INTERNATIONAL STANDARD

ISO/IEC 5965

First edition
2021-08

Information technology — Swordfish Scalable Storage Management API Specification

Reference number
ISO/IEC 5965:2021(E)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of document should be noted (see www.iso.org/directives or www.iec.ch/members_experts/refdocs).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents) or the IEC list of patent declarations received (see patents.iec.ch).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see www.iso.org/iso/foreword.html. In the IEC, see www.iec.ch/understanding-standards.

This document was prepared by SNIA (as Swordfish Scalable Storage Management API Specification, Version 1.1.0c) and drafted in accordance with its editorial rules. It was adopted, under the JTC 1 PAS procedure, by Joint Technical Committee ISO/IEC JTC 1, *Information technology*.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html and www.iec.ch/national-committees.

Table of Contents

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 (DSP0266) from the DMTF.

2 Scope

Swordfish extends the Redfish Scalable Platforms Management API Specification, as defined by ISO 30115 It defines 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

Swordfish's storage model is built around well-defined classes of service, which provide a means to map high-level business goals and objectives to specific, storage-based actions and requirements, in a clear and consistent way that 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 class of service includes snapshots, replication, mapping and masking, and provisioning.

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.

The Swordfish specification defines additional data models and behaviors for the management of storage systems and storage infrastructure. A Swordfish implementation shall conform to all requirements specified in the Redfish specifications.

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.

This document defines the Swordfish Scalable Storage Management API.

2.1 Audience Assumptions

As 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 ISO 30115 and any updates or clarifications introduced by the DMTF. This document cannot be fully understood without that context.

3 Normative References

3.1 Overview

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

3.2 Approved references

Table 3: Approved normative references

Tag	Title (Version)	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

Table 3: Approved normative references, cont.

Tag	Title (Version)	Author	URL
ISO-Direct	ISO/IEC Directives, Part 2 Principles and rules for the structure and drafting of ISO and IEC documents (Seventh Edition, 2016)	ISO/IEC	http://isotc.iso.org/livelink/livelink/fetch/2000/2122/4230450/4230456/ISO_IEC_Directives *Part_2 Principles_and_rules_for_the structure_and_drafting_of_ISO_and_IEC* documents *-2016%287th_edition%29-_PDF.pdf?* nodeid=17667902&vernum=-2
OData	Open Data Protocol (v. 4.0)	OASIS	https://www.oasis-open.org/standards#odatav4.0
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 (4.0)	OASIS	http://docs.oasis-open.org/odata/odata/v4.0/odata-v4.0-part3-csdl.html
ITIL	ITIL Glossary (2011)	ITIL	https://www.axelos.com/Corporate/media/Files/Glossaries/ITIL_2011_Glossary_GB-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
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 1.0.1	SNIA	https://redfish.dmtf.org/registries/swordfish/v1/SwordfishFeatureRegistry.1.0.1.json

Table 3: Approved normative references, cont.

Tag	Title (Version)	Author	URL
Messages	Swordfish Message Registry, version 1.0.2	SNIA	https://redfish.dmtf.org/registries/swordfish/v1/Swordfish.1.0.2.json
EnergyStar	ENERGY STAR Data Center Storage Version 1.1 Updated Program Requirements – April 1, 2019	EPA	https://www.energystar.gov/sites/default/files/ENERGY STAR Data Center Storage Final Version 1.1 Specification Rev. April 2019.pdf
ISO-20648	Information technology — TLS specification for storage systemsi	ISO/IEC	https://www.iso.org/standard/68622.html
ISO-30115	ISO/IEC 30115(en) Information technology — Redfish scalable platforms management API specification	ISO/IEC	https://www.iso.org/standard/53235.html

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

Table 4 summarizes the terms are used in this document.

Table 4: Swordfish terms

Term	Definition
Entity	An instance of a schema element.
Model	A set of entities and the relationships between them that define the semantics, behavior and state of that set.
OData service	A REST-based service that allows resources, identified using Uniform Resource Locators (URLs) and defined in a model, to be published and edited by Web clients using simple HTTP messages.
Resource	A central element in a model, which represents a physical construct or a logical service, and is further defined by other model entities.
Schema	A formal language representation of a model that conforms to a metamodel.
Service Document	A particular resource that is directly accessed via an OData service entry point. This resource serves as a starting point for locating and accessing the other resources and associated metadata that together make up an instance of a Swordfish service.
Swordfish service	An extension to the Redfish Service that conforms to the Swordfish specification, and provides REST-ful 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	Defines Redfish Resources according to OData schema representation. You can directly translate a Redfish Schema to a JSON Schema representation.
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	Resource that serves as the starting point for locating and accessing the other resources and associated metadata that together 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 in 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
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 (DSP0266) from the DMTF.

5.2 Relation to Redfish

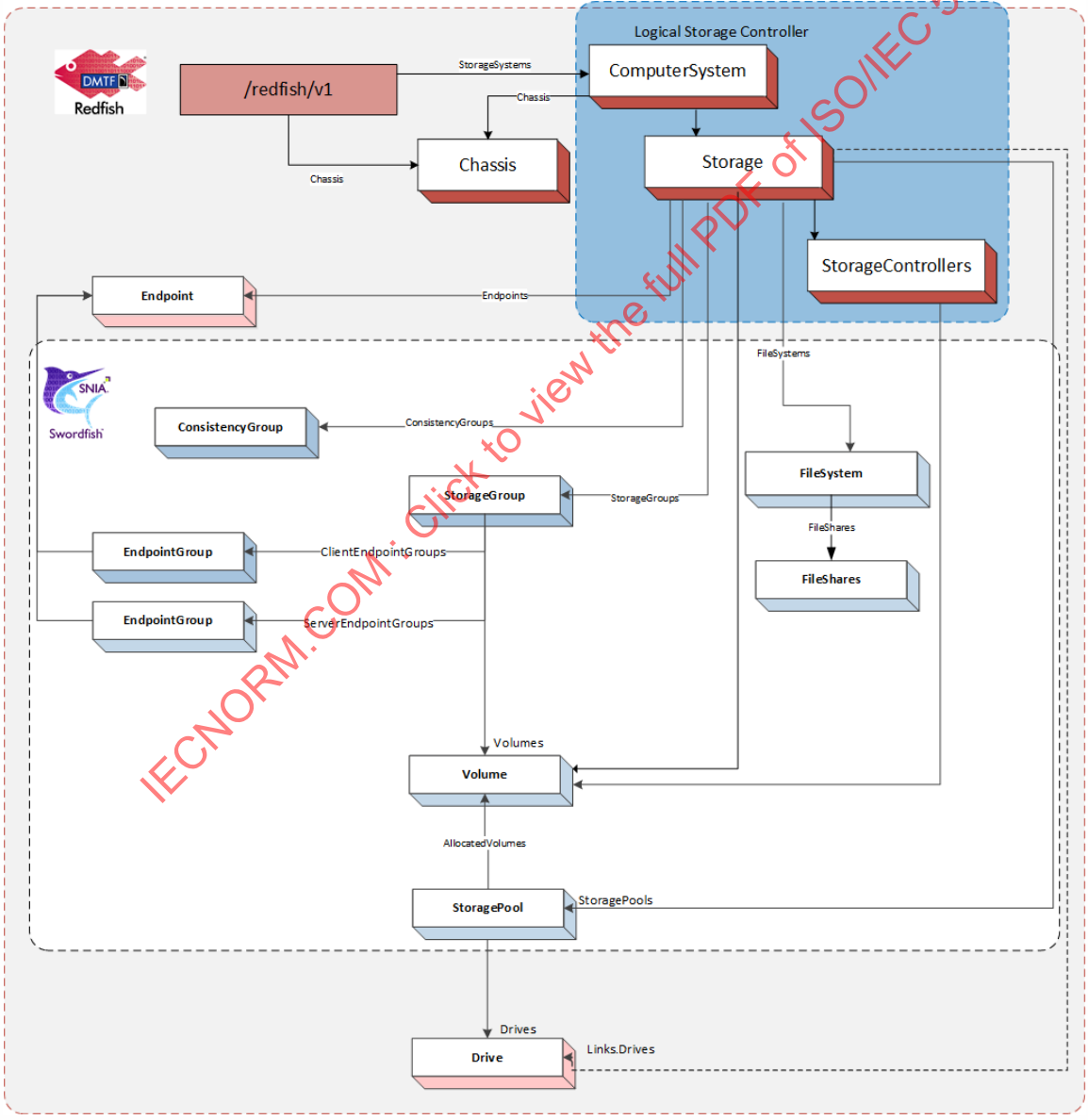

Figure 1: Extension Overview

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

Storage systems managed by the Swordfish service are located in the `ServiceRoot` via the `StorageSystems` resource collection. They are modeled using Redfish `ComputerSystems`. 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 Integrated Configuration

The SIC uses the same ComputerSystem model instantiation as the server where the physical element resides.

The logical storage controller is modeled using the Redfish `Storage` and `StorageController` 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` the `StoragePools`. These relationships are illustrated in Figure 2.

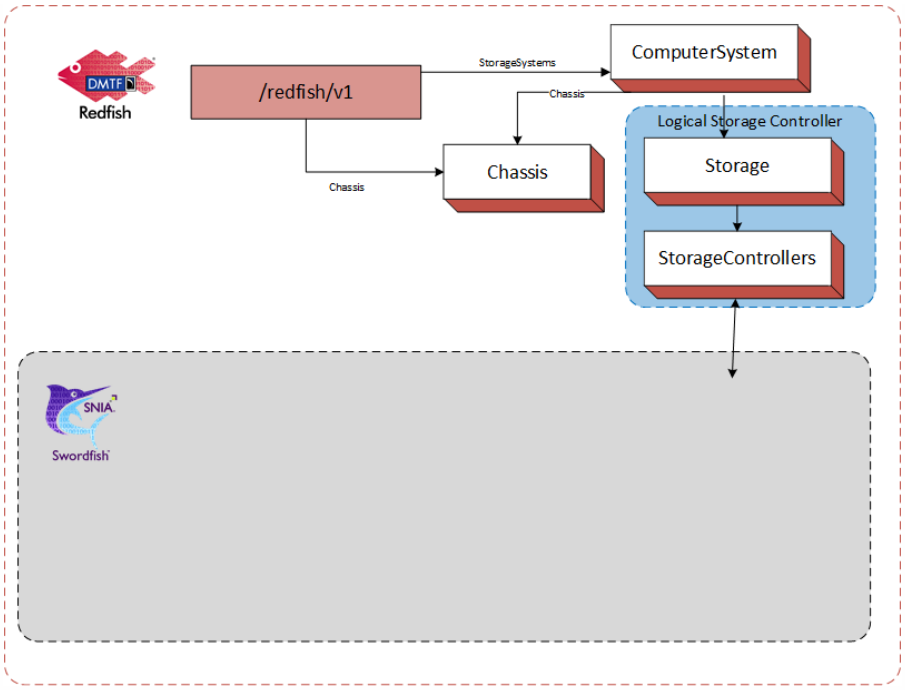

Figure 2: Integrated Configuration Overview

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 3.

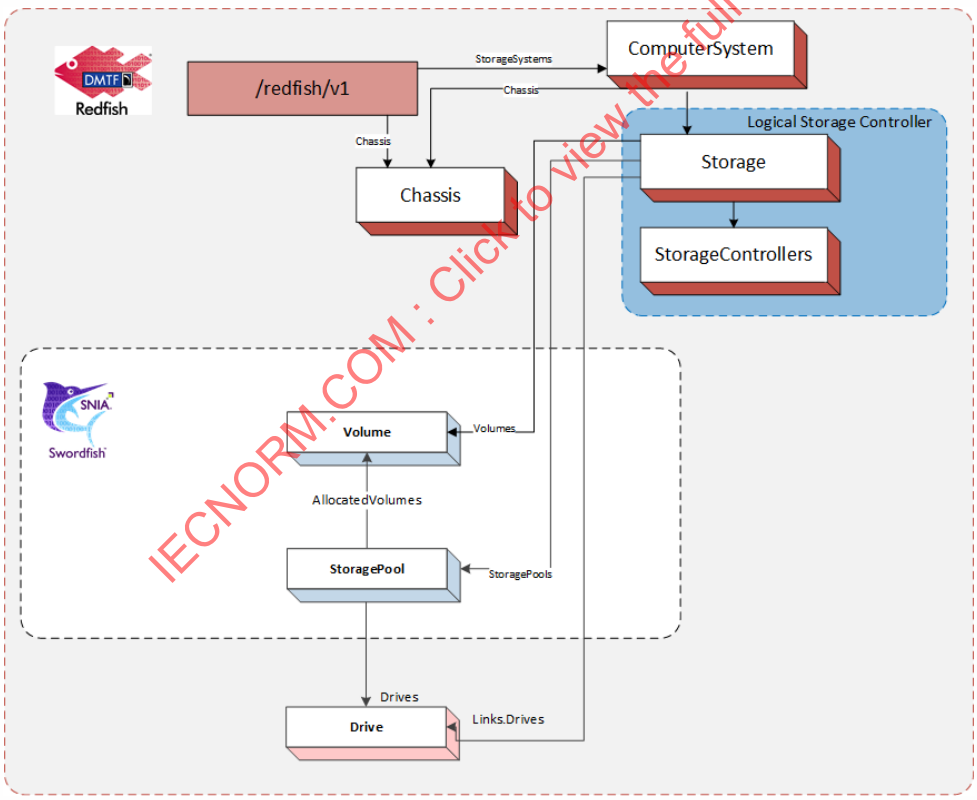

Figure 3: Integrated Configuration Example

2. Swordfish Standalone Configuration

The SSC uses separate ComputerSystem/StorageSystem model instantiation(s) to represent/model the logical controller(s) for the system.

