existing source volume. *For more information, see the Actions section below.*

Property	Type	Att ributes	Notes
#Vol ume.ChangeRAIDLayout *(v1.5+)* {}	object		This action shall request the system to change the RAID layout of the volume. Depending on the combination of the submitted parameters, this could be changing the RAID type, changing the span count, changing the number of drives used by the volume, or another configuration change supported by the system. Note that usage of this action while online may potentially cause data loss if the available capacity is reduced. *For more information, see the Actions section below.*
#Vol ume.CheckConsistency {}	object		This defines the name of the custom action supported on this resource. *For more information, see the Actions section below.*
#Volume .CreateReplicaTarget *(v1.4+)* {}	object		This action shall be used to create a new volume resource to provide expanded data protection through a replica relationship with the specified source volume. *For more information, see the Actions section below.*

Property	Type	Att ributes	Notes
• *#Volume.ForceEnable** *(v1.5+)* {}	object		This action shall request the system to force the volume to enabled state regardless of data loss scenarios. *For more information, see the Actions section below.*
#Volume.Initialize *(v1.5+)* {}	object		This defines the name of the custom action supported on this resource. If InitializeMethod is not specified in the request body, but the property InitializeMethod is specified, the property InitializeMethod value should be used. If neither is specified, the InitializeMethod should be Foreground. *For more information, see the Actions section below.*
#Volume.Remov eReplicaRelationship *(v1.4+)* {}	object		This action shall be used to disable data synchronization between a source and target volume, remove the replication relationship, and optionally delete the target volume. *For more information, see the Actions section below.*

Property	Type	Att ributes	Notes
#Volu me.ResumeReplication *(v1.4+)* {}	object		This action shall be used to resume the active data synchronization between a source and target volume, without otherwise altering the replication relationship. *For more information, see the Actions section below.*
#Volume.ReverseRep licationRelationship *(v1.4+)* {}	object		This action shall be used to reverse the replication relationship between a source and target volume. *For more information, see the Actions section below.*
#Vol ume.SplitReplication *(v1.4+)* {}	object		This action shall be used to split the replication relationship and suspend data synchronization between a source and target volume. *For more information, see the Actions section below.*
#Volum e.SuspendReplication *(v1.4+)* {}	object		This action shall be used to suspend active data synchronization between a source and target volume, without otherwise altering the replication relationship. *For more information, see the Actions section below.*

}

Property	Type	Att ributes	Notes
AllocatedPools *(v1.1+)* {	object		The value of this property shall contain references to all storage pools allocated from this volume. Contains a link to a resource.
@odata.id	string	*rea d-only*	Link to Collection of *StoragePool*. See the StoragePool schema for details.
}			
ALUA *(v1.10+)* {	object	** (null)**	This shall identify the ALUA properties for this volume.
ANAGroupId *(v1.10+)*	number	*re ad-only (null)*	This shall contain the ANA group id for this volume.
}			
BlockSizeBytes	integer (By)	*r ead-onl y {=html} (null)*	This property shall contain size of the smallest addressable unit of the associated volume.
Capacity *(v1.1+)* {}	object		Information about the utilization of capacity allocated to this storage volume. For property details, see Capacity v1.0.0).
CapacityBytes	integer (By)	*re ad-writ e {=html} (null)*	This property shall contain the size in bytes of the associated volume.

Property	Type	Att ributes	Notes
CapacitySources *(v1.1+)* [{	array		Fully or partially consumed storage from a source resource. Each entry provides capacity allocation information from a named source resource.
@odata.id	string	*read -write*	Link to a CapacitySource resource. See the Links section and the *CapacitySource* schema for details.
}]			
Compressed *(v1.4+)*	boolean	*rea d-write (null)*	This property shall contain a boolean indicator if the Volume is currently utilizing compression or not.
Connections *(v1.9+)* [{	array		The value of this property shall contain references to all Connections that include this volume.
@odata.id	string (URI)	*rea d-only*	The value of this property shall be the unique identifier for the resource and it shall be of the form defined in the Redfish specification.
}]			
Deduplicated *(v1.4+)*	boolean	*rea d-write (null)*	This property shall contain a boolean indicator if the Volume is currently utilizing deduplication or not.

Property	Type	Att ributes	Notes
Description	string	*re ad-only (null)*	This property shall contain the description of this resource. The value shall conform with the 'Description' clause of the Redfish Specification.
DisplayName *(v1.4+)*	string	*rea d-write (null)*	This property shall contain a user-configurable string to name the volume.
Encrypted	boolean	*rea d-write (null)*	This property shall contain a boolean indicator if the Volume is currently utilizing encryption or not.
EncryptionTypes [ ]	array (string (enum))	*read -write*	This property shall contain the types of encryption used by this Volume. *For the possible property values, see EncryptionTypes in Property details.*
Id	string	*re ad-only re quired*	This property shall contain the identifier for this resource. The value shall conform with the 'Id' clause of the Redfish Specification.
Identifiers [{ }]	array (object)		This property shall contain a list of all known durable names for the associated volume. For property details, see Identifier v1.19.1).

Property	Type	Att ributes	Notes
InitializeMethod *(v1.6+)*	string (enum)	*r ead-onl y\<br\> {=html} (null)*	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. *For the possible property values, see InitializeMethod in Property details.*
IOPerfModeEnabled *(v1.5+)*	boolean	*rea d-write (null)*	This property shall indicate whether IO performance mode is enabled for the volume.
IOStatistics *(v1.2+, deprecated v1.10)* {}	object		The value shall represent IO statistics for this volume. For property details, see IOStatistics. *Deprecated in v1.10 and later. This property is deprecated in favor of the IOStatistics property in VolumeMetrics.*

Property	Type	Att ributes	Notes
IsBootCapable *(v1.7+)*	boolean	*rea d-write (null)*	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.
Links {	object		The Links property, as described by the Redfish Specification, shall contain references to resources that are related to, but not contained by (subordinate to), this resource.
CacheDataVolumes *(v1.6+)* [{	array		This shall be a pointer to the cache data volumes this volume serves as a cache volume. The corresponding VolumeUsage property shall be set to CacheOnly when this property is used.
@odata.id	string	*rea d-only*	Link to another Volume resource.
}]			

Property	Type	Att ributes	Notes
CacheVolumeSource *(v1.6+)* {	object	* (null)*	This shall be a pointer to the cache volume source for this volume. The corresponding VolumeUsage property shall be set to Data when this property is used.
@odata.id	string	*rea d-only*	Link to another Volume resource.
}			
ClassOfService *(v1.1+)* {	object		This property shall contain a reference to the ClassOfService that this storage volume conforms to. See the *ClassOfService* schema for details on this property.
@odata.id	string	*rea d-only*	Link to a ClassOfService resource. See the Links section and the *ClassOfService* schema for details.
}			
ClientEndpoints *(v1.4+)* [{	array		The value of this property shall be references to the client Endpoints this volume is associated with.
@odata.id	string (URI)	*rea d-only*	The value of this property shall be the unique identifier for the resource and it shall be of the form defined in the Redfish specification.