The logical storage controller is modeled using Redfish a `ComputerSystems` with properties set as a `StorageSystem`. The physical infrastructure is modeled using Redfish `Chassis`. Managed resources are then connected to the `Storage` resource, including `Volumes`, `StoragePools`, `ConsistencyGroups`, and `StorageGroups`. These relationships are illustrated in Figure 4.

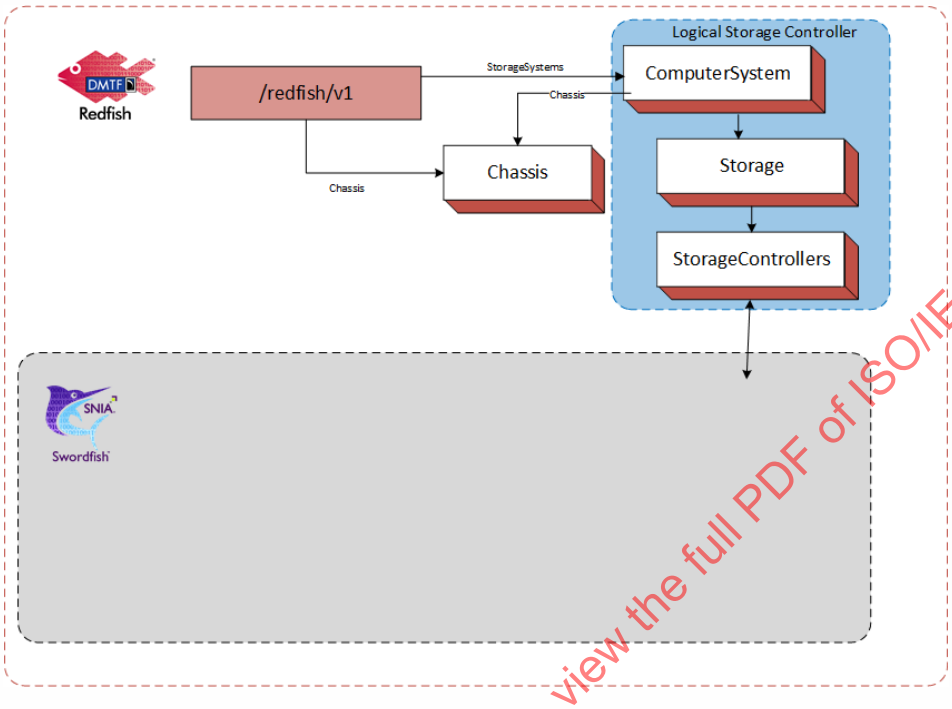

Figure 4: Standalone Configuration Overview

This configuration works well when the storage system needs a new `ComputerSystem` instance to model the logical controller. An example of a Storage System for a hosted service configuration is shown in Figure 5.

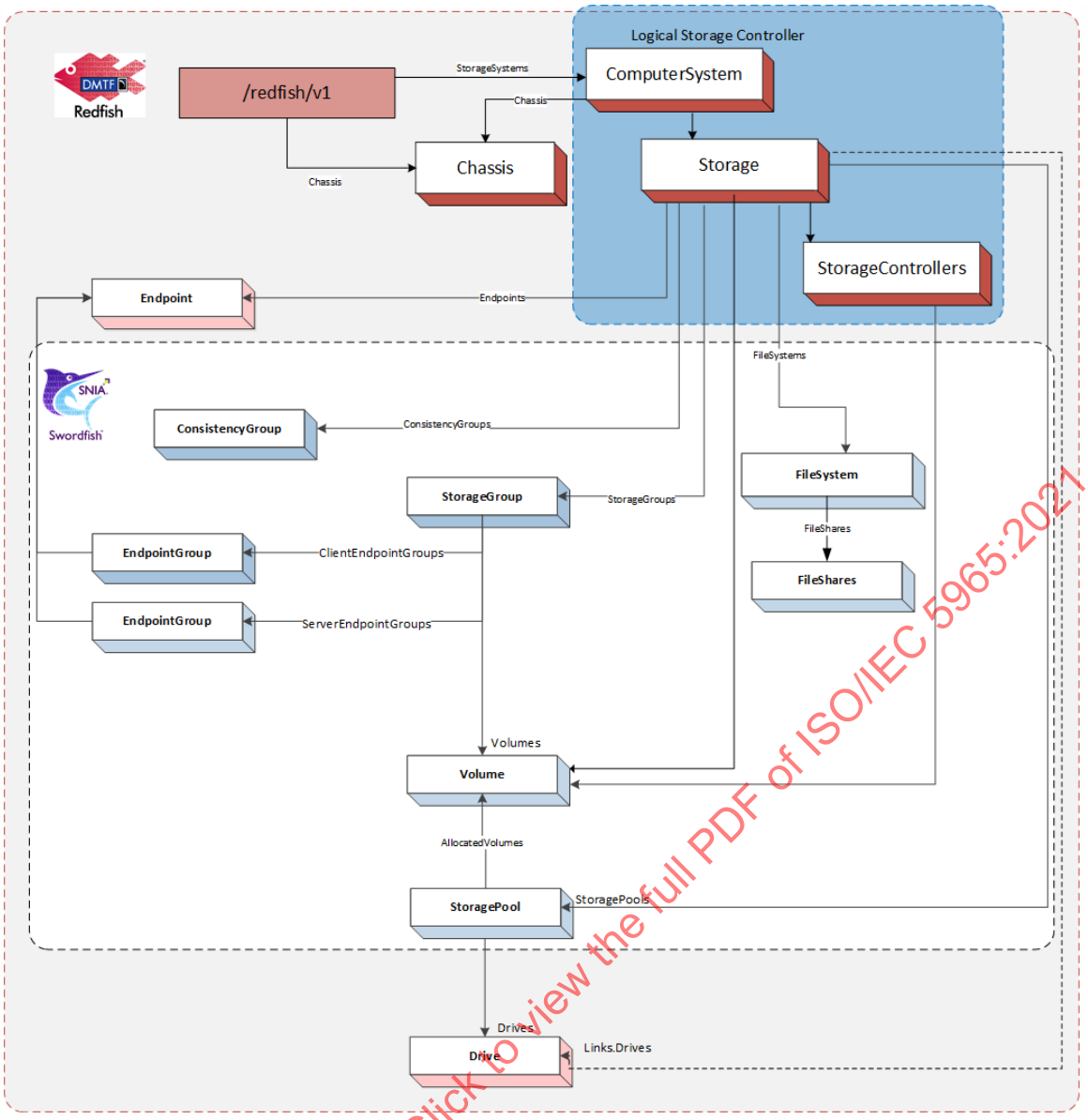

Figure 5: Standalone Configuration Example

5.4 The ServiceRoot and ServiceContainer entities

5.4.1 Overview

A **GET** of `/redfish/v1` will return the `ServiceRoot` entity. A **GET** of `/redfish/v1/odata` will return the `ServiceContainer` instances that represents the OData service document. Each of these instances provides links to the remainder of the system.

The following are the elements utilized for Swordfish management.

- `Systems`: A reference to a Systems resource collection;
- `Chassis`: A reference to a Chassis resource collection;

- `StorageSystems`: A reference to a StorageSystems resource collection.

5.4.2 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 collection and the Systems collection.

5.4.3 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 (like 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.

5.4.4 The StorageSystems resource collection

A reference to a `ComputerSystemCollection` with members of type `ComputerSystem` that support storage services. These `ComputerSystem` resources represent systems that support Swordfish storage management services. They will have an entry with the value of "StorageServer" in their `HostingRoles` property.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.

5.5 Swordfish model overview

5.5.1 The Storage resource

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

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

- `Drives`: A reference to a 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 collects `Endpoint` resources used to access storage.
- `EndpointGroups`: A reference to a resource collection that collects `EndpointGroups` resources.
- `FileSystems`: A reference to a resource collection that collects `FileSystem` resources.
- `StorageGroups`: A reference to a resource collection that collects `StorageGroup` resources.
- `ConsistencyGroups`: A reference to a resource collection that collects `ConsistencyGroup` resources.
- `StoragePools`: A reference to a resource collection that collects `StorageGroup` resources.
- `Volumes`: A reference to a resource collection that collects `Volume` resources.

5.5.1.1 The Endpoint resource

Endpoints represent one end of a protocol specific connection that supports sending or receiving messages according to a particular protocol.

5.5.1.2 The Endpoint Collection resource

The `Endpoint` Group is resource collection that references a set of `Endpoint` resources.

5.5.1.3 The ConsistencyGroup resource

ConsistencyGroups represent a set of volumes that are managed consistently and collectively as a group, to allow system and application level activities to be performed on a set of data that spans volumes. This activities include device-level replication activities 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.

5.5.1.4 The ConsistencyGroup Collection resource

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

5.5.1.5 The StorageGroup resource

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.

5.5.1.6 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 identify resources provisioned or supported by the storage pool:

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

5.5.1.7 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.

5.5.1.8 The FileSystem resource

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

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 the `SNIA.Swordfish.Discovery` supported feature 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:

- Class of Service
- Energy Star for Storage

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 Class of Service Feature

6.4.1 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 StorageService is located in the ServiceRoot (and ServiceContainer) via the StorageServices resource collection.

6.4.2 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`. These relationships are shown in Figure 6.

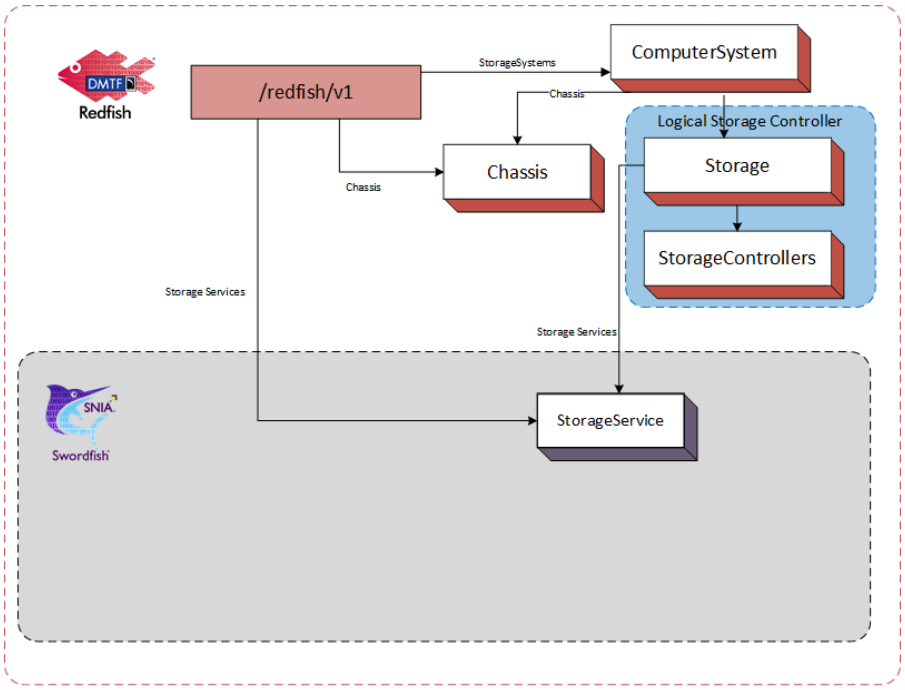

Figure 6: Integrated Service Overview

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.

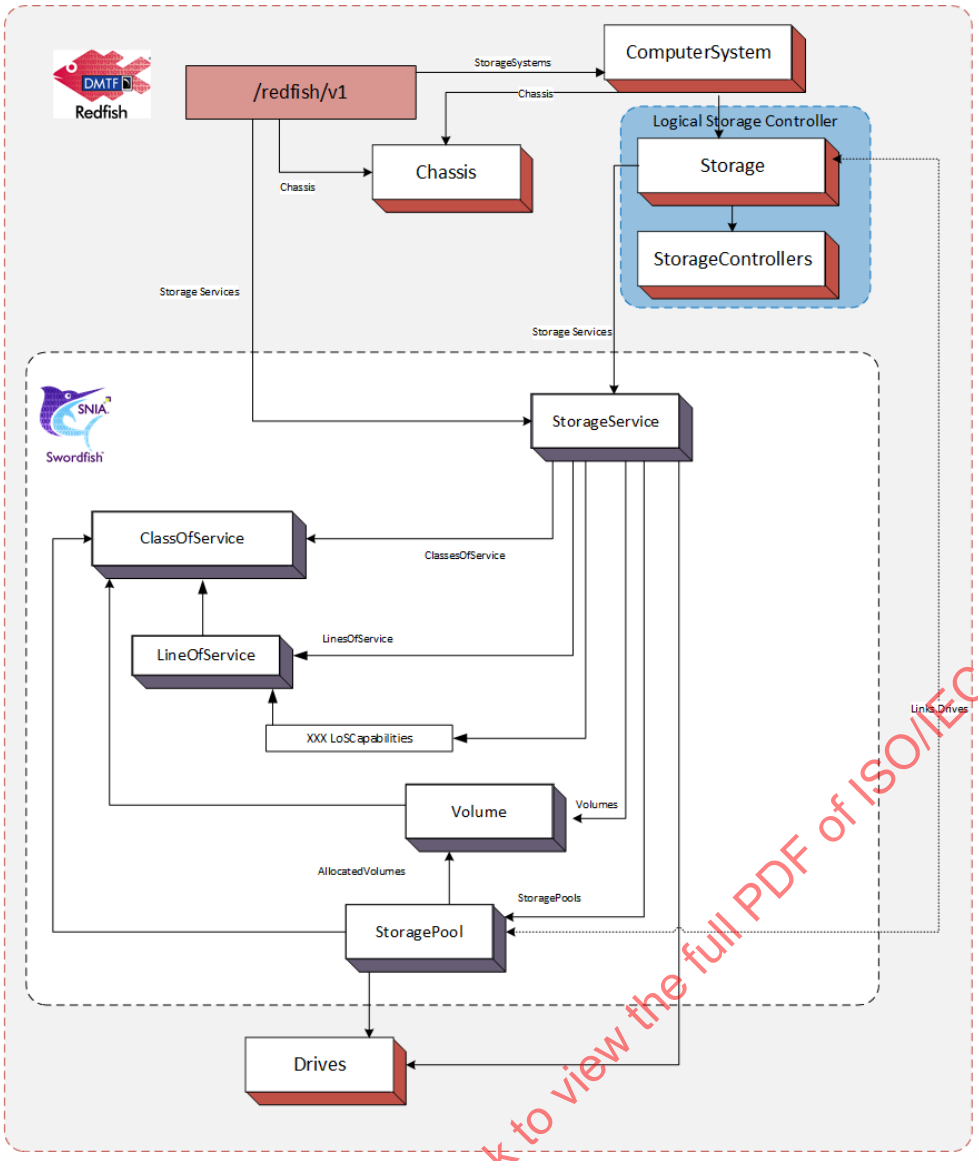

Figure 7: Integrated Service Example

2. Hosted Service Configuration

The storage systems managed by the Swordfish storage service are located in the `ServiceRoot` (and `ServiceContainer`) via the `StorageSystems` resource collection. They are modeled using Redfish `ComputerSystems`. The physical infrastructure is modeled using Redfish `Chassis`. These relationships are shown in Figure 8.

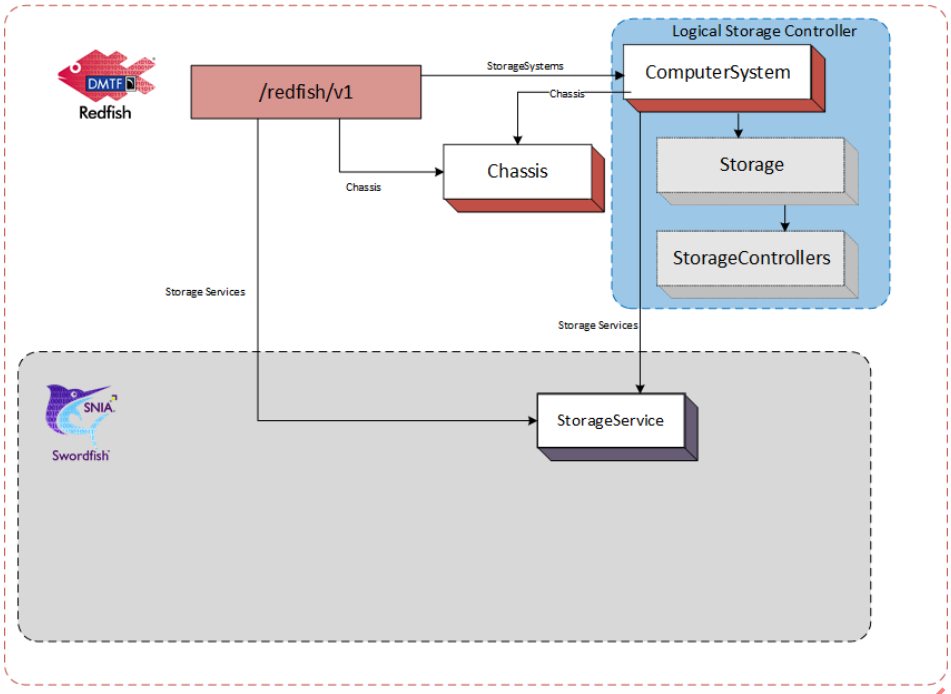

Figure 8: Hosted Service Overview

This configuration works well when the storage system hosts the storage service directly. 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.

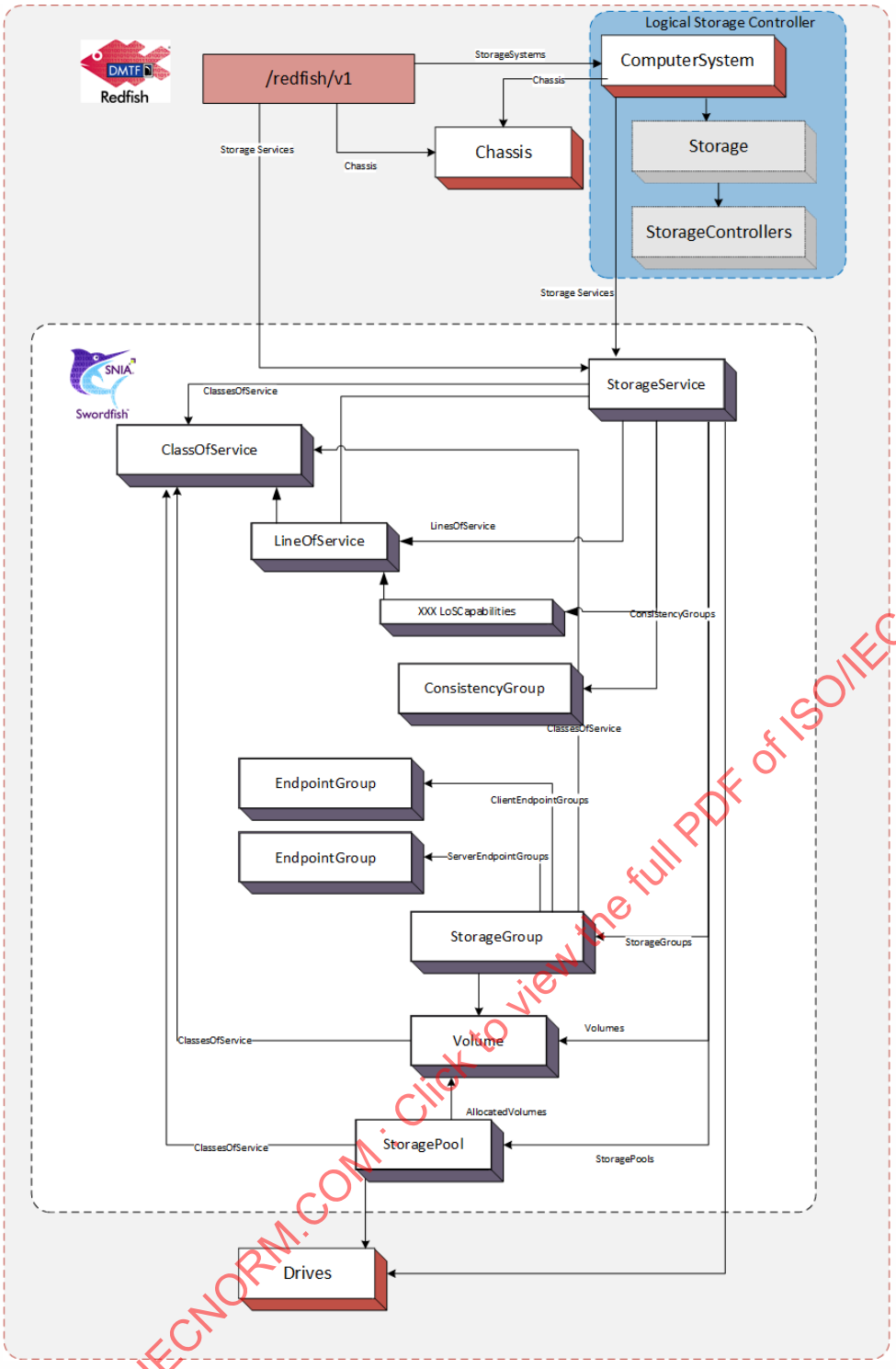

Figure 9: Hosted Service Example

6.4.3 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.

6.4.4 The StorageService resource

6.4.4.1 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 collects`Endpoint` 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`.

6.4.4.2 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.

6.4.4.3 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.

6.4.4.4 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.

6.4.4.5 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

6.4.4.6 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 and ComputerSystem. Swordfish implementations 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 Swordfish Extension of the Redfish ServiceRoot

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

The first is `StorageSystems`. This property references a collection of `ComputerSystem` resources that each support Swordfish functionality. Each such `ComputerSystem` shall have an entry in its `HostingRoles` property with the value of `StorageServer`.

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: Default and Nullable Interaction

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: Schema annotations

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

Table 8: Schema annotations, cont.

Name	Applies to	Description
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.
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 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.

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.

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)

8.3 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/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 `ServiceContainer 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`.

> Note: A Swordfish service is a Redfish service with extensions to support storage management. No additional service entry-points are necessary.

Both the `ServiceRoot` and `ServiceContainer` contain a resource collection named `Systems` that lists `ComputerSystem` instances. A `ComputerSystem` instance that supports Swordfish defined services will have a value of “StorageServer” in an entry of its `HostingRoles` property.

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.

8.4 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 StorageSystems requirements

For Hosted Service Configurations, this property of the `ServiceRoot` references a collection of `ComputerSystem` resources that each support Swordfish functionality. Each `ComputerSystem` included in the `StorageSystems` entry in the `ServiceRoot` shall have:

- an entry in its `HostingRoles` property with the value of `StorageServer`
- at least one entry in its `StorageServices.Members` property.

For Integrated Service Configurations, the StorageSystems concept is realized through the `StorageController` resource. Each `StorageController` instantiated as a Swordfish StorageSystem shall have:

- at least one entry in its `StorageController.Links` property `StorageServices` collection identifying related `StorageServices`

8.6 Entity Sets

The Swordfish model does not currently expose any explicitly defined entity sets. OData specifies that an entity set is defined for each NavigationProperty that is defined as a collection and that has the ContainsTarget attribute set to true. In all other cases, Swordfish assumes that an entity set is defined globally within the implementation for each entity type. This is effectively the same as if the entity sets were explicitly defined in the ServiceRoot entity container.

8.7 Addressing entities within a collection

An instance (entity) of an EntityType is uniquely identified within its entity set by its key. The URI for the reference may specify the key using one of two general strategies

1. OData recommends specifying the key value within parenthesis following the path segment that identifies the referencing entity set. (See clause “Canonical URL” in in OData)
2. Redfish common practice is to use an alternative form that adds a path segment having the value of the key following the path segment that identifies the referencing collection. (See clause “Alternate Key-as-Segment Syntax” in OData.)

A Swordfish implementation shall support both strategies.

8.8 Addressing members of a ResourceCollection

Redfish specifies that subclasses of ResourceCollection shall include a Members collection property (See clause “Collection resource response” in DSP0266)

Redfish allows a POST request to a ResourceCollection to be equivalent to the same POST request to the Members property of that ResourceCollection. For a particular ResourceCollection, if a Swordfish implementation supports either form, it shall support both.

It is common practice in Redfish to also eliminate the Members property from any request URI that navigates through a type hierarchy that includes a Member within a ResourceCollection. Care should be taken when defining and using a ResourceCollection subclass to not introduce ambiguities when an explicit reference to a Members property is dropped from a request URI.

8.9 HTTP status codes

8.9.1 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 both a clause reference (relative to the 1.7.0 version of the Redfish specification), and a small wording extract for additional context. For example:

Swordfish refines the requirements in **x.y.z** of 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.

8.9.2 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 clauses **7.5.1** and **12.2** of the *Redfish Specification*.

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.

8.9.3 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.

8.9.4 Actions

Swordfish supports the approach to Actions in **5.6.3** of 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 **7.10** of 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 Swordfish implementation. Each data type or entity within the schema includes a description that defines its implementation requirements and their interaction

9.2 Common properties

9.2.1 Properties defined for all Redfish schemas

The properties summarized in Table 9 are included in every Redfish schema, and therefore may be encountered in any Response payload. They are documented here to avoid repetition in the Resource Guide tables for each schema.

Table 9: Common Properties

Property	Datatype	Attributes	Notes
@odata.context	string	read-only	The @odata.context property is a URL to a metadata document with a fragment describing the data (typically rooted at the top-level singleton or collection). Technically the metadata document only has to define, or reference, any of the types that it directly uses, and different payloads could reference different metadata documents. However, since the @odata.context provides a root URL for resolving relative references (such as @odata.id's), we return the canonical metadata document.
@odata.id	string	read-only	The @odata.id property is a string that indicates the unique identifier of a resource.

Table 9: Common Properties, cont.

Property	Datatype	Attributes	Notes
@odata.type	string	read-only	The @odata.type property is a URL fragment that indicates the type of the resource.
Description	string	read-write	The Description property is used to convey a human-readable description of the resource.
Id	string	read-write	The Id property of a resource uniquely identifies the resource within the Resource Collection that contains it. The value of Id is unique within a Resource Collection.
Name	string	read-write	The Name property is used to convey a human-readable moniker for a resource. The type of the Name property is a string. The value of Name is NOT necessarily unique across resource instances within a Resource Collection.
Oem { }	object	read-write	This is the manufacturer/provider specific extension moniker used to divide the Oem object into sections. See the Resource schema for details on this property.

9.2.2 Links

The Links property represents 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. All directly referenced (subordinate) properties defined for a resource can be found from the root of the resource.

9.2.3 Actions

The Actions property contains the actions supported by a resource.

9.2.4 OEM

The OEM property is used for OEM extensions.

9.2.5 RelatedItem

The RelatedItem property is represented as a set of links. The links point to a resource, or part of a resource, as defined by that resource's schema definition.

This representation is not intended to be a strong linking methodology like other references. Instead it is used to show a relationship between elements or sub-elements in disparate parts of the service. For example, Fans may be in one area of the system and Processors in another area of the system. It could be that the relationship between the two is not obvious. The RelatedItem property can be used to show that one is related to the other. In this example, it might indicate that a specific fan is cooling a specific processor.