Property	Type	Att ributes	Notes
}]			
ConsistencyGroups *(v1.4+)* [{	array		The value of this property shall be references to the ConsistencyGroups this volume is associated with.
@odata.id	string	*rea d-only*	Link to a ConsistencyGroup resource. See the Links section and the *ConsistencyGroup* schema for details.
}]			
Controllers *(v1.9+)* [{	array		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.
@odata.id	string (URI)	*rea d-only*	The value of this property shall be the unique identifier for the resource and it shall be of the form defined in the Redfish specification.
}]			

Property	Type	Att ributes	Notes
DedicatedSpareDrives *(v1.2+)* [{	array		The value of this property shall be a reference to the resources that this volume is associated with and shall reference resources of type Drive. This property shall only contain references to Drive entities which are currently assigned as a dedicated spare and are able to support this Volume.
@odata.id	string (URI)	*rea d-only*	The value of this property shall be the unique identifier for the resource and it shall be of the form defined in the Redfish specification.
}]			
Drives [{	array		The value of this property shall be a reference to the resources that this volume is associated with and shall reference resources of type Drive. This property shall only contain references to Drive entities which are currently members of the Volume, not hot spare Drives which are not currently a member of the volume.

Property	Type	Att ributes	Notes
@odata.id	string (URI)	*rea d-only*	The value of this property shall be the unique identifier for the resource and it shall be of the form defined in the Redfish specification.
}]			
JournalingMedia *(v1.5+)* {	object	* (null)*	This shall be a pointer to the journaling media used for this Volume to address the write hole issue. Valid when Wri teHoleProtectionPolicy property is set to 'Journaling'.
@odata.id	string (URI)	*rea d-only*	The value of this property shall be the unique identifier for the resource and it shall be of the form defined in the Redfish specification.
}			
Oem {}	object		This property shall contain the OEM extensions. All values for properties contained in this object shall conform to the Redfish S pecification-described requirements. For property details, see Oem.

Property	Type	Att ributes	Notes
O wningStorageResource (v1.5+) {	object		This shall be a pointer to the Storage resource that owns or contains this volume.
@odata.id	string (URI)	*rea d-only*	The value of this property shall be the unique identifier for the resource and it shall be of the form defined in the Redfish specification.
}			
OwningStorageService (v1.4+) {	object		This shall be a pointer to the StorageService that owns or contains this volume. See the *StorageService* schema for details on this property.
@odata.id	string	*rea d-only*	Link to a StorageService resource. See the Links section and the *StorageService* schema for details.
}			
ProvidingStoragePool (v1.10+) {	object	* (null)*	This property shall contain a pointer to the StoragePool resource that provides this volume resource. See the *StoragePool* schema for details on this property.

Property	Type	Att ributes	Notes
@odata.id	string	*rea d-only*	Link to a StoragePool resource. See the Links section and the *StoragePool* schema for details.
}			
ServerEndpoints *(v1.4+)* [{	array		The value of this property shall be references to the server Endpoints this volume is associated with.
@odata.id	string (URI)	*rea d-only*	The value of this property shall be the unique identifier for the resource and it shall be of the form defined in the Redfish specification.
}]			
SpareResourceSets *(v1.3+)* [{	array		Each referenced SpareResourceSet shall contain resources that may be utilized to replace the capacity provided by a failed resource having a compatible type.
@odata.id	string	*read -write*	Link to a SpareResourceSet resource. See the Links section and the *SpareResourceSet* schema for details.
}]			

Property	Type	Att ributes	Notes
StorageGroups *(v1.4+)* [{	array		The value of this property shall be references to the StorageGroups this volume is associated with.
@odata.id	string	*rea d-only*	Link to a StorageGroup resource. See the Links section and the *StorageGroup* schema for details.
}]			
}			
LogicalUnitNumber *(v1.4+)*	integer	*re ad-only (null)*	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.
LowSpaceWarn ingThresholdPercents *(v1.1+)* [ ]	array (%) (integer, null)	*read -write*	Each time the following value is less than one of the values in the array the LOW_S PACE_THRESHOLD_WARNING event shall be triggered: Across all CapacitySources entries, percent = (SUM(AllocatedBytes) - SUM(ConsumedBytes))/SUM(AllocatedBytes).

Property	Type	Att ributes	Notes
Manufacturer *(v1.1+)*	string	*re ad-only (null)*	This property shall contain a value that represents the manufacturer or implementer of the storage volume.
MaxBlockSizeBytes *(v1.1+)*	integer (By)	*r ead-onl y {=html} (null)*	This property shall contain size of the largest addressable unit of this storage volume.
MediaSpanCount *(v1.4+)*	integer	*re ad-only (null)*	This property shall indicate the number of media elements used per span in the secondary RAID for a hierarchical RAID type.
Metrics *(v1.9+)* {	object		This property shall contain a link to a resource of type VolumeMetrics that specifies the metrics for this volume. IO metrics are reported in the IOStatistics property. See the *VolumeMetrics* schema for details on this property.
@odata.id	string	*rea d-only*	Link to a VolumeMetrics resource. See the Links section and the *VolumeMetrics* schema for details.
}			

Property	Type	Att ributes	Notes
Model (v1.1+)	string	*re ad-only (null)*	The value is assigned by the manufacturer and shall represents a specific storage volume implementation.
Name	string	*re ad-only re quired*	This property shall contain the name of this resource or array member. The value shall conform with the 'Name' clause of the Redfish Specification.
NVM eNamespaceProperties (v1.5+) {	object	*(null)**	This property shall contain properties to use when Volume is used to describe an NVMe Namespace.
FormattedLBASize (v1.5+)	string	*re ad-only (null)*	This property shall contain the LBA data size and metadata size combination that the namespace has been formatted with. This is a 4-bit data structure.
IsShareable (v1.5+)	boolean	*rea d-write (null)*	This property shall indicate whether the namespace is shareable.

Property	Type	Att ributes	Notes
LBAFormat *(v1.9+)* {	object	* (null)*	This property shall describe the current LBA format ID and corresponding detailed properties, such as the LBA data size and metadata size. Use the LBAFormats property to describe namespace capabilities in a collection capabilities annotation.
LBADataSizeBytes *(v1.9+)*	integer	*re ad-only (null)*	This shall be the LBA data size reported in bytes.
LBAFormatType *(v1.9+)*	string (enum)	*r ead-onl y {=html} (null)*	This shall be the LBA format type. This property is intended for capabilities instrumentation. *For the possible property values, see LBAFormatType in Property details.*
LBAMetadataSizeBytes *(v1.9+)*	integer	*re ad-only (null)*	This shall be the LBA metadata size reported in bytes.

Property	Type	Att ributes	Notes
• *RelativePerformance*(enum) *(v1.9+)*	string	*r ead-onl y {=html} (null)*	This shall be the LBA Relative Performance type. This field indicates the relative performance of the LBA format indicated relative to other LBA formats supported by the controller. This property is intended for capabilities instrumentation. *For the possible property values, see RelativePerformance in Property details.*
}			
LBAFormats *(v1.9+)* [{	array		This property shall describe the LBA format IDs and corresponding detailed properties, such as the LBA data size and metadata size. This property is intended for use in a collection capabilities annotation. Use the LBAFormat property on an instance of a namespace.
LBADataSizeBytes *(v1.9+)*	integer	*re ad-only (null)*	This shall be the LBA data size reported in bytes.