9.2.6 Status

9.2.6.1 Overview

The `Status` property is common to many Redfish schema. Its attributes are summarized in Table 10.

Table 10: Status property attributes

Health	string (enum)	read-only	This represents the health state of this resource in the absence of its dependent resources. See Health in Property Details, below, for the possible values of this property.
HealthRollup	string (enum)	read-only	This represents the overall health state from the view of this resource. See HealthRollup in Property Details, below, for the possible values of this property.
Oem { }	object	read-write	Oem extension object.
State	string (enum)	read-only	This indicates the known state of the resource, such as if it is enabled. See State in Property Details, below, for the possible values of this property.

9.2.6.2 Property details

Health:

This represents the health state of this resource in the absence of its dependent resources. Its possible values are summarized in Table 11.

Table 11: Health ENUM values, cont.

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

HealthRollup:

This represents the overall health state from the view of this resource. Its possible values are summarized in Table 12.

Table 12: HealthRollup ENUM values

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

State:

This indicates the known state of the resource, such as if it is enabled. Its possible values are summarized in Table 13.

Table 13: State ENUM values, cont.

string	Description
Absent	This function or resource is not present or not detected.
Disabled	This function or resource has been disabled.
Enabled	This function or resource has been enabled.
InTest	This function or resource is undergoing testing.
Quiesced	The element is enabled but only processes a restricted set of commands.
StandbyOffline	This function or resource is enabled, but awaiting an external action to activate it.
StandbySpare	This function or resource is part of a redundancy set and is awaiting a failover or other external action to activate it.
Starting	This function or resource is starting.
UnavailableOffline	This function or resource is present but cannot be used.
Updating	The element is updating and may be unavailable or degraded.

9.2.7 Location

9.2.7.1 Location properties overview

The properties of a Location object are summarized in Table 14.

Table 14: Location Properties

Property	Type	Notes	
AltitudeMeters	number (m)	read-only (null)	The altitude of the resource in meters.
Info	string	read-only (null)	This indicates the location of the resource.
InfoFormat	string	read-only (null)	This represents the format of the Info property.
Latitude	number (deg)	read-only (null)	The latitude resource.
Longitude	number (deg)	read-only (null)	The longitude resource in degrees.
Oem { }	object	read-write	Oem extension object. See the Resource schema for details on this property.
PartLocation {	object	read-write	Postal address of the addressed resource.
LocationOrdinalValue	number	read-only (null)	The number that represents the location of the part. If LocationType is slot and this unit is in slot 2 then the LocationOrdinalValue will be 2.
LocationType	string (enum)	read-only	The type of location of the part, such as slot, bay, socket and slot. See LocationType in Property Details, below, for the possible values of this property.
Orientation	string (enum)	read-only	The orientation for the ordering of the slot enumeration used by the LocationOrdinalValue property. See Orientation in Property Details, below, for the possible values of this property.

Table 14: Location Properties, cont.

Property	Type	Notes
Reference	string (enum)	read-only
ServiceLabel }	string	read-only (null)
Placement {	object	read-write
Rack	string	read-write (null)
RackOffset	number	read-write (null)
RackOffsetUnits	string (enum)	read-write
Row }	string	read-write (null)
PostalAddress {	object	read-write
AdditionalCode	string	read-write (null)
Building	string	read-write (null)
City	string	read-write (null)
Community	string	read-write (null)
Country	string	read-write (null)
District	string	read-write (null)
Division	string	read-write (null)
Floor	string	read-write (null)
GPSCoords	string	read-write (null)
HouseNumber	number	read-write (null)
HouseNumberSuffix	string	read-write (null)
Landmark	string	read-write (null)
LeadingStreetDirection	string	read-write (null)
Location	string	read-write (null)

Table 14: Location properties, cont.

Property	Type	Notes
Name	string	read-write (null)
Neighborhood	string	read-write (null)
POBox	string	read-write (null)
PlaceType	string	read-write (null)
PostalCode	string	read-write (null)
Road	string	read-write (null)
RoadBranch	string	read-write (null)
RoadPostModifier	string	read-write (null)
RoadPreModifier	string	read-write (null)
RoadSection	string	read-write (null)
RoadSubBranch	string	read-write (null)
Room	string	read-write (null)
Seat	string	read-write (null)
Street	string	read-write (null)
StreetSuffix	string	read-write (null)
Territory	string	read-write (null)
TrailingStreetSuffix	string	read-write (null)
Unit }	string	read-write (null)

9.2.7.2 Property details

LocationType:

The type of location of the part, such as slot, bay, socket and slot. The enum's potential values are sumamrized in Table 15.

Table 15: LocationType ENUM values

string	Description
Bay	Defines a bay as the type of location.
Connector	Defines a connector as the type of location.
Slot	Defines a slot as the type of location.
Socket	Defines a socket as the type of location.

Orientation:

The orientation for the ordering of the slot enumeration used by the LocationOrdinalValue property. The enum's potential values are sumamrized in Table 16.

Table 16: LocationType ENUM values

string	Description
BackToFront	Defines the ordering for the LocationOrdinalValue is back to front.
BottomToTop	Defines the ordering for the LocationOrdinalValue is bottom to top.
FrontToBack	Defines the ordering for the LocationOrdinalValue is front to back.
LeftToRight	Defines the ordering for the LocationOrdinalValue is left to right.
RightToLeft	Defines the ordering for the LocationOrdinalValue is right to left.
TopToBottom	Defines the ordering for the LocationOrdinalValue is top to bottom.

RackOffsetUnits:

The type of Rack Units in use. The enum's potential values are sumamrized in Table 17.

Table 17: LocationType ENUM values

string	Description
EIA_310	Defines a rack unit as being equal to 1.75 in (44.45 mm).
OpenU	Defines a rack unit as being equal to 48 mm (1.89 in).

Reference:

The reference point for the part location. This is used to give guidance as to the general location of the part. The enum's potential values are sumamrized in Table 18.

Table 18: LocationType ENUM values

string	Description
Bottom	Defines the part as being in the bottom of the unit.
Front	Defines the part as being in the front of the unit.
Left	Defines the part as being in the left of the unit.
Middle	Defines the part as being in the middle of the unit.
Rear	Defines the part as being in the rear of the unit.
Right	Defines the part as being in the right of the unit.
Top	Defines the part as being in the top of the unit.

9.3 Complex Types

The Table 19 defines a number of complex types that are used frequently in Swordfish schema. Multiple references to each complex type may be seen in later sections. For detailed definitions and properties contained in each complex type, refer to the schema definitions as referenced in the table.

Table 19: Common complex types

Type Name	Notes
Capacity {}	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 datastore. See the Capacity.v1_1_0 schema for details.
CapacityInfo {}	This composition may be used to represent the utilization of storage capacity. See the Capacity.v1_1_0 schema for details.
IOStatistics {}	See the IOStatistics.v1_0_1 schema for details on this property.

Table 19: Common complex types, cont.

Type Name	Notes
IOWorkload {}	This structure may be used to describe an IO Workload. See the IOPerformanceLoSCapabilities.v1_0_0 schema for details.
IOWorkloadComponent{}	This structure may be used to describe a component of an IO workload. See the IOPerformanceLoSCapabilities.v1_1_1 schema for details.
ReplicaInfo {}	The value shall define the characteristics of a replica. See the StorageReplicaInfo.v1_1_0 schema for details.
ReplicaRequest {}	See the DataProtectionLineOfService.v1_1_0 schema for details.
Schedule {}	Schedule a series of occurrences. See the Schedule.v1_1_0 schema for details.

9.4 CapacitySource 1.1.2

9.4.1 Decription

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.

9.4.2 Properties

The properties defined for the CapacitySource 1.1.2 object are summarized in Table 20.

Table 20: CapacitySource 1.1.2 properties

Property	Type	Notes
@odata.etag	string *read-only*	The value of this property shall be a string that is defined by the ETag HTTP header definition in RFC7232.
Actions *(v1.1.2+)* {}	object	The Actions property shall contain the available actions for this resource.
Description	string *read-only (null)*	This object represents the description of this resource. The resource values shall comply with the Redfish Specification-described requirements.
Id	string *read-only required*	This property represents an identifier for the resource. The resource values shall comply with the Redfish Specification-described requirements.
Name	string *read-only required*	This object represents the name of this resource or array member. The resource values shall comply with the Redfish Specification-described requirements. This string value shall be of the 'Name' reserved word format.
Oem {}	object	This property shall contain the OEM extensions. All values for properties that this object contains shall conform to the Redfish Specification-described requirements. See the *Resource* schema for details on this property.

Table 20: CapacitySource 1.1.2 properties, cont.

Property	Type	Notes
ProvidedCapacity {	object	The value shall be the amount of space that has been provided from the ProvidingDrives, ProvidingVolumes, ProvidingMemory or ProvidingPools.
Data {	object	The value shall be capacity information relating to provisioned user data.
AllocatedBytes	integer (By) *read-write (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) *read-only (null)*	The value shall be the number of logical bytes currently consumed in this data store for this data type.
GuaranteedBytes	integer (By) *read-write (null)*	The value shall be the number of bytes the storage system guarantees can be allocated in this data store for this data type.
ProvisionedBytes	integer (By) *read-write (null)*	The value shall be the maximum number of bytes that can be allocated in this data store for this data type.
}		
IsThinProvisioned	boolean *read-only (null)*	If the value is false, the capacity shall be fully allocated. The default value shall be false.

Table 20: CapacitySource 1.1.2 properties, cont.

Property	Type	Notes
Metadata {	object	The value shall be capacity information relating to provisioned system (non-user accessible) data.
AllocatedBytes	integer (By) *read-write (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) *read-only (null)*	The value shall be the number of logical bytes currently consumed in this data store for this data type.
GuaranteedBytes	integer (By) *read-write (null)*	The value shall be the number of bytes the storage system guarantees can be allocated in this data store for this data type.
ProvisionedBytes	integer (By) *read-write (null)*	The value shall be the maximum number of bytes that can be allocated in this data store for this data type.
}		
Snapshot {	object	The value shall be capacity information relating to provisioned snapshot or backup data.

Table 20: CapacitySource 1.1.2 properties, cont.

Property	Type	Notes
AllocatedBytes	integer (By) *read-write (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) *read-only (null)*	The value shall be the number of logical bytes currently consumed in this data store for this data type.
GuaranteedBytes	integer (By) *read-write (null)*	The value shall be the number of bytes the storage system guarantees can be allocated in this data store for this data type.
ProvisionedBytes	integer (By) *read-write (null)*	The value shall be the maximum number of bytes that can be allocated in this data store for this data type.
}		
}		
ProvidedClassOfService {	object	The value shall reference the provided ClassOfService from the ProvidingDrives, ProvidingVolumes, ProvidingMemoryChunks, ProvidingMemory or ProvidingPools.
@odata.id	string *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.
}		

Table 20: CapacitySource 1.1.2 properties, cont.

Property	Type	Notes
ProvidingDrives {	object	If present, the value shall be a reference to a contributing drive or drives. Contains a link to a resource.
@odata.id	string *read-only*	Link to Collection of *Drive*. See the Drive schema for details.
}		
ProvidingMemory *(v1.1+)* {	object	If present, the value shall be a reference to the contributing memory.
@odata.id	string *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.
}		
ProvidingMemoryChunks *(v1.1+)* {	object	If present, the value shall be a reference to the contributing memory chunks.
@odata.id	string *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.
}		
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 *read-only*	Link to Collection of *StoragePool*. See the StoragePool schema for details.
}		

Table 20: CapacitySource 1.1.2 properties, cont.

Property	Type	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 *read-only*	Link to Collection of *Volume*. See the Volume schema for details.
}		

9.5 ClassOfServiceCollection

9.5.1 URIs

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

9.5.2 Properties

The properties defined for the ClassOfServiceCollection object are summarized in Table 21.

Table 21: ClassOfServiceCollection properties

Property	Type	Notes
@odata.etag	string *read-only*	The value of this property shall be a string that is defined by the ETag HTTP header definition in RFC7232.
Description	string *read-only (null)*	This object represents the description of this resource. The resource values shall comply with the Redfish Specification-described requirements.
Members [{	array	The value of each member entry shall reference a ClassOfService or LineOfService resource.
@odata.id	string *read-only*	Link to a LineOfService resource. See the Links section and the *LineOfService* schema for details.
}]		
Members@odata.nextLink	string *read-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 *read-only*	This object represents the name of this resource or array member. The resource values shall comply with the Redfish Specification-described requirements. This string value shall be of the 'Name' reserved word format.

Table 21: ClassOfServiceCollection properties, cont.

Property	Type	Notes
Oem {}	object	This property shall contain the OEM extensions. All values for properties contained in this object shall conform to the Redfish Specification-described requirements. See the *Resource* schema for details on this property.

9.6 ConsistencyGroup 1.0.1

9.6.1 Decription

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.

9.6.2 URIs

/redfish/v1/StorageServices/*{StorageServiceId}*/ConsistencyGroups/*{ConsistencyGroupId}*
/redfish/v1/StorageServices/*{StorageServiceId}*/Volumes/*{VolumeId}*/ConsistencyGroups/*{ConsistencyGroupId}*
/redfish/v1/Systems/*{ComputerSystemId}*/Storage/*{StorageId}*/ConsistencyGroups/*{ConsistencyGroupId}*

9.6.3 Properties

The properties defined for the ConsistencyGroup 1.0.1 object are summarized in Table 22.

Table 22: ConsistencyGroup 1.0.1 properties

Property	Type	Notes
@odata.etag	string *read-only*	The value of this property shall be a string that is defined by the ETag HTTP header definition in RFC7232.
Actions {	object	The Actions property shall contain the available actions for this resource.

Table 22: ConsistencyGroup 1.0.1 properties, cont.

Property	Type	Notes
#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.*
#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.*
#ConsistencyGroup.RemoveReplicaRelationship {}	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.*

Table 22: ConsistencyGroup 1.0.1 properties, cont.

Property	Type	Notes
#ConsistencyGroup.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.*
#ConsistencyGroup.ReverseReplicationRelationship {}	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.*
#ConsistencyGroup.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.*

Table 22: ConsistencyGroup 1.0.1 properties, cont.

Property	Type	Notes
#ConsistencyGroup.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) *read-write (null)*	The property shall set the consistency method used by this group. *For the possible property values, see ConsistencyMethod in Property details.*
ConsistencyType	string (enum) *read-write (null)*	The property shall set the consistency type used by this group. *For the possible property values, see ConsistencyType in Property details.*
Description	string *read-only (null)*	This object represents the description of this resource. The resource values shall comply with the Redfish Specification-described requirements.
Id	string *read-only required*	This property represents an identifier for the resource. The resource values shall comply with the Redfish Specification-described requirements.

Table 22: ConsistencyGroup 1.0.1 properties, cont.

Property	Type	Notes
IsConsistent	boolean *read-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.
Oem {}	object	This property shall contain the OEM extensions. All values for properties contained in this object shall conform to the Redfish Specification-described requirements. See the *Resource* schema for details on this property.
}		
Name	string *read-only required*	This object represents the name of this resource or array member. The resource values shall comply with the Redfish Specification-described requirements. This string value shall be of the 'Name' reserved word format.
Oem {}	object	This property shall contain the OEM extensions. All values for properties that this object contains shall conform to the Redfish Specification-described requirements. See the *Resource* schema for details on this property.

Table 22: ConsistencyGroup 1.0.1 properties, cont.

Property	Type	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 *read-only*	Link to a ReplicaInfo resource. See the Links section and the *StorageReplicaInfo* schema for details.
}		
ReplicaTargets [{	array	The value shall reference the target replicas that are sourced by this replica.
@odata.id	string *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.
}]		
Status {}	object	The property shall contain the status of the ConsistencyGroup. See the *Resource* schema for details on this property.
Volumes [{	array	An array of references to volumes managed by this storage group.
@odata.id	string *read-only*	Link to a Volume resource. See the Links section and the *Volume* schema for details.
}]		

9.6.4 Actions

9.6.4.1 AssignReplicaTarget

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.

URIs:

/redfish/v1/StorageServices/*{StorageServiceId}*/ConsistencyGroups/*{ConsistencyGroupId}*/Actions/ConsistencyGroup.AssignReplicaTarget
/redfish/v1/StorageServices/*{StorageServiceId}*/Volumes/*{VolumeId}*/ConsistencyGroups/*{ConsistencyGroupId}*/Actions/ConsistencyGroup.AssignReplicaTarget
/redfish/v1/Systems/*{ComputerSystemId}*/Storage/*{StorageId}*/ConsistencyGroups/*{ConsistencyGroupId}*/Actions

/ConsistencyGroup.AssignReplicaTarget

Parameters

The parameters for the action (which can be included in the POST body to the URI shown in the 'target' property of the action) are summarized in Table 23.

Table 23: #ConsistencyGroup.AssignReplicaTarget action parameters

Parameter	Type	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.*
ReplicaUpdateMode	string (enum) *required*	This parameter shall specify the replica update mode. *For the possible property values, see ReplicaUpdateMode in Property details.*
TargetConsistencyGroup	string *required*	This parameter shall contain the Uri to the existing consistency group.

9.6.4.2 CreateReplicaTarget

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.

URIs:

/redfish/v1/StorageServices/*{StorageServiceId}*/ConsistencyGroups/*{ConsistencyGroupId}*/Actions/ConsistencyGroup.CreateReplicaTarget
/redfish/v1/StorageServices/*{StorageServiceId}*/Volumes/*{VolumeId}*/ConsistencyGroups/*{ConsistencyGroupId}*/Actions/ConsistencyGroup.CreateReplicaTarget
/redfish/v1/Systems/*{ComputerSystemId}*/Storage/*{StorageId}*/ConsistencyGroups/*{ConsistencyGroupId}*/Actions/ConsistencyGroup.CreateReplicaTarget

Parameters

The parameters for the action (which can be included in the POST body to the URI shown in the 'target' property of the action) are summarized in Table 24.

Table 24: ConsistencyGroup.CreateReplicaTarget action parameters, cont.

Parameter	Type	Notes
ConsistencyGroupName	string *required*	This parameter shall contain the Name for the target consistency group.
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.*
ReplicaUpdateMode	string (enum) *required*	This parameter shall specify the replica update mode. *For the possible property values, see ReplicaUpdateMode in Property details.*
TargetStoragePool	string *required*	This parameter shall contain the Uri to the existing StoragePool in which to create the target consistency group.

9.6.4.3 RemoveReplicaRelationship

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.

URIs:

/redfish/v1/StorageServices/*{StorageServiceId}*/ConsistencyGroups/*{ConsistencyGroupId}*/Actions/ConsistencyGroup.RemoveReplicaRelationship
/redfish/v1/StorageServices/*{StorageServiceId}*/Volumes/*{VolumeId}*/ConsistencyGroups/*{ConsistencyGroupId}*/Actions/ConsistencyGroup.RemoveReplicaRelationship
/redfish/v1/Systems/*{ComputerSystemId}*/Storage/*{StorageId}*/ConsistencyGroups/*{ConsistencyGroupId}*/Actions/ConsistencyGroup.RemoveReplicaRelationship

Parameters

The parameters for the action (which can be included in the POST body to the URI shown in the 'target' property of the action) are summarized in Table 25.

Table 25: #ConsistencyGroup.RemoveReplicaRelationship action parameters

Parameter	Type	Notes
DeleteTargetConsistencyGroup	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.
TargetConsistencyGroup	string *required*	This parameter shall contain the Uri to the existing target consistency group.

9.6.4.4 ResumeReplication

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

URIs:

/redfish/v1/StorageServices/*{StorageServiceId}*/ConsistencyGroups/*{ConsistencyGroupId}*/Actions/ConsistencyGroup.ResumeReplication
/redfish/v1/StorageServices/*{StorageServiceId}*/Volumes/*{VolumeId}*/ConsistencyGroups/*{ConsistencyGroupId}*/Actions/ConsistencyGroup.ResumeReplication
/redfish/v1/Systems/*{ComputerSystemId}*/Storage/*{StorageId}*/ConsistencyGroups/*{ConsistencyGroupId}*/Actions/ConsistencyGroup.ResumeReplication

Parameters

The parameters for the action (which can be included in the POST body to the URI shown in the 'target' property of the action) are summarized in Table 26.

Table 26: #ConsistencyGroup.ResumeReplication action parameters

Parameter	Type	Notes
TargetConsistencyGroup	string *required*	This parameter shall contain the Uri to the existing target consistency group.

9.6.4.5 ReverseReplicationRelationship

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

URIs:

/redfish/v1/StorageServices/*{StorageServiceId}*/ConsistencyGroups/*{ConsistencyGroupId}*/Actions/ConsistencyGroup.ReverseReplicationRelationship
/redfish/v1/StorageServices/*{StorageServiceId}*/Volumes/*{VolumeId}*/ConsistencyGroups/*{ConsistencyGroupId}*/

Actions/ConsistencyGroup.ReverseReplicationRelationship
/redfish/v1/Systems/*{ComputerSystemId}*/Storage/*{StorageId}*/ConsistencyGroups/*{ConsistencyGroupId}*/Actions
/ConsistencyGroup.ReverseReplicationRelationship

Parameters

The parameters for the action (which can be included in the POST body to the URI shown in the 'target' property of the action) are summarized in Table 27.

Table 27: ConsistencyGroup.ReverseReplicationRelationship action parameters

Parameter	Type	Notes
TargetConsistencyGroup	string *required*	This parameter shall contain the Uri to the existing target consistency group.

9.6.4.6 SplitReplication

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

URIs:

/redfish/v1/StorageServices/*{StorageServiceId}*/ConsistencyGroups/*{ConsistencyGroupId}*/Actions/ConsistencyGr
oup.SplitReplication
/redfish/v1/StorageServices/*{StorageServiceId}*/Volumes/*{VolumeId}*/ConsistencyGroups/*{ConsistencyGroupId}*/
Actions/ConsistencyGroup.SplitReplication
/redfish/v1/Systems/*{ComputerSystemId}*/Storage/*{StorageId}*/ConsistencyGroups/*{ConsistencyGroupId}*/Actions
/ConsistencyGroup.SplitReplication

Parameters

The parameters for the action (which can be included in the POST body to the URI shown in the 'target' property of the action) are summarized in Table 28.

Table 28: #ConsistencyGroup.SplitReplication action parameters

Parameter	Type	Notes
TargetConsistencyGroup	string *required*	This parameter shall contain the Uri to the existing target consistency group.

9.6.4.7 SuspendReplication

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

URIs:

/redfish/v1/StorageServices/*{StorageServiceId}*/ConsistencyGroups/*{ConsistencyGroupId}*/Actions/ConsistencyGroup.SuspendReplication
/redfish/v1/StorageServices/*{StorageServiceId}*/Volumes/*{VolumeId}*/ConsistencyGroups/*{ConsistencyGroupId}*/Actions/ConsistencyGroup.SuspendReplication
/redfish/v1/Systems/*{ComputerSystemId}*/Storage/*{StorageId}*/ConsistencyGroups/*{ConsistencyGroupId}*/Actions/ConsistencyGroup.SuspendReplication

Parameters

The parameters for the action (which can be included in the POST body to the URI shown in the 'target' property of the action) are summarized in Table 29.

Table 29: #ConsistencyGroup.SuspendReplication action parameters

Parameter	Type	Notes
TargetConsistencyGroup	string *required*	This parameter shall contain the Uri to the existing target consistency group.

9.6.5 Property details

9.6.5.1 ConsistencyMethod

The property shall set the consistency method used by this group.

The values defined for the ConsistencyMethod ENUM are summarized in Table 30.

Table 30: ConsistencyMethod ENUM 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.

9.6.5.2 ConsistencyType

The property shall set the consistency type used by this group.

The values defined for the ConsistencyType ENUM are summarized in Table 31.

Table 31: ConsistencyType ENUM 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.

9.6.5.3 ReplicaType

This parameter shall contain the type of replica relationship to be created.

The values defined for the ReplicaType ENUM are summarized in Table 32.

Table 32: ReplicaType ENUM 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.

9.6.5.4 ReplicaUpdateMode

This parameter shall specify the replica update mode.

The values defined for the ReplicaUpdateMode ENUM are summarized in Table 33.

Table 33: ReplicaUpdateMode ENUM 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.

9.7 ConsistencyGroupCollection

9.7.1 URIs

/redfish/v1/StorageServices/*{StorageServiceId}*/ConsistencyGroups
/redfish/v1/StorageServices/*{StorageServiceId}*/Volumes/*{VolumeId}*/ConsistencyGroups
/redfish/v1/Systems/*{ComputerSystemId}*/Storage/*{StorageId}*/ConsistencyGroups

9.7.2 Properties

The properties defined for the ConsistencyGroupCollection object are summarized in Table 34.

Table 34: ConsistencyGroupCollection properties, cont.

Property	Type	Notes
@odata.etag	string *read-only*	The value of this property shall be a string that is defined by the ETag HTTP header definition in RFC7232.
Description	string *read-only (null)*	This object represents the description of this resource. The resource values shall comply with the Redfish Specification-described requirements.
Members [{	array	The value of each member entry shall reference a ConsistencyGroup resource.
@odata.id	string *read-only*	Link to a ConsistencyGroup resource. See the Links section and the *ConsistencyGroup* schema for details.
}]		
Members@odata.nextLink	string *read-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 *read-only*	This object represents the name of this resource or array member. The resource values shall comply with the Redfish Specification-described requirements. This string value shall be of the 'Name' reserved word format.
Oem {}	object	This property shall contain the OEM extensions. All values for properties contained in this object shall conform to the Redfish Specification-described requirements. See the *Resource* schema for details on this property.

9.8 DataProtectionLoSCapabilities 1.1.3

9.8.1 Decription

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.

9.8.2 URIs

/redfish/v1/StorageServices/*{StorageServiceId}*/DataProtectionLoSCapabilities

9.8.3 Properties

The properties defined for the DataProtectionLoSCapabilities 1.1.3 object are summarized in Table 35.

Table 35: DataProtectionLoSCapabilities 1.1.3 properties

Property	Type	Notes
@odata.etag	string *read-only*	The value of this property shall be a string that is defined by the ETag HTTP header definition in RFC7232.
Actions *(v1.1+)* {}	object	The Actions property shall contain the available actions for this resource.
Description	string *read-only (null)*	This object represents the description of this resource. The resource values shall comply with the Redfish Specification-described requirements.
Id	string *read-only required*	This property represents an identifier for the resource. The resource values shall comply with the Redfish Specification-described requirements.
Identifier {}	object	The value shall be unique within the managed ecosystem. See the *v1_9_1.v1_9_1* schema for details on this property.