Property	Type	Att ributes	Notes
LBAFormatType *(v1.9+)*	string (enum)	*r ead-onl y {=html} (null)*	This shall be the LBA format type. This property is intended for capabilities instrumentation. *For the possible property values, see LBAFormatType in Property details.*
LBAMetadataSizeBytes *(v1.9+)*	integer	*re ad-only (null)*	This shall be the LBA metadata size reported in bytes.
• *RelativePerformance* (v1.9+)*	string (enum)	*r ead-onl y {=html} (null)*	This shall be the LBA Relative Performance type. This field indicates the relative performance of the LBA format indicated relative to other LBA formats supported by the controller. This property is intended for capabilities instrumentation. *For the possible property values, see RelativePerformance in Property details.*

}]

Property	Type	Att ributes	Notes
• *LBAFormatsSupported (v1.8+)* [ ]	array (string (enum))	*r ead-onl y {=html} (null)*	This shall be a list of the LBA formats supported for the namespace, or potential namespaces. *For the possible property values, see LBAFormatsSupported in Property details.*

Property	Type	Att ributes	Notes
MetadataTrans ferredAtEndOfDataLBA *(v1.5+)*	boolean	*re ad-only (null)*	This property shall indicate whether or not the metadata is transferred at the end of the LBA creating an extended data LBA.
NamespaceFeatures *(v1.5+)* {	object	* (null)*	This property shall contain a set of Namespace Features.
SupportsA tomicTransactionSize *(v1.5+)*	boolean	*re ad-only (null)*	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 optimal write size (NOWS) are defined for this namespace and should be used by the host for I/O optimization.
SupportsDeallocat edOrUnwrittenLBError *(v1.5+)*	boolean	*re ad-only (null)*	This property shall indicate that the controller supports deallocated or unwritten logical block error for this namespace.

Property	Type	Att ributes	Notes
Suppor tsIOPerformanceHints *(v1.5+)*	boolean	*re ad-only (null)*	This property shall indicate that the Namespace 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.
SupportsNGUIDReuse *(v1.5+)*	boolean	*re ad-only (null)*	This property shall indicate that the namespace supports the use of an NGUID (namespace globally unique identifier) value.
Supp ortsThinProvisioning *(v1.5+)*	boolean	*re ad-only (null)*	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.

}

Property	Type	Att ributes	Notes
NamespaceId *(v1.5+)*	string	*re ad-only (null)*	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. Pattern: `^0[xX ](([a-fA-F]|[0-9])*)$`
NamespaceType *(v1.9+)*	string (enum)	*r ead-onl y {=html} (null)*	This shall identify the type of namespace. *For the possible property values, see NamespaceType in Property details.*
NumberLBAFormats *(v1.5+)*	integer (By)	*r ead-onl y {=html} (null)*	This property shall contain the number of LBA data size and metadata size combinations supported by this namespace. The value of this property is between 0 and 16. LBA formats with an index set beyond this value will not be supported.

Property	Type	Att ributes	Notes
NVMeVersion *(v1.5+)*	string	*re ad-only (null)*	This property shall contain the version of the NVMe Base Specification supported.
Suppor tsIOPerformanceHints *(v1.10+)*	boolean	*re ad-only (null)*	This property shall indicate whether the namespace supports IO performance hints.
SupportsMultiple NamespaceAttachments *(v1.10+)*	boolean	*re ad-only (null)*	This property shall indicate whether the namespace may be attached to two or more controllers.
Type *(v1.8+)*	string (enum)	*r ead-onl y {=html} (null)*	This shall identify the type of namespace. *For the possible property values, see Type in Property details.*
}			
Oem {}	object		This property shall contain the OEM extensions. All values for properties that this object contains shall conform to the Redfish S pecification-described requirements. For property details, see Oem.
Operations [{	array		This property shall contain a list of all currently running on the Volume.

Property	Type	Att ributes	Notes
Associ atedFeaturesRegistry {	object		This resource shall be used to represent a Feature registry for a Redfish implementation. See the *FeaturesRegistry* schema for details on this property.
@odata.id	string	*rea d-only*	Link to a FeaturesRegistry resource. See the Links section and the *FeaturesRegistry* schema for details.
}			
Operation *(v1.9+)*	string (enum)	*r ead-onl y {=html} (null)*	This property shall contain the type of the operation. *For the possible property values, see Operation in Property details.*
OperationName *(deprecated v1.9)*	string	*re ad-only (null)*	The name of the operation. *Deprecated in v1.9 and later. This property is deprecated in favor of the Operation property using the Operation enum.*
PercentageComplete	integer	*re ad-only (null)*	The percentage of the operation that has been completed.
}]			

Property	Type	Att ributes	Notes
OptimumIOSizeBytes	integer (By)	*r ead-onl y {=html} (null)*	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.
ProvisioningPolicy *(v1.4+)*	string (enum)	*re ad-writ e {=html} (null)*	This property shall specify the volume's supported storage allocation policy. *For the possible property values, see ProvisioningPolicy in Property details.*
RAIDType *(v1.3.1+)*	string (enum)	*r ead-onl y {=html} (null)*	This property shall contain the RAID type of the associated Volume. *For the possible property values, see RAIDType in Property details.*
ReadCachePolicy *(v1.4+)*	string (enum)	*re ad-writ e {=html} (null)*	This property shall contain a boolean indicator of the read cache policy for the Volume. *For the possible property values, see ReadCachePolicy in Property details.*

Property	Type	Att ributes	Notes
Recoverabl eCapacitySourceCount *(v1.3+)*	integer	*rea d-write (null)*	The value is the number of available capacity source resources currently available in the event that an equivalent capacity source resource fails.
Rema iningCapacityPercent *(v1.2+)*	integer	*re ad-only (null)*	If present, this value shall return {[(SUM(AllocatedBytes) - SU M(ConsumedBytes)]/SUM (AllocatedBytes)}*100 represented as an integer value.
RemoteReplicaTargets *(v1.8+)* [ ]	array (string, null)	*rea d-only*	The value shall reference the URIs to the remote target replicas that are sourced by this replica. Remote indicates that the replica is managed by a separate Swordfish service instance.
ReplicaInfo *(v1.1+)* {}	object		This property shall describe the replica relationship between this storage volume and a corresponding source volume. For property details, see ReplicaInfo v1.4.0).
ReplicaTargets *(v1.3+)* [{	array		The value shall reference the target replicas that are sourced by this replica.

Property	Type	Att ributes	Notes
@odata.id	string (URI)	*rea d-only*	The value of this property shall be the unique identifier for the resource and it shall be of the form defined in the Redfish specification.
}]			
ReplicationEnabled *(v1.9+)*	boolean	*rea d-write (null)*	The property shall indicate whether or not replication is enabled on the volume. This property shall be consistent with the state reflected at the storage pool level.
Status {}	object		The property shall contain the status of the Volume. For property details, see Status.
StorageGroups *(v1.1+, deprecated v1.9)* {	object		The value of this property shall contain references to all storage groups that include this volume. Contains a link to a resource. *Deprecated in v1.9 and later. This property is deprecated in favor of the Connections property.*
@odata.id	string	*rea d-only*	Link to Collection of *StorageGroup*. See the StorageGroup schema for details.
}			

Property	Type	Att ributes	Notes
StripSizeBytes *(v1.4+)*	integer (By)	*re ad-writ e {=html} (null)*	The number of consecutively addressed virtual disk blocks (bytes) mapped to consecutively addressed blocks on a single member extent of a disk array. Synonym for stripe depth and chunk size.
VolumeType	string (enum)	*r ead-onl y {=html} (null)*	This property shall contain the type of the associated Volume. *For the possible property values, see VolumeType in Property details.*
VolumeUsage *(v1.4+)*	string (enum)	*r ead-onl y {=html} (null)*	This property shall contain the volume usage type for the Volume. *For the possible property values, see VolumeUsage in Property details.*
WriteCachePolicy *(v1.4+)*	string (enum)	*re ad-writ e {=html} (null)*	This property shall contain a boolean indicator of the write cache policy for the Volume. *For the possible property values, see WriteCachePolicy in Property details.*

Property	Type	Att ributes	Notes
WriteCacheState *(v1.4+)*	string (enum)	*r ead-onl y {=html} (null)*	This property shall contain the WriteCacheState policy setting for the Volume. *For the possible property values, see WriteCacheState in Property details.*
Write HoleProtectionPolicy *(v1.4+)*	string (enum)	*read -write*	This property specifies the policy that is enabled to address the write hole issue on the RAID volume. If no policy is enabled at the moment, this property shall be set to 'Off'. *For the possible property values, see Wri teHoleProtectionPolicy in Property details.*

9.6.40.4 Actions

9.6.40.4.1 AssignReplicaTarget *(v1.4+)*   Description

This action shall be used to establish a replication relationship by assigning an exist-ing volume to serve as a target replica for an existing source volume.

Action URI

{Base URI of target resource}/Actions/Volume.AssignReplicaTarget

Action parameters

The parameters for the action which are included in the POST body to the URI shown in the 'target' property of the Action are summarized in Table 149.

Table 149: AssignReplicaTarget action parameters

Parameter Name	Type	Attributes	Notes
ReplicaType	string (enum)	*required*	This parameter shall contain the type of replica relationship to be created (e.g., Clone, Mirror, Snap). *For the possible property values, see ReplicaType in Property details.*
R eplicaUpdateMode	string (enum)	*required*	This parameter shall specify the replica update mode. *For the possible property values, see Rep licaUpdateMode in Property details.*
TargetVolume	string	*required*	This parameter shall contain the Uri to the existing target volume.

9.6.40.4.2 ChangeRAIDLayout *(v1.5+)*   Description

This action shall request the system to change the RAID layout of the volume. Depending on the combination of the submitted parameters, this could be changing the RAID type, changing the span count, changing the number of drives used by the volume, or another configuration change supported by the system. Note that usage of this action while online may potentially cause data loss if the available capacity is reduced.

Action URI

{Base URI of target resource}/Actions/Volume.ChangeRAIDLayout

Action parameters

The parameters for the action which are included in the POST body to the URI shown in the 'target' property of the Action are summarized in Table 150.

Table 150: ChangeRAIDLayout action parameters

Parameter Name	Type	Attributes	Notes
Drives [{	array	*optional*	This parameter shall contain an array of the drives to be used by the volume.
@odata.id	string (URI)	*read-only*	The value of this property shall be the unique identifier for the resource and it shall be of the form defined in the Redfish specification.
}]			
MediaSpanCount	integer	*optional*	This parameter shall contain the requested number of media elements used per span in the secondary RAID for a hierarchical RAID type.

Parameter Name	Type	Attributes	Notes
RAIDType	string (enum)	*optional*	This parameter shall contain the requested RAID type for the volume. *For the possible property values, see RAIDType in Property details.*
StripSizeBytes	integer	*optional*	This parameter shall contain the number of blocks (bytes) requested for the strip size.

9.6.40.4.3 CheckConsistency   Description

This defines the name of the custom action supported on this resource.

Action URI

{Base URI of target resource}/Actions/Volume.CheckConsistency

Action parameters

This action takes no parameters.

9.6.40.4.4 CreateReplicaTarget *(v1.4+)*   Description

This action shall be used to create a new volume resource to provide expanded data protection through a replica relationship with the specified source volume.

Action URI

{Base URI of target resource}/Actions/Volume.CreateReplicaTarget

Action parameters

The parameters for the action which are included in the POST body to the URI shown in the 'target' property of the Action are summarized in Table 151.

Table 151: CreateReplicaTarget action parameters

Parameter Name	Type	Attributes	Notes
ReplicaType	string (enum)	*required*	This parameter shall contain the type of replica relationship to be created (e.g., Clone, Mirror, Snap). *For the possible property values, see ReplicaType in Property details.*
R eplicaUpdateMode	string (enum)	*required*	This parameter shall specify the replica update mode. *For the possible property values, see Rep licaUpdateMode in Property details.*
T argetStoragePool	string	*required*	This parameter shall contain the Uri to the existing StoragePool in which to create the target volume.
VolumeName	string	*optional*	This parameter shall contain the Name for the target volume.

9.6.40.4.5 ForceEnable *(v1.5+)*   Description

This action shall request the system to force the volume to enabled state regardless of data loss scenarios.

Action URI

{Base URI of target resource}/Actions/Volume.ForceEnable

Action parameters

This action takes no parameters.

9.6.40.4.6 Initialize *(v1.5+)*   Description

This defines the name of the custom action supported on this resource. If InitializeMethod is not specified in the request body, but the property InitializeMethod is specified, the property InitializeMethod value should be used. If neither is specified, the InitializeMethod should be Foreground.

Action URI

{Base URI of target resource}/Actions/Volume.Initialize

Action parameters

The parameters for the action which are included in the POST body to the URI shown in the 'target' property of the Action are summarized in Table 152.

Table 152: Initialize action parameters

Parameter Name	Type	Attributes	Notes
InitializeMethod	string (enum)	*optional*	This defines the property name for the action. *For the possible property values, see InitializeMethod in Property details.*

Parameter Name	Type	Attributes	Notes
InitializeType	string (enum)	*optional*	This defines the property name for the action. *For the possible property values, see InitializeType in Property details.*

9.6.40.4.7 RemoveReplicaRelationship *(v1.4+)*   Description

This action shall be used to disable data synchronization between a source and target volume, remove the replication relationship, and optionally delete the target volume.

Action URI

{Base URI of target resource}/Actions/Volume.RemoveReplicaRelationship

Action parameters

The parameters for the action which are included in the POST body to the URI shown in the 'target' property of the Action are summarized in Table 153.

Table 153: RemoveReplicaRelationship action parameters

Parameter Name	Type	Attributes	Notes
De leteTargetVolume	boolean	*optional*	This parameter shall indicate whether or not to delete the target volume as part of the operation. If not defined, the system should use its default behavior.

Parameter Name	Type	Attributes	Notes
TargetVolume	string	*required*	This parameter shall contain the Uri to the existing target volume.

9.6.40.4.8 ResumeReplication *(v1.4+)*   Description

This action shall be used to resume the active data synchronization between a source and target volume, without otherwise altering the replication relationship.

Action URI

{Base URI of target resource}/Actions/Volume.ResumeReplication

Action parameters

The parameters for the action which are included in the POST body to the URI shown in the 'target' property of the Action are summarized in Table 154.