Table 35: DataProtectionLoSCapabilities 1.1.3 properties, cont

Property	Type	Notes
Links {	object	The value of this property shall contains links to other resources that are not contained in this resource.
Oem {}	object	This property shall contain the OEM extensions. All values for properties contained in this object shall conform to the Redfish Specification-described requirements. See the *Resource* schema for details on this property.
SupportedReplicaOptions [{	array	The collection shall contain known and supported replica Classes of Service.
@odata.id	string *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.
}]		
SupportedReplicaOptions@odata.count	integer *read-only*	The value of this property shall be an integer representing the number of items in a collection.
}		
Name	string *read-only required*	This object represents the name of this resource or array member. The resource values shall comply with the Redfish Specification-described requirements. This string value shall be of the 'Name' reserved word format.

Table 35: DataProtectionLoSCapabilities 1.1.3 properties, cont

Property	Type	Notes
Oem {}	object	This property shall contain the OEM extensions. All values for properties that this object contains shall conform to the Redfish Specification-described requirements. See the *Resource* schema for details on this property.
SupportedLinesOfService [{	array	The collection shall contain known and supported DataProtectionLinesOfService.
@odata.id	string *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.
}]		
SupportedMinLifetimes [ ]	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.
SupportedRecoveryGeographicObjectives [ ]	array (string (enum)) *read-write (null)*	The value of each entry shall specify a supported failure domain. The enumeration literals of this enumeration shall represent a geographic scope in which all components within that scope have similar vulnerabilities. *For the possible property values, see SupportedRecoveryGeographicObjectives in Property details.*

Table 35: DataProtectionLoSCapabilities 1.1.3 properties, cont.

Property	Type	Notes
SupportedRecoveryPointObjectiveTimes [ ]	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.
SupportedRecoveryTimeObjectives [ ]	array (string (enum)) *read-write (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. The enumeration literals shall represent the relative time required to make a replica available as a source. *For the possible property values, see SupportedRecoveryTimeObjectives in Property details.*
SupportedReplicaTypes [ ]	array (string (enum)) *read-write (null)*	The value of each entry shall specify a supported replica type. The enumeration literals may be used to specify the intended outcome of the replication. *For the possible property values, see SupportedReplicaTypes in Property details.*
SupportsIsolated	boolean *read-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.

9.8.4 Property details

9.8.4.1 SupportedRecoveryGeographicObjectives

The value of each entry shall specify a supported failure domain. The enumeration literals of this enumeration shall represent a geographic scope in which all components within that scope have similar vulnerabilities.

The values defined for the SupportedRecoveryGeographicObjectives ENUM are summarized in Table 36.

Table 36: SupportedRecoveryGeographicObjectives ENUM 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.

9.8.4.2 SupportedRecoveryTimeObjectives

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. The enumeration literals shall represent the relative time required to make a replica available as a source.

The values defined for the SupportedRecoveryTimeObjectives ENUM are summarized in Table 37.

Table 37: SupportedRecoveryTimeObjectives ENUM 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.

9.8.4.3 SupportedReplicaTypes

The value of each entry shall specify a supported replica type. The enumeration literals may be used to specify the intended outcome of the replication.

The values defined for the SupportedReplicaTypes ENUM are summarized in Table 38.

Table 38: SupportedReplicaTypes ENUM values, cont.

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.

9.9 DataSecurityLoSCapabilities 1.1.3

9.9.1 Decription

This resource may be used to describe data security capabilities.

9.9.2 URIs

/redfish/v1/StorageServices/*{StorageServiceId}*/DataSecurityLoSCapabilities

9.9.3 Properties

The properties defined for the DataSecurityLoSCapabilities 1.1.3 object are summarized in Table 39.

Table 39: DataSecurityLoSCapabilities 1.1.3 properties

Property	Type	Notes
@odata.etag	string *read-only*	The value of this property shall be a string that is defined by the ETag HTTP header definition in RFC7232.
Actions *(v1.1+)* {}	object	The Actions property shall contain the available actions for this resource.
Description	string *read-only (null)*	This object represents the description of this resource. The resource values shall comply with the Redfish Specification-described requirements.
Id	string *read-only required*	This property represents an identifier for the resource. The resource values shall comply with the Redfish Specification-described requirements.
Identifier {}	object	The value identifies this resource. The value shall be unique within the managed ecosystem. See the *v1_9_1.v1_9_1* schema for details on this property.

Table 39: DataSecurityLoSCapabilities 1.1.3 properties, cont.

Property	Type	Notes
Name	string *read-only required*	This object represents the name of this resource or array member. The resource values shall comply with the Redfish Specification-described requirements. This string value shall be of the 'Name' reserved word format.
Oem {}	object	This property shall contain the OEM extensions. All values for properties that this object contains shall conform to the Redfish Specification-described requirements. See the *Resource* schema for details on this property.
SupportedAntivirusEngineProviders [ ]	array (string, null) *read-write*	The entry values shall specify supported AntiVirus providers.
SupportedAntivirusScanPolicies [ ]	array (string (enum)) *read-write (null)*	The enumeration literal shall specify supported policies that trigger an AntiVirus scan. The enumberation literals shall specify types of antivirus scan triggers. *For the possible property values, see SupportedAntivirusScanPolicies in Property details.*

Table 39: DataSecurityLoSCapabilities 1.1.3 properties, cont.

Property	Type	Notes
SupportedChannelEncryptionStrengths [ ]	array (string (enum)) *read-write (null)*	The enumeration literal shall specify supported key sizes in a symmetric encryption algorithm (AES) for transport channel encryption. The enumeration literals shall specify Key sizes in a symmetric encryption algorithm, (see NIST SP 800-57 part 1 (http:/csrc.nist.gov/publications/nistpubs/800-57/sp800-57_part1_rev3_general.pdf). *For the possible property values, see SupportedChannelEncryptionStrengths in Property details.*
SupportedDataSanitizationPolicies [ ]	array (string (enum)) *read-write (null)*	The enumeration literal shall specify supported data sanitization policies. The enumberation literals shall specify types of data sanitization policies. *For the possible property values, see SupportedDataSanitizationPolicies in Property details.*
SupportedHostAuthenticationTypes [ ]	array (string (enum)) *read-write (null)*	The enumeration literal shall specify supported authentication types for hosts (servers) or initiator endpoints. The enumeration literals shall specify authentication algorithms. *For the possible property values, see SupportedHostAuthenticationTypes in Property details.*
SupportedLinesOfService [{	array	The collection shall contain supported DataSecurity service options.
@odata.id	string *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.
}]		

Table 39: DataSecurityLoSCapabilities 1.1.3 properties, cont.

Property	Type	Notes
SupportedMediaEncryptionStrengths [ ]	array (string (enum)) *read-write (null)*	The enumeration literal shall specify supported key sizes in a symmetric encryption algorithm (AES) for media encryption.
		The enumeration literals shall specify Key sizes in a symmetric encryption algorithm, (see NIST SP 800-57 part 1 (http:/csrc.nist.gov/publications/nistpubs/800-57/sp800-57_part1_rev3_general.pdf). *For the possible property values, see SupportedMediaEncryptionStrengths in Property details.*
SupportedSecureChannelProtocols [ ]	array (string (enum)) *read-write (null)*	The enumeration literal shall specify supported protocols that provide encrypted communication. The enumeration literals shall specify types of Secure channel protocols. *For the possible property values, see SupportedSecureChannelProtocols in Property details.*
SupportedUserAuthenticationTypes [ ]	array (string (enum)) *read-write (null)*	The enumeration literal shall specify supported authentication types for users (or programs). The enumeration literals shall specify authentication algorithms. *For the possible property values, see SupportedUserAuthenticationTypes in Property details.*

9.9.4 Property details

9.9.4.1 SupportedAntivirusScanPolicies

The enumeration literal shall specify supported policies that trigger an AntiVirus scan. The enumberation literals shall specify types of antivirus scan triggers.

The values defined for the SupportedAntivirusScanPolicies ENUM are summarized in Table 40.

Table 40: SupportedAntivirusScanPolicies ENUM 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.

9.9.4.2 SupportedChannelEncryptionStrengths

The enumeration literal shall specify supported key sizes in a symmetric encryption algorithm (AES) for transport channel encryption. The enumeration literals shall specify Key sizes in a symmetric encryption algorithm, (see NIST SP 800-57 part 1 (http:/csrc.nist.gov/publications/nistpubs/800-57/sp800-57_part1_rev3_general.pdf).

The values defined for the SupportedChannelEncryptionStrengths ENUM are summarized in Table 41.

Table 41: SupportedChannelEncryptionStrengths ENUM 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.

9.9.4.3 SupportedDataSanitizationPolicies

The enumeration literal shall specify supported data sanitization policies. The enumberation literals shall specify types of data sanitization policies.

The values defined for the SupportedDataSanitizationPolicies ENUM are summarized in Table 42.

Table 42: SupportedDataSanitizationPolicies ENUM 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.

9.9.4.4 SupportedHostAuthenticationTypes

The enumeration literal shall specify supported authentication types for hosts (servers) or initiator endpoints. The enumeration literals shall specify authentication algorithms.

The values defined for the SupportedHostAuthenticationTypes ENUM are summarized in Table 43.

Table 43: SupportedHostAuthenticationTypes ENUM values, cont.

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.
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.

9.9.4.5 SupportedMediaEncryptionStrengths

The enumeration literal shall specify supported key sizes in a symmetric encryption algorithm (AES) for media encryption. The enumeration literals shall specify Key sizes in a symmetric encryption algorithm, (see NIST SP 800-57 part 1 (http:/csrc.nist.gov/publications/nistpubs/800-57/sp800-57_part1_rev3_general.pdf).

The values defined for the SupportedMediaEncryptionStrengths ENUM are summarized in Table 44.

Table 44: SupportedMediaEncryptionStrengths ENUM 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.

9.9.4.6 SupportedSecureChannelProtocols

The enumeration literal shall specify supported protocols that provide encrypted communication. The enumeration literals shall specify types of Secure channel protocols.

The values defined for the SupportedSecureChannelProtocols ENUM are summarized in Table 45.

Table 45: SupportedSecureChannelProtocols ENUM values, cont.

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 ISO/IEC 20648

9.9.4.7 SupportedUserAuthenticationTypes

The enumeration literal shall specify supported authentication types for users (or programs). The enumeration literals shall specify authentication algorithms.

The values defined for the SupportedUserAuthenticationTypes ENUM are summarized in Table 46.

Table 46: SupportedUserAuthenticationTypes ENUM 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.
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.

9.10 DataStorageLoSCapabilities 1.2.1

9.10.1 Decription

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

9.10.2 URIs

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

9.10.3 Properties

The properties defined for the DataStorageLoSCapabilities 1.2.1 object are summarized in Table 47.

Table 47: DataStorageLoSCapabilities 1.2.1 properties

Property	Type	Notes
@odata.etag	string *read-only*	The value of this property shall be a string that is defined by the ETag HTTP header definition in RFC7232.
Actions *(v1.1+)* {}	object	The Actions property shall contain the available actions for this resource.
Description	string *read-only* *(null)*	This object represents the description of this resource. The resource values shall comply with the Redfish Specification-described requirements.
Id	string *read-only* *required*	This property represents an identifier for the resource. The resource values shall comply with the Redfish Specification-described requirements.
Identifier {}	object	The value shall be unique within the managed ecosystem. See the *v1_9_1.v1_9_1* schema for details on this property.
MaximumRecoverableCapacitySourceCount *(v1.2+)*	integer *read-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.

Table 47: DataStorageLoSCapabilities 1.2.1 properties

Property	Type	Notes
Name	string *read-only required*	This object represents the name of this resource or array member. The resource values shall comply with the Redfish Specification-described requirements. This string value shall be of the 'Name' reserved word format.
Oem {}	object	This property shall contain the OEM extensions. All values for properties that this object contains shall conform to the Redfish Specification-described requirements. See the *Resource* schema for details on this property.
SupportedAccessCapabilities [ ]	array (string (enum)) *read-write (null)*	Each entry specifies a storage access capability. StorageAccessCapability enumeration literals may be used to describe abilities to read or write storage. *For the possible property values, see SupportedAccessCapabilities in Property details.*
SupportedLinesOfService [{	array	The collection shall contain known and supported DataStorageLinesOfService.
@odata.id	string *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.
}]		
SupportedProvisioningPolicies [ ]	array (string (enum)) *read-write (null)*	This collection specifies supported storage allocation policies. The enumeration literals may be used to specify space provisioning policy. *For the possible property values, see SupportedProvisioningPolicies in Property details.*

Table 47: DataStorageLoSCapabilities 1.2.1 properties, cont.

Property	Type	Notes
SupportedRecoveryTimeObjectives [ ]	array (string (enum)) *read-write (null)*	This collection specifies supported expectations for time to access the primary store after recovery. The enumeration literals shall represent the relative time required to make a replica available as a source. *For the possible property values, see SupportedRecoveryTimeObjectives in Property details.*
SupportsSpaceEfficiency	boolean *read-write (null)*	The value specifies whether storage compression or deduplication is supported. The default value for this property is false.

9.10.4 Property details

9.10.4.1 SupportedAccessCapabilities

Each entry specifies a storage access capability. StorageAccessCapability enumeration literals may be used to describe abilities to read or write storage.

The values defined for the SupportedAccessCapabilities ENUM are summarized in Table 48.

Table 48: SupportedAccessCapabilities ENUM 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.

9.10.4.2 SupportedProvisioningPolicies

This collection specifies supported storage allocation policies. The enumeration literals may be used to specify space provisioning policy.

The values defined for the SupportedProvisioningPolicies ENUM are summarized in Table 49.

Table 49: SupportedProvisioningPolicies ENUM values

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

9.10.4.3 SupportedRecoveryTimeObjectives

This collection specifies supported expectations for time to access the primary store after recovery. The enumeration literals shall represent the relative time required to make a replica available as a source.

The values defined for the SupportedRecoveryTimeObjectives ENUM are summarized in Table 50.