Table 154: ResumeReplication action parameters

Parameter Name	Type	Attributes	Notes
TargetVolume	string	*required*	This parameter shall contain the Uri to the existing target volume.

9.6.40.4.9 ReverseReplicationRelationship *(v1.4+)*   Description

This action shall be used to reverse the replication relationship between a source and target volume.

Action URI

{Base URI of target resource}/Actions/Volume.ReverseReplicationRelationship

Action parameters

The parameters for the action which are included in the POST body to the URI shown in the 'target' property of the Action are summarized in Table 155.

Table 155: ReverseReplicationRelationship action parameters

Parameter Name	Type	Attributes	Notes
TargetVolume	string	*required*	This parameter shall contain the Uri to the existing target volume.

9.6.40.4.10 SplitReplication *(v1.4+)* Description

This action shall be used to split the replication relationship and suspend data synchronization between a source and target volume.

Action URI

{Base URI of target resource}/Actions/Volume.SplitReplication

Action parameters

The parameters for the action which are included in the POST body to the URI shown in the 'target' property of the Action are summarized in Table 156.

Table 156: SplitReplication action parameters

Parameter Name	Type	Attributes	Notes
TargetVolume	string	*required*	This parameter shall contain the Uri to the existing target volume.

9.6.40.4.11 SuspendReplication *(v1.4+)* Description

This action shall be used to suspend active data synchronization between a source and target volume, without otherwise altering the replication relationship.

Action URI

{Base URI of target resource}/Actions/Volume.SuspendReplication

Action parameters

The parameters for the action which are included in the POST body to the URI shown in the 'target' property of the Action are summarized in Table 157.

Table 157: SuspendReplication action parameters

Parameter Name	Type	Attributes	Notes
TargetVolume	string	*required*	This parameter shall contain the Uri to the existing target volume.

9.6.40.5 Property details

9.6.40.5.1 AccessCapabilities   The defined property values are listed in Table 158. Each entry shall specify a current storage access capability.

Table 158: AccessCapabilities property values

string	Description
Append	This enumeration literal shall indicate that the storage may be written only to append.
Execute	This value shall indicate that Execute access is allowed by the file share.
Read	This enumeration literal shall indicate that the storage may be read.
Streaming	This enumeration literal shall indicate that the storage may be read sequentially.
Write	This enumeration literal shall indicate that the storage may be written multiple times.
WriteOnce	This enumeration literal shall indicate that the storage may be written only once.

9.6.40.5.2 EncryptionTypes   The defined property values are listed in Table 159. This property shall contain the types of encryption used by this Volume.

Table 159: EncryptionTypes property values

string	Description
ControllerAssisted	The volume is being encrypted by the storage controller entity.
NativeDriveEncryption	The volume is utilizing the native drive encryption capabilities of the drive hardware.
SoftwareAssisted	The volume is being encrypted by software running on the system or the operating system.

9.6.40.5.3 InitializeMethod   The defined property values are listed in Table 160. This defines the property name for the action.

Table 160: InitializeMethod property values

string	Description
Background	The volume will be available for use immediately, with data erasure and preparation to happen as background tasks.
Foreground	Data erasure and preparation tasks will complete before the volume is presented as available for use.
Skip	The volume will be available for use immediately, with no preparation.

9.6.40.5.4 InitializeType   The defined property values are listed in Table 161. This defines the property name for the action.

Table 161: InitializeType property values

string	Description
Fast	The volume is prepared for use quickly, typically by erasing just the beginning and end of the space so that partitioning can be performed.
Slow	The volume is prepared for use slowly, typically by completely erasing the volume.

9.6.40.5.5 LBAFormatsSupported The defined property values are listed in Table 162. This shall be a list of the LBA formats supported for the namespace, or potential namespaces.

Table 162: LBAFormatsSupported property values

string	Description
LBAFormat0	LBAFormat0 is a required type. Indicates the LBA data size supported.
LBAFormat1	Indicates the LBA data size if supported.
LBAFormat10	Indicates the LBA data size supported if supported.
LBAFormat11	Indicates the LBA data size supported if supported.
LBAFormat12	Indicates the LBA data size supported if supported.
LBAFormat13	Indicates the LBA data size supported if supported.
LBAFormat14	Indicates the LBA data size supported if supported.
LBAFormat15	Indicates the LBA data size supported if supported.
LBAFormat2	Indicates the LBA data size supported if supported.

string	Description
LBAFormat3	Indicates the LBA data size supported if supported.
LBAFormat4	Indicates the LBA data size supported if supported.
LBAFormat5	Indicates the LBA data size supported if supported.
LBAFormat6	Indicates the LBA data size supported if supported.
LBAFormat7	Indicates the LBA data size supported if supported.
LBAFormat8	Indicates the LBA data size supported if supported.
LBAFormat9	Indicates the LBA data size supported if supported.

9.6.40.5.6 LBAFormatType   The defined property values are listed in Table 163. This shall be the LBA format type. This property is intended for capabilities instrumentation.

Table 163: LBAFormatType property values

string	Description
LBAFormat0	LBAFormat0 is a required type. Indicates the LBA data size supported.
LBAFormat1	Indicates the LBA data size if supported.
LBAFormat10	Indicates the LBA data size supported if supported.
LBAFormat11	Indicates the LBA data size supported if supported.
LBAFormat12	Indicates the LBA data size supported if supported.

string	Description
LBAFormat13	Indicates the LBA data size supported if supported.
LBAFormat14	Indicates the LBA data size supported if supported.
LBAFormat15	Indicates the LBA data size supported if supported.
LBAFormat2	Indicates the LBA data size supported if supported.
LBAFormat3	Indicates the LBA data size supported if supported.
LBAFormat4	Indicates the LBA data size supported if supported.
LBAFormat5	Indicates the LBA data size supported if supported.
LBAFormat6	Indicates the LBA data size supported if supported.
LBAFormat7	Indicates the LBA data size supported if supported.
LBAFormat8	Indicates the LBA data size supported if supported.
LBAFormat9	Indicates the LBA data size supported if supported.

9.6.40.5.7 NamespaceType   The defined property values are listed in Table 164. This shall identify the type of namespace.

Table 164: NamespaceType property values

string	Description
Block	The namespace is configured for use with a block storage interface.

string	Description
Computational	The namespace is configured for use with a computational storage interface.
KeyValue	The namespace is configured for use with a KeyValue interface.
ZNS	The namespace is configured for use with a zoned storage interface.

9.6.40.5.8 Operation   The defined property values are listed in Table 165. This property shall contain the type of the operation.

Table 165: Operation property values

string	Description
ChangeRAIDType	A ChangeRAIDType operation is being performed.
ChangeStripSize *(v1.10+)*	A ChangeStripSize operation is being performed.
CheckConsistency	A CheckConsistency operation is being performed.
Compress	A Compress operation is being performed.
Decrypt	A Decrypt operation is being performed.
Deduplicate	A Deduplicate operation is being performed.
Delete	A Delete operation is being performed.
Encrypt	An Encrypt operation is being performed.
Format	A Format operation is being performed.
Initialize	An Initialize operation is being performed.
Rebuild	A Rebuild operation is being performed.
Replicate	A Replicate operation is being performed.
Resize	A Resize operation is being performed.
Sanitize	A Sanitize operation is being performed.