Table 50: SupportedRecoveryTimeObjectives ENUM values, cont.

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.

9.11 DriveCollection

9.11.1 URIs

/redfish/v1/StorageServices/*{StorageServiceId}*/Drives
/redfish/v1/StorageServices/*{StorageServiceId}*/FileSystems/*{FileSystemId}*/CapacitySources/*{CapacitySourceId}*/ProvidingDrives
/redfish/v1/StorageServices/*{StorageServiceId}*/StoragePools/*{StoragePoolId}*/CapacitySources/*{CapacitySourceI*

d}/ProvidingDrives
/redfish/v1/StorageServices/*{StorageServiceId}*/Volumes/*{VolumeId}*/CapacitySources/*{CapacitySourceId}*/ProvidingDrives

9.11.2 Properties

The properties defined for the DriveCollection object are summarized in Table 51.

Table 51: DriveCollection properties

Property	Type	Notes
@odata.etag	string *read-only*	The value of this property shall be a string that is defined by the ETag HTTP header definition in RFC7232.
Description	string *read-only (null)*	This object represents the description of this resource. The resource values shall comply with the Redfish Specification-described requirements.
Members [{	array	The value of each entry of this property shall reference a Drive resource.
@odata.id	string *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.
}]		
Members@odata.nextLink	string *read-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 *read-only*	This object represents the name of this resource or array member. The resource values shall comply with the Redfish Specification-described requirements. This string value shall be of the 'Name' reserved word format.

Table 51: DriveCollection properties, cont.

Property	Type	Notes
Oem {}	object	This property shall contain the OEM extensions. All values for properties contained in this object shall conform to the Redfish Specification-described requirements. See the *Resource* schema for details on this property.

9.12 EndpointGroup 1.2.0

9.12.1 Decription

An EndpointGroup represents a collection of endpoints that are managed as a unit. By grouping together a collection of Endpoints, the EndpointGroup allows a collection of entities from differing sources or hosts to be manipulated uniformly and efficiently.

For any given EndpointGroup, all of its endpoints act exclusively as either server endpoints or client endpoints, as indicated by the value of the EndpointType property. Similarly, each Endpoint within a group has the same AccessState.

A server or client may define multiple EndpointGroup entities that access the same set of resources or functionality. A group may be designated as preferred, which signifies that access should be directed through its members in preference to the Endpoints listed in other EndpointGroups. If the value of EndpointType is Server, an EndpointGroup entity can be used to represent target port group as defined by SCSI. In that mode, the value of the TargetEndpointGroupIdentifier should correspond to the target port group number. (See clause “Device Identification VPD page” as defined in the SCSI Primary Commands specification.)

9.12.2 URIs

/redfish/v1/StorageServices/*{StorageServiceId}*/EndpointGroups/*{EndpointGroupId}*

9.12.3 Properties

The properties defined for the EndpointGroup 1.2.0 object are summarized in Table 52.

Table 52: EndpointGroup 1.2.0 properties

Property	Type	Notes
@odata.etag	string *read-only*	The value of this property shall be a string that is defined by the ETag HTTP header definition in RFC7232.
AccessState	string (enum) *read-write (null)*	Access to all associated resources through all aggregated endpoints shall share this access state. *For the possible property values, see AccessState in Property details.*
Actions *(v1.1+)* {}	object	The Actions property shall contain the available actions for this resource.
Description	string *read-only (null)*	This object represents the description of this resource. The resource values shall comply with the Redfish Specification-described requirements.
Endpoints [{	array	The value of each entry shall reference an Endpoint resource.
@odata.id	string *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.
}]		
GroupType	string (enum) *read-write (null)*	The group contains only endpoints of a given type Client/Initiator or Server/Target. If this endpoint group represents a SCSI target group, the value of GroupType shall be Server. *For the possible property values, see GroupType in Property details.*
Id	string *read-only required*	This property represents an identifier for the resource. The resource values shall comply with the Redfish Specification-described requirements.

Table 52: EndpointGroup 1.2.0 properties, cont.

Property	Type	Notes
Identifier {}	object	The value shall be unique within the managed ecosystem. See the *v1_9_1.v1_9_1* schema for details on this property.
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 Specification-described requirements. See the *Resource* schema for details on this property.
}		
Name	string *read-only required*	This object represents the name of this resource or array member. The resource values shall comply with the Redfish Specification-described requirements. This string value shall be of the 'Name' reserved word format.
Oem {}	object	This property shall contain the OEM extensions. All values for properties that this object contains shall conform to the Redfish Specification-described requirements. See the *Resource* schema for details on this property.
Preferred	boolean *read-write (null)*	A value of True in this property shall indicate that access to the associated resource through the endpoints in this endpoint group is preferred over access through other endpoints. The default value for this property is false.

Table 52: EndpointGroup 1.2.0 properties, cont.

Property	Type	Notes
TargetEndpointGroupIdentifier	integer *read-write (null)*	If this endpoint group represents a SCSI target group, the value of this property shall contain a SCSI defined identifier for this group, which corresponds to the TARGET PORT GROUP field in the REPORT TARGET PORT GROUPS response and the TARGET PORT GROUP field in an INQUIRY VPD page 85 response, type 5h identifier. See the INCITS SAM-5 specification.

9.12.4 Property details

9.12.4.1 AccessState

Access to all associated resources through all aggregated endpoints shall share this access state.

The values defined for the AccessState ENUM are summarized in Table 53.

Table 53: AccessState ENUM values, cont.

string	Description
NonOptimized	In the context of this enumeration literal, each endpoint shall be in an Active/NonOptimized state.
Optimized	In the context of this enumeration literal, each endpoint shall be in an Active/Optimized state.
Standby	In the context of this enumeration literal, each endpoint shall be in a Standby state.
Transitioning	In the context of this enumeration literal, at least one endpoint shall be transitioning to a new AccessState.
Unavailable	In the context of this enumeration literal, each endpoint shall be in an unavailable state.

9.12.4.2 GroupType

The group contains only endpoints of a given type Client/Initiator or Server/Target. If this endpoint group represents a SCSI target group, the value of GroupType shall be Server.

The values defined for the GroupType ENUM are summarized in Table 54.

Table 54: GroupType ENUM values

string	Description
Client	The group contains the client (initiator) endpoints.
Server	The group contains the server (target) endpoints.

9.13 EndpointGroupCollection

9.13.1 URIs

/redfish/v1/StorageServices/*{StorageServiceId}*/EndpointGroups

9.13.2 Properties

The properties defined for the EndpointGroupCollection object are summarized in Table 55.

Table 55: EndpointGroupCollection properties

Property	Type	Notes
@odata.etag	string *read-only*	The value of this property shall be a string that is defined by the ETag HTTP header definition in RFC7232.
Description	string *read-only* *(null)*	This object represents the description of this resource. The resource values shall comply with the Redfish Specification-described requirements.
Members [{	array	The value of each member entry shall reference an endpoint group resource.
@odata.id	string *read-only*	Link to a EndpointGroup resource. See the Links section and the *EndpointGroup* schema for details.
}]		
Members@odata.nextLink	string *read-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.

Table 55: EndpointGroupCollection properties, cont.

Property	Type	Notes
Name	string *read-only*	This object represents the name of this resource or array member. The resource values shall comply with the Redfish Specification-described requirements. This string value shall be of the 'Name' reserved word format.
Oem {}	object	This property shall contain the OEM extensions. All values for properties contained in this object shall conform to the Redfish Specification-described requirements. See the *Resource* schema for details on this property.

9.14 FeaturesRegistry 1.0.0

9.14.1 Decription

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

9.14.2 Properties

The properties defined for the FeaturesRegistry 1.0.0 object are summarized in Table 56.

Table 56: FeaturesRegistry 1.0.0 properties

Property	Type	Notes
@odata.etag	string *read-only*	The value of this property shall be a string that is defined by the ETag HTTP header definition in RFC7232.
Actions {}	object	The Actions property shall contain the available actions for this resource.
Description	string *read-only (null)*	This object represents the description of this resource. The resource values shall comply with the Redfish Specification-described requirements.

Table 56: FeaturesRegistry 1.0.0 properties, cont.

Property	Type	Notes
Features {	object * required*	The pattern property shall represent the suffix to be used in the FeatureId and shall be unique within this message registry.
(pattern) {	object	Property names follow regular expression pattern "[A-Za-z0-9]+"
CorrespondingProfileDefinition	string *read-only* *required* *(null)*	If present, the value shall define a profile definition that contains the named profile declaration.
Description	string *read-only* *required* *(null)*	The value shall be a detailed description of the feature.
FeatureName	string *read-only* *required* *(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 *read-only* *required* *(null)*	The value shall uniquely identify the version of the feature, using the major.minor.errata format.
}		

Table 56: FeaturesRegistry 1.0.0 properties, cont.

Property	Type	Notes
(pattern) {} [ ]	array, boolean, integer, number, object, string *(null)*	Property names follow regular expression pattern "^([a-zA-Z_][a-zA-Z0-9_]*)?@(odata|Redfish|Message)\.[a-zA-Z_][a-zA-Z0-9_]*$"
}		
Id	string *read-only* *required*	This property represents an identifier for the resource. The resource values shall comply with the Redfish Specification-described requirements.
Language	string *read-only* *required*	The value of this property shall be a string consisting of an RFC 5646 language code.
Name	string *read-only* *required*	This object represents the name of this resource or array member. The resource values shall comply with the Redfish Specification-described requirements. This string value shall be of the 'Name' reserved word format.
Oem {}	object	This property shall contain the OEM extensions. All values for properties that this object contains shall conform to the Redfish Specification-described requirements. See the *Resource* schema for details on this property.
OwningEntity	string *read-only* *required*	The value of this property shall be a string that represents the publisher of this registry.
RegistryPrefix	string *read-only* *required*	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.
RegistryVersion	string *read-only* *required*	The value of this property shall be the version of this message registry. The format of this string shall be of the format majorversion.minorversion.errata.

9.15 FileShare 1.1.3

9.15.1 Decription

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

9.15.2 Properties

The properties defined for the FileShare 1.1.3 object are summarized in Table 57.

Table 57: FileShare 1.1.3 properties

Property	Type	Notes
@odata.etag	string *read-only*	The value of this property shall be a string that is defined by the ETag HTTP header definition in RFC7232.
Actions *(v1.1+)* {}	object	The Actions property shall contain the available actions for this resource.
CASupported	boolean *read-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.
DefaultAccessCapabilities [ ]	array (string (enum)) *read-only* *(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 defaul access privilege. The types of default access can include Read, Write, and/or Execute. StorageAccessCapability enumeration literals may be used to describe abilities to read or write storage. *For the possible property values, see DefaultAccessCapabilities in Property details.*

Table 57: FileShare 1.1.3 properties, cont.

Property	Type	Notes
Description	string *read-only* *(null)*	This object represents the description of this resource. The resource values shall comply with the Redfish Specification-described requirements.
EthernetInterfaces {	object	The value shall be a link to an EthernetInterfaceCollection with members that provide access to the file share.
@odata.id	string *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.
}		
ExecuteSupport	boolean *read-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 *read-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.
FileShareQuotaType	string (enum) *read-write* *(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 FileShareTotalQuotaBytes property. *For the possible property values, see FileShareQuotaType in Property details.*
FileShareRemainingQuotaBytes	integer (By) *read-only* *(null)*	If present, the value of this property shall indicate the remaining number of bytes that may be consumed by this file share.

Table 57: FileShare 1.1.3 properties, cont.

Property	Type	Notes
FileShareTotalQuotaBytes	integer (By) *read-write (null)*	If present, the value of this property shall indicate the maximum number of bytes that may be consumed by this file share.
FileSharingProtocols [ ]	array (string (enum)) *read-only (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. The values shall indicate the file sharing protocols supported by the file system. At least one value shall be present. *For the possible property values, see FileSharingProtocols in Property details.*
Id	string *read-only required*	This property represents an identifier for the resource. The resource values shall comply with the Redfish Specification-described requirements.
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.
@odata.id	string *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.
}		
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 *read-only*	Link to a FileSystem resource. See the Links section and the *FileSystem* schema for details.
}		

Table 57: FileShare 1.1.3 properties, cont.

Property	Type	Notes
Oem {}	object	This property shall contain the OEM extensions. All values for properties contained in this object shall conform to the Redfish Specification-described requirements. See the *Resource* schema for details on this property.
}		
LowSpaceWarningThresholdPercents [ ]	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_SPACE_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 *read-only required*	This object represents the name of this resource or array member. The resource values shall comply with the Redfish Specification-described requirements. This string value shall be of the 'Name' reserved word format.
Oem {}	object	This property shall contain the OEM extensions. All values for properties that this object contains shall conform to the Redfish Specification-described requirements. See the *Resource* schema for details on this property.
RemainingCapacityPercent *(v1.1+)*	integer *read-only (null)*	If present, this value shall return ([(SUM(AllocatedBytes) - SUM(ConsumedBytes)]/SUM(AllocatedBytes)) * 100 represented as an integer value.

Table 57: FileShare 1.1.3 properties, cont.

Property	Type	Notes
RootAccess	boolean *read-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. See the *Resource* schema for details on this property.
WritePolicy	string (enum) *read-only (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.*

9.15.3 Property details

9.15.3.1 DefaultAccessCapabilities

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 defaul access privilege. The types of default access can include Read, Write, and/or Execute. StorageAccessCapability enumeration literals may be used to describe abilities to read or write storage.

The values defined for the DefaultAccessCapabilities ENUM are summarized in Table 58.

Table 58: DefaultAccessCapabilities ENUM 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.

9.15.3.2 FileShareQuotaType

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.

The values defined for the FileShareQuotaType ENUM are summarized in Table 59.