9.6.40.5.9 ProvisioningPolicy   The defined property values are listed in Table 166. This property shall specify the volume's supported storage allocation policy.

Table 166: ProvisioningPolicy property values

string	Description
Fixed	This enumeration literal specifies storage shall be fully allocated.
Thin	This enumeration literal specifies storage may be over allocated.

9.6.40.5.10 RAIDType   The defined property values are listed in Table 167. This parameter shall contain the requested RAID type for the volume.

Table 167: RAIDType property values

string	Description
None *(v1.4.2+)*	A placement policy with no redundancy at the device level.
RAID0	A placement policy where consecutive logical blocks of data are uniformly distributed across a set of independent storage devices without offering any form of redundancy. This is commonly referred to as data striping. This form of RAID will encounter data loss with the failure of any storage device in the set.
RAID00	A placement policy that creates a RAID 0 stripe set over two or more RAID 0 sets. This is commonly referred to as RAID 0+0. This form of data layout is not fault tolerant; if any storage device fails there will be data loss.

string	Description
RAID01	A data placement policy that creates a mirrored device (RAID 1) over a set of striped devices (RAID 0). This is commonly referred to as RAID 0+1 or RAID 0/1. Data stored using this form of RAID is able to survive a single RAID 0 data set failure without data loss.
RAID1	A placement policy where each logical block of data is stored on more than one independent storage device. This is commonly referred to as mirroring. Data stored using this form of RAID is able to survive a single storage device failure without data loss.
RAID10	A placement policy that creates a striped device (RAID 0) over a set of mirrored devices (RAID 1). This is commonly referred to as RAID 1/0. Data stored using this form of RAID is able to survive storage device failures in each RAID 1 set without data loss.
RAID10E	A placement policy that uses a RAID 0 stripe set over two or more RAID 10 sets. This is commonly referred to as Enhanced RAID 10. Data stored using this form of RAID is able to survive a single device failure within each nested RAID 1 set without data loss.
RAID10Triple	A placement policy that uses a striped device (RAID 0) over a set of triple mirrored devices (RAID 1Triple). This form of RAID can survive up to two failures in each triple mirror set without data loss.

string	Description
RAID1E	A placement policy that uses a form of mirroring implemented over a set of independent storage devices where logical blocks are duplicated on a pair of independent storage devices so that data is uniformly distributed across the storage devices. This is commonly referred to as RAID 1 Enhanced. Data stored using this form of RAID is able to survive a single storage device failure without data loss.
RAID1Triple	A placement policy where each logical block of data is mirrored three times across a set of three independent storage devices. This is commonly referred to as three-way mirroring. This form of RAID can survive two device failures without data loss.
RAID3	A placement policy using parity-based protection where logical bytes of data are uniformly distributed across a set of independent storage devices and where the parity is stored on a dedicated independent storage device. Data stored using this form of RAID is able to survive a single storage device failure without data loss. If the storage devices use rotating media, they are assumed to be rotationally synchronized, and the data stripe size should be no larger than the exported block size.
RAID4	A placement policy using parity-based protection where logical blocks of data are uniformly distributed across a set of independent storage devices and where the parity is stored on a dedicated independent storage device. Data stored using this form of RAID is able to survive a single storage device failure without data loss.

string	Description
RAID5	A placement policy using parity-based protection for storing stripes of 'n' logical blocks of data and one logical block of parity across a set of 'n+1' independent storage devices where the parity and data blocks are interleaved across the storage devices. Data stored using this form of RAID is able to survive a single storage device failure without data loss.
RAID50	A placement policy that uses a RAID 0 stripe set over two or more RAID 5 sets of independent storage devices. Data stored using this form of RAID is able to survive a single storage device failure within each RAID 5 set without data loss.
RAID6	A placement policy using parity-based protection for storing stripes of 'n' logical blocks of data and two logical blocks of independent parity across a set of 'n+2' independent storage devices where the parity and data blocks are interleaved across the storage devices. Data stored using this form of RAID is able to survive any two independent storage device failures without data loss.
RAID60	A placement policy that uses a RAID 0 stripe set over two or more RAID 6 sets of independent storage devices. Data stored using this form of RAID is able to survive two device failures within each RAID 6 set without data loss.

string	Description
RAID6TP	A placement policy that uses parity-based protection for storing stripes of 'n' logical blocks of data and three logical blocks of independent parity across a set of 'n+3' independent storage devices where the parity and data blocks are interleaved across the storage devices. This is commonly referred to as Triple Parity RAID. Data stored using this form of RAID is able to survive any three independent storage device failures without data loss.

9.6.40.5.11 ReadCachePolicy   The defined property values are listed in Table 168. This property shall contain a boolean indicator of the read cache policy for the Volume.

Table 168: ReadCachePolicy property values

string	Description
AdaptiveReadAhead	A caching technique in which the controller dynamically determines whether to pre-fetch data anticipating future read requests, based on previous cache hit ratio.
Off	The read cache is disabled.
ReadAhead	A caching technique in which the controller pre-fetches data anticipating future read requests.

9.6.40.5.12 RelativePerformance   The defined property values are listed in Table 169. This shall be the LBA Relative Performance type. This field indicates the relative performance of the LBA format indicated relative to other LBA formats supported by the controller. This property is intended for capabilities instrumentation.

Table 169: RelativePerformance property values

string	Description
Best	Best performance.
Better	Better performance.
Degraded	Degraded performance.
Good	Good performance.

9.6.40.5.13 ReplicaType   The defined property values are listed in Table 170. This parameter shall contain the type of replica relationship to be created (e.g., Clone, Mirror, Snap).

Table 170: ReplicaType property values

string	Description
Clone	This enumeration literal shall indicate that replication shall create a point in time, full copy the source.
Mirror	This enumeration literal shall indicate that replication shall create and maintain a copy of the source.
Snapshot	This enumeration literal shall indicate that replication shall create a point in time, virtual copy of the source.
TokenizedClone	This enumeration literal shall indicate that replication shall create a token based clone.

9.6.40.5.14 ReplicaUpdateMode   The defined property values are listed in Table 171. This parameter shall specify the replica update mode.

Table 171: ReplicaUpdateMode property values

string	Description
Active	This enumeration literal shall indicate Active-Active (i.e. bidirectional) synchronous updates.
Adaptive	This enumeration literal shall indicate that an implementation may switch between synchronous and asynchronous modes.
Asynchronous	This enumeration literal shall indicate Asynchronous updates.
Synchronous	This enumeration literal shall indicate Synchronous updates.

9.6.40.5.15 Type   The defined property values are listed in Table 172. This shall identify the type of namespace.

Table 172: Type property values

string	Description
Block	The namespace is configured for use with a block storage interface.
Computational	The namespace is configured for use with a computational storage interface.
KeyValue	The namespace is configured for use with a KeyValue interface.
ZNS	The namespace is configured for use with a zoned storage interface.

9.6.40.5.16 VolumeType   The defined property values are listed in Table 173. This property shall contain the type of the associated Volume.

Table 173: VolumeType property values

string	Description
Mirrored	The volume is a mirrored device.
NonRedundant	The volume is a non-redundant storage device.
RawDevice	The volume is a raw physical device without any RAID or other virtualization applied.
SpannedMirrors	The volume is a spanned set of mirrored devices.
SpannedStripesWithParity	The volume is a spanned set of devices which uses parity to retain redundant information.
StripedWithParity	The volume is a device which uses parity to retain redundant information.

9.6.40.5.17 VolumeUsage   The defined property values are listed in Table 174. This property shall contain the volume usage type for the Volume.

Table 174: VolumeUsage property values

string	Description
CacheOnly	The volume shall be allocated for use as a non-consumable cache only volume.
Data	The volume shall be allocated for use as a consumable data volume.
ReplicationReserve	The volume shall be allocated for use as a non-consumable reserved volume for replication use.
SystemData	The volume shall be allocated for use as a consumable data volume reserved for system use.
SystemReserve	The volume shall be allocated for use as a non-consumable system reserved volume.

9.6.40.5.18 WriteCachePolicy   The defined property values are listed in Table 175. This property shall contain a boolean indicator of the write cache policy for the Volume.

Table 175: WriteCachePolicy property values

string	Description
Off *(v1.4.1+)*	Indicates that the write cache shall be disabled.
ProtectedWriteBack	A caching technique in which the completion of a write request is signaled as soon as the data is in cache, and actual writing to non-volatile media is guaranteed to occur at a later time.
UnprotectedWriteBack	A caching technique in which the completion of a write request is signaled as soon as the data is in cache; actual writing to non-volatile media is not guaranteed to occur at a later time.
WriteThrough	A caching technique in which the completion of a write request is not signaled until data is safely stored on non-volatile media.

9.6.40.5.19 WriteCacheState   The defined property values are listed in Table 176. This property shall contain the WriteCacheState policy setting for the Volume.

Table 176: WriteCacheState property values

string	Description
Degraded	Indicates an issue with the cache state in which the cache space is diminished or disabled due to a failure or an outside influence such as a discharged battery.
Protected	Indicates that the cache state type in use generally protects write requests on non-volatile media.

string	Description
Unprotected	Indicates that the cache state type in use generally does not protect write requests on non-volatile media.

9.6.40.5.20 WriteHoleProtectionPolicy   The defined property values are listed in Table 177. This property specifies the policy that is enabled to address the write hole issue on the RAID volume. If no policy is enabled at the moment, this property shall be set to 'Off'.

Table 177: WriteHoleProtectionPolicy property values

string	Description
DistributedLog	The policy that distributes additional log (e.q. checksum of the parity) among the volume's capacity sources to address write hole issue. Additional data is used to detect data corruption on the volume.
Journaling	The policy that uses separate block device for write-ahead logging to address write hole issue. All write operations on the RAID volume are first logged on dedicated journaling device that is not part of the volume.
Oem	The policy that is Oem specific. The mechanism details are unknown unless provided separately by the Oem.
Off	The support for addressing the write hole issue is disabled. The volume is not performing any additional activities to close the RAID write hole.

9.6.41 VolumeCollection

9.6.41.1 URIs   /redfish/v1/CompositionService/ResourceBlocks/{*Resource-BlockId*}/Storage/{*StorageId*}/Volumes   /redfish/v1/CompositionService/Resource-

Blocks/*{ResourceBlockId}*/Systems/*{ComputerSystemId}*/Storage/*{StorageId}*/
Volumes   /redfish/v1/ResourceBlocks/*{ResourceBlockId}*/Storage/*{StorageId}*/Vol-
umes /redfish/v1/ResourceBlocks/*{ResourceBlockId}*/Systems/*{ComputerSystemId}*/
Storage/*{StorageId}*/Volumes   /redfish/v1/Storage/*{StorageId}*/ConsistencyGroups/
{ConsistencyGroupId}/Volumes /redfish/v1/Storage/*{StorageId}*/FileSystems/*{FileSys-
temId}*/CapacitySources/*{CapacitySourceId}*/ProvidingVolumes /redfish/v1/Storage/
{StorageId}/StoragePools/*{StoragePoolId}*/AllocatedVolumes   /redfish/v1/Storage/
{StorageId}/StoragePools/*{StoragePoolId}*/CapacitySources/*{CapacitySourceId}*/
ProvidingVolumes   /redfish/v1/Storage/*{StorageId}*/Volumes   /redfish/v1/Storage-
Services/*{StorageServiceId}*/ConsistencyGroups/*{ConsistencyGroupId}*/Volumes   /
redfish/v1/StorageServices/*{StorageServiceId}*/FileSystems/*{FileSystemId}*/Capac-
itySources/*{CapacitySourceId}*/ProvidingVolumes   /redfish/v1/StorageServices/
{StorageServiceId}/StoragePools/*{StoragePoolId}*/AllocatedVolumes   /redfish/v1/
StorageServices/*{StorageServiceId}*/StoragePools/*{StoragePoolId}*/CapacitySources/
{CapacitySourceId}/ProvidingVolumes   /redfish/v1/StorageServices/*{StorageServi-
ceId}*/Volumes /redfish/v1/StorageServices/*{StorageServiceId}*/Volumes/*{VolumeId}*/
CapacitySources/*{CapacitySourceId}*/ProvidingVolumes /redfish/v1/Systems/*{Com-
puterSystemId}*/Storage/*{StorageId}*/ConsistencyGroups/*{ConsistencyGroupId}*/
Volumes   /redfish/v1/Systems/*{ComputerSystemId}*/Storage/*{StorageId}*/FileSys-
tems/*{FileSystemId}*/CapacitySources/*{CapacitySourceId}*/ProvidingVolumes   /
redfish/v1/Systems/*{ComputerSystemId}*/Storage/*{StorageId}*/StoragePools/*{Stor-
agePoolId}*/AllocatedVolumes   /redfish/v1/Systems/*{ComputerSystemId}*/Storage/
{StorageId}/StoragePools/*{StoragePoolId}*/CapacitySources/*{CapacitySourceId}*/
ProvidingVolumes   /redfish/v1/Systems/*{ComputerSystemId}*/Storage/*{StorageId}*/
Volumes

9.6.41.2 Properties   The properties defined for the VolumeCollection schema are
summarized in Table 178.

Table 178: VolumeCollection properties

Property	Type	Att ributes	Notes
Description	string	*re ad-only (null)*	This property shall contain the description of this resource. The value shall conform with the 'Description' clause of the Redfish Specification.
Members [{	array		The value of each member entry shall reference a Volume resource.
@odata.id	string	*rea d-only*	Link to a Volume resource. See the Links section and the *Volume* schema for details.
}]			
Me mbers@odata.nextLink	string (URI)	*rea d-only*	The value of this property shall be a URI to a resource, with the same @odata.type, containing the next set of partial members.
Name	string	*rea d-only*	This property shall contain the name of this resource or array member. The value shall conform with the 'Name' clause of the Redfish Specification.

Property	Type	Att ributes	Notes
Oem {}	object		This property shall contain the OEM extensions. All values for properties contained in this object shall conform to the Redfish S pecification-described requirements. For property details, see Oem.

9.6.42 VolumeMetrics 1.1.1

9.6.42.1 Description The VolumeMetrics schema shall contain the usage and health statistics for a volume in a Redfish implementation.

9.6.42.2 URIs /redfish/v1/CompositionService/ResourceBlocks/*{Resource-BlockId}*/Storage/*{StorageId}*/Volumes/*{VolumeId}*/Metrics /redfish/v1/Compo-sitionService/ResourceBlocks/*{ResourceBlockId}*/Systems/*{ComputerSystemId}*/ Storage/*{StorageId}*/Volumes/*{VolumeId}*/Metrics /redfish/v1/ResourceBlocks/ *{ResourceBlockId}*/Storage/*{StorageId}*/Volumes/*{VolumeId}*/Metrics /redfish/v1/Re-sourceBlocks/*{ResourceBlockId}*/Systems/*{ComputerSystemId}*/Storage/*{StorageId}*/ Volumes/*{VolumeId}*/Metrics /redfish/v1/Storage/*{StorageId}*/ConsistencyGroups/ *{ConsistencyGroupId}*/Volumes/*{VolumeId}*/Metrics /redfish/v1/Storage/*{StorageId}*/ FileSystems/*{FileSystemId}*/CapacitySources/*{CapacitySourceId}*/ProvidingVolumes/ *{VolumeId}*/Metrics /redfish/v1/Storage/*{StorageId}*/StoragePools/*{StoragePoolId}*/ AllocatedVolumes/*{VolumeId}*/Metrics /redfish/v1/Storage/*{StorageId}*/Storage-Pools/*{StoragePoolId}*/CapacitySources/*{CapacitySourceId}*/ProvidingVolumes/ *{VolumeId}*/Metrics /redfish/v1/Storage/*{StorageId}*/Volumes/*{VolumeId}*/Metrics / redfish/v1/StorageServices/*{StorageServiceId}*/ConsistencyGroups/*{Consistency-GroupId}*/Volumes/*{VolumeId}*/Metrics /redfish/v1/StorageServices/*{StorageSer-viceId}*/FileSystems/*{FileSystemId}*/CapacitySources/*{CapacitySourceId}*/Provid-ingVolumes/*{VolumeId}*/Metrics /redfish/v1/StorageServices/*{StorageServiceId}*/ StoragePools/*{StoragePoolId}*/AllocatedVolumes/*{VolumeId}*/Metrics /redfish/v1/ StorageServices/*{StorageServiceId}*/StoragePools/*{StoragePoolId}*/CapacitySources/

{CapacitySourceId}/ProvidingVolumes/*{VolumeId}*/Metrics   /redfish/v1/StorageSer-vices/*{StorageServiceId}*/Volumes/*{VolumeId}*/CapacitySources/*{CapacitySourceId}*/ProvidingVolumes/*{ProvidingVolumeId}*/Metrics   /redfish/v1/StorageServices/*{StorageServiceId}*/Volumes/*{VolumeId}*/Metrics   /redfish/v1/Systems/*{Computer-SystemId}*/Storage/*{StorageId}*/ConsistencyGroups/*{ConsistencyGroupId}*/Volumes/*{VolumeId}*/Metrics   /redfish/v1/Systems/*{ComputerSystemId}*/Storage/*{StorageId}*/FileSystems/*{FileSystemId}*/CapacitySources/*{CapacitySourceId}*/ProvidingVolumes/*{VolumeId}*/Metrics   /redfish/v1/Systems/*{ComputerSystemId}*/Storage/*{StorageId}*/StoragePools/*{StoragePoolId}*/AllocatedVolumes/*{VolumeId}*/Metrics   /redfish/v1/Systems/*{ComputerSystemId}*/Storage/*{StorageId}*/StoragePools/*{StoragePoolId}*/CapacitySources/*{CapacitySourceId}*/ProvidingVolumes/*{VolumeId}*/Metrics   /red-fish/v1/Systems/*{ComputerSystemId}*/Storage/*{StorageId}*/Volumes/*{VolumeId}*/Metrics

9.6.42.3 Properties   The properties defined for the VolumeMetrics 1.1.1 schema are summarized in Table 179.

Table 179: VolumeMetrics 1.1.1 properties

Property	Type	Att ributes	Notes
Actions {}	object		This property shall contain the available actions for this resource.
C onsistencyCheckCount *(v1.1+)*	number	*re ad-only (null)*	This property shall contain the number of consistency checks completed over the lifetime of the volume.
Consis tencyCheckErrorCount	number	*re ad-only (null)*	This property shall contain the number of consistency check errors over the lifetime of the volume.

Property	Type	Att ributes	Notes
Correct ableIOReadErrorCount	integer	*re ad-only (null)*	This property shall contain the number of the correctable read errors for the lifetime of the volume.
Correcta bleIOWriteErrorCount	integer	*re ad-only (null)*	This property shall contain the number of the correctable write errors for the lifetime of the volume.
Description	string	*re ad-only (null)*	This property shall contain the description of this resource. The value shall conform with the 'Description' clause of the Redfish Specification.
Id	string	*re ad-only re quired*	This property shall contain the identifier for this resource. The value shall conform with the 'Id' clause of the Redfish Specification.
IOStatistics *(v1.1+)* {}	object		The value shall represent IO statistics for this volume. For property details, see IOStatistics.
Name	string	*re ad-only re quired*	This property shall contain the name of this resource or array member. The value shall conform with the 'Name' clause of the Redfish Specification.

Property	Type	Att ributes	Notes
Oem {}	object		This property shall contain the OEM extensions. All values for properties that this object contains shall conform to the Redfish S pecification-described requirements. For property details, see Oem.
RebuildErrorCount	number	*re ad-only (null)*	This property shall contain the number of rebuild errors over the lifetime of the volume.
StateChangeCount	number	*re ad-only (null)*	This property shall contain the number of state changes (changes in Status.State) for this volume.
Uncorrect ableIOReadErrorCount	integer	*re ad-only (null)*	This property shall contain the number of the uncorrectable read errors for the lifetime of the volume.
Uncorrecta bleIOWriteErrorCount	integer	*re ad-only (null)*	This property shall contain the number of the uncorrectable write errors for the lifetime of the volume.

Annex A: Bibliography

A.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.

A.2 Informational references

The informational references are summarized in Table A.1.

Tag	Title (Version)	Author	URL
E rrors	Swordfish Scalable Storage Management Error Handling Guide	SNIA	<https://www.snia.org/forums/smi/swordfish>
Me trics	Swordfish Metrics White Paper	SNIA	<https://www.snia.org/forums/smi/swordfish>
NVMe	Swordfish NVMe Model Overview and Mapping Guide	SNIA	<https://www.snia.org/forums/smi/swordfish>
Pro files	Swordfish Profile Bundle	SNIA	<https://www.snia.org/forums/smi/swordfish>
Pro files	Swordfish Profile Bundle	SNIA	<https://www.snia.org/forums/smi/swordfish>
Prope rties	Swordfish Property Guide	SNIA	<https://www.snia.org/forums/smi/swordfish>
S chema	Swordfish Schema and Registries Bundle	SNIA	<https://www.snia.org/forums/smi/swordfish>
Temp lates	Swordfish Templates Bundle	SNIA	<https://www.snia.org/forums/smi/swordfish>

Tag	Title (Version)	Author	URL
TLS	TLS Specification for Storage Systems	SNIA	<https://www.snia.org/tech_activities/standards/curr_standards/tls>
Users Guide	Swordfish Scalable Storage Management API User's Guide	SNIA	<https://www.snia.org/forums/smi/swordfish>

Table A.1: Informational References