Table 59: FileShareQuotaType ENUM values

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

9.15.3.3 FileSharingProtocols

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. The values shall indicate the file sharing protocols supported by the file system. At least one value shall be present.

The values defined for the FileSharingProtocols ENUM are summarized in Table 60.

Table 60: FileSharingProtocols ENUM 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.

9.15.3.4 WritePolicy

The value of this property shall define how writes are replicated to the shared source.

The values defined for the WritePolicy ENUM are summarized in Table 61.

Table 61: WritePolicy ENUM 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.

9.16 FileShareCollection

9.16.1 URIs

/redfish/v1/Storage/*{StorageId}*/FileSystems/*{FileSystemsId}*/ExportedShares
/redfish/v1/StorageServices/*{StorageServiceId}*/FileSystems/*{FileSystemsId}*/ExportedShares

9.16.2 Properties

The properties defined for the FileShareCollection object are summarized in Table 62.

Table 62: FileShareCollection properties

Property	Type	Notes
@odata.etag	string *read-only*	The value of this property shall be a string that is defined by the ETag HTTP header definition in RFC7232.
Description	string *read-only* *(null)*	This object represents the description of this resource. The resource values shall comply with the Redfish Specification-described requirements.

Table 62: FileShareCollection properties, cont.

Property	Type	Notes
Members [{	array	This property shall contain references to the members of this FileSystem collection.
@odata.id	string *read-only*	Link to a FileShare resource. See the Links section and the *FileShare* schema for details.
}]		
Members@odata.nextLink	string *read-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 *read-only*	This object represents the name of this resource or array member. The resource values shall comply with the Redfish Specification-described requirements. This string value shall be of the 'Name' reserved word format.
Oem {}	object	This property shall contain the OEM extensions. All values for properties contained in this object shall conform to the Redfish Specification-described requirements. See the *Resource* schema for details on this property.

9.17 FileSystem 1.2.2

9.17.1 Decription

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

9.17.2 URIs

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

9.17.3 Properties

The properties defined for the FileSystem 1.2.2 object are summarized in Table 63.

Table 63: FileSystem 1.2.2 properties, cont

Property	Type	Notes
@odata.etag	string *read-only*	The value of this property shall be a string that is defined by the ETag HTTP header definition in RFC7232.
AccessCapabilities [ ]	array (string (enum)) *read-write (null)*	This property shall be an array containing entries for the supported IO access capabilities. Each entry shall specify a current storage access capability. StorageAccessCapability enumeration literals may be used to describe abilities to read or write storage. *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) *read-only (null)*	The value of this property shall be the block size of the file system in bytes.
Capacity {}	object	The value of this property shall be the capacity allocated to the file system in bytes. See the *CapacitySource.v1_0_0* schema for details on this property.
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-only*	Link to a CapacitySource resource. See the Links section and the *CapacitySource* schema for details.
}]		

Table 63: FileSystem 1.2.2 properties, cont

Property	Type	Notes
CasePreserved	boolean *read-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 *read-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.
CharacterCodeSet [ ]	array (string (enum)) *read-write (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. The values shall indicate the character code standards supported by the file system. *For the possible property values, see CharacterCodeSet in Property details.*
ClusterSizeBytes	integer (By) *read-write (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 *read-only (null)*	This object represents the description of this resource. The resource values shall comply with the Redfish Specification-described requirements.
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.

Table 63: FileSystem 1.2.2 properties, cont

Property	Type	Notes
@odata.id	string *read-only*	Link to Collection of *FileShare*. See the FileShare schema for details.
}		
Id	string *read-only required*	This property represents an identifier for the resource. The resource values shall comply with the Redfish Specification-described requirements.
Identifiers *(v1.1.1+)* [{ }]	array (object)	This property shall contain a list of all known durable names for this file system. This type shall contain any additional identifiers for a resource. See the *v1_9_1*v1_9_1 schema for details on this property.
ImportedShares *(v1.0.1+)* [{	array	The value shall be an array of imported file shares.
ImportedShare	 *read-write*	
}]		
IOStatistics *(v1.2+)* {}	object	The value shall represent IO statistics for this FileSystem. See the *IOStatistics* schema for details on this property.
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.
@odata.id	string *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.
}		

Table 63: FileSystem 1.2.2 properties, cont

Property	Type	Notes
Oem {}	object	This property shall contain the OEM extensions. All values for properties contained in this object shall conform to the Redfish Specification-described requirements. See the *Resource* schema for details on this property.
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 *read-only*	Link to another FileSystem resource.
}]		
ReplicaCollection@odata.count	integer *read-only*	The value of this property shall be an integer representing the number of items in a collection.
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-only*	Link to a SpareResourceSet resource. See the Links section and the *SpareResourceSet* schema for details.
}]		
SpareResourceSets@odata.count	integer *read-only*	The value of this property shall be an integer representing the number of items in a collection.
}		

Table 63: FileSystem 1.2.2 properties, cont

Property	Type	Notes
LowSpaceWarningThresholdPercents [ ]	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_SPACE_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).
MaxFileNameLengthBytes	integer (By) *read-write (null)*	If specified, this value shall specify the maximum length of a file name within the file system.
Name	string *read-only required*	This object represents the name of this resource or array member. The resource values shall comply with the Redfish Specification-described requirements. This string value shall be of the 'Name' reserved word format.
Oem {}	object	This property shall contain the OEM extensions. All values for properties that this object contains shall conform to the Redfish Specification-described requirements. See the *Resource* schema for details on this property.
RecoverableCapacitySourceCount *(v1.2+)*	integer *read-write (null)*	The value is the number of available capacity source resources currently available in the event that an equivalent capacity source resource fails.
RemainingCapacity {}	object	The value of this property shall be the remaining capacity allocated to the file system in bytes. See the *CapacitySource.v1_0_0* schema for details on this property.

Table 63: FileSystem 1.2.2 properties, cont

Property	Type	Notes
RemainingCapacityPercent *(v1.1+)*	integer *read-only* *(null)*	If present, this value shall return {[(SUM(AllocatedBytes) - SUM(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 *read-only*	Link to a ReplicaInfo resource. See the Links section and the *StorageReplicaInfo* schema for details.
}		
ReplicaTargets *(v1.2.1+)* [{	array	The value shall reference the target replicas that are sourced by this replica.
@odata.id	string *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.
}]		

9.17.4 Property details

9.17.4.1 AccessCapabilities

This property shall be an array containing entries for the supported IO access capabilities. Each entry shall specify a current storage access capability. StorageAccessCapability enumeration literals may be used to describe abilities to read or write storage.

The values defined for the AccessCapabilities ENUM are summarized in Table 64.

Table 64: AccessCapabilities ENUM 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.

9.17.4.2 CharacterCodeSet

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. The values shall indicate the character code standards supported by the file system.

The values defined for the CharacterCodeSet ENUM are summarized in Table 65.

Table 65: CharacterCodeSet ENUM 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 characer encoding is supported by the file system.
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.

9.18 FileSystemCollection

9.18.1 URIs

/redfish/v1/Storage/*{StorageId}*/FileSystems/*{FileSystemsId}*/ExportedShares/*{FileShareId}*
/redfish/v1/StorageServices/*{StorageServiceId}*/FileSystems/*{FileSystemsId}*/ExportedShares/*{FileShareId}*

9.18.2 Properties

The properties defined for the FileSystemCollection object are summarized in Table 66.

Table 66: FileSystemCollection properties

Property	Type	Notes
@odata.etag	string *read-only*	The value of this property shall be a string that is defined by the ETag HTTP header definition in RFC7232.
Description	string *read-only (null)*	This object represents the description of this resource. The resource values shall comply with the Redfish Specification-described requirements.
Members [{	array	This property shall contain references to the members of this FileSystem collection.
@odata.id	string *read-only*	Link to a FileSystem resource. See the Links section and the *FileSystem* schema for details.
}]		
Members@odata.nextLink	string *read-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 *read-only*	This object represents the name of this resource or array member. The resource values shall comply with the Redfish Specification-described requirements. This string value shall be of the 'Name' reserved word format.

Table 66: FileSystemCollection properties, cont.

Property	Type	Notes
Oem {}	object	This property shall contain the OEM extensions. All values for properties contained in this object shall conform to the Redfish Specification-described requirements. See the *Resource* schema for details on this property.

9.19 HostedStorageServices

9.19.1 URIs

/redfish/v1/Systems/*{ComputerSystemId}*/HostedServices/StorageServices

9.19.2 Properties

The properties defined for the HostedStorageServices object are summarized in Table 67.

Table 67: HostedStorageServices properties

Property	Type	Notes
@odata.etag	string *read-only*	The value of this property shall be a string that is defined by the ETag HTTP header definition in RFC7232.
Description	string *read-only* *(null)*	This object represents the description of this resource. The resource values shall comply with the Redfish Specification-described requirements.
Members [{	array	The value of each member entry shall reference a StorageService resource.
@odata.id	string *read-only*	Link to a StorageService resource. See the Links section and the *StorageService* schema for details.
}]		

Table 67: HostedStorageServices properties, cont.

Property	Type	Notes
Members@odata.nextLink	string *read-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 *read-only*	This object represents the name of this resource or array member. The resource values shall comply with the Redfish Specification-described requirements. This string value shall be of the 'Name' reserved word format.
Oem {}	object	This property shall contain the OEM extensions. All values for properties contained in this object shall conform to the Redfish Specification-described requirements. See the *Resource* schema for details on this property.

9.20 IOConnectivityLoSCapabilities 1.1.3

9.20.1 Decription

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

9.20.2 URIs

/redfish/v1/StorageServices/*{StorageServiceId}*/IOConnectivityLoSCapabilities

9.20.3 Properties

The properties defined for the IOConnectivityLoSCapabilities 1.1.3 object are summarized in Table 68.

Table 68: IOConnectivityLoSCapabilities 1.1.3 properties

Property	Type	Notes
@odata.etag	string *read-only*	The value of this property shall be a string that is defined by the ETag HTTP header definition in RFC7232.
Actions *(v1.1+)* {}	object	The Actions property shall contain the available actions for this resource.
Description	string *read-only* *(null)*	This object represents the description of this resource. The resource values shall comply with the Redfish Specification-described requirements.
Id	string *read-only* *required*	This property represents an identifier for the resource. The resource values shall comply with the Redfish Specification-described requirements.
Identifier {}	object	The value identifies this resource. The value shall be unique within the managed ecosystem. See the *v1_9_1.v1_9_1* schema for details on this property.
MaxSupportedBytesPerSecond	integer (By/s) *read-write* *(null)*	The value shall be the maximum bytes per second that a connection can support.

Table 68: IOConnectivityLoSCapabilities 1.1.3 properties, cont.

Property	Type	Notes
MaxSupportedIOPS *(v1.1+)*	integer ([IO]/s) *read-write (null)*	The value shall be the maximum IOPS that a connection can support.
Name	string *read-only required*	This object represents the name of this resource or array member. The resource values shall comply with the Redfish Specification-described requirements. This string value shall be of the 'Name' reserved word format.
Oem {}	object	This property shall contain the OEM extensions. All values for properties that this object contains shall conform to the Redfish Specification-described requirements. See the *Resource* schema for details on this property.
SupportedAccessProtocols [ ]	array (string (enum)) *read-write (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 SupportedAccessProtocols in Property details.*
SupportedLinesOfService [{	array	The collection shall contain known and supported IOConnectivityLinesOfService.
@odata.id	string *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.
}]		

9.20.4 Property details

9.20.4.1 SupportedAccessProtocols

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*'}).

The values defined for the SupportedAccessProtocols ENUM are summarized in Table 69.

Table 69: SupportedAccessProtocols ENUM values

string	Description
AHCI	This value shall indicate conformance to the Intel Advanced Host Controller Interface (AHCI) 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.
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.
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	
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.

Table 69: SupportedAccessProtocols ENUM values, cont.

string	Description
OEM	This value shall indicate conformance to an OEM-specific architecture and additional information may be included in the OEM section.
PCIe	This value shall indicate conformance to the PCI-SIG PCI Express Base Specification.
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.
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 Tranmission 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.
USB	This value shall indicate conformance to the USB Implementers Forum Universal Serial Bus Specification.

9.21 IOPerformanceLoSCapabilities 1.1.3

9.21.1 Decription

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

9.21.2 URIs

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

9.21.3 Properties

The properties defined for the IOPerformanceLoSCapabilities 1.1.3 object are summarized in Table 70.

Table 70: IOPerformanceLoSCapabilities 1.1.3 properties

Property	Type	Notes
@odata.etag	string *read-only*	The value of this property shall be a string that is defined by the ETag HTTP header definition in RFC7232.
Actions *(v1.1+)* {}	object	The Actions property shall contain the available actions for this resource.
Description	string *read-only (null)*	This object represents the description of this resource. The resource values shall comply with the Redfish Specification-described requirements.
Id	string *read-only required*	This property represents an identifier for the resource. The resource values shall comply with the Redfish Specification-described requirements.
Identifier {}	object	The value shall be unique within the managed ecosystem. See the *v1_9_1.v1_9_1* schema for details on this property.
IOLimitingIsSupported	boolean *read-write (null)*	If true, the system should limit IOPS to MaxIOOperationsPerSecondPerTerabyte * (Volume Size in Terabytes). Otherwise, the system shall not inforce a limit. The default value for this property is false.
MaxSamplePeriod	string (s) *read-write (null)*	The value shall be an ISO 8601 duration specifying the maximum sampling period over which average values are calculated.

Table 70: IOPerformanceLoSCapabilities 1.1.3 properties, cont.

Property	Type	Notes
MinSamplePeriod	string (s) *read-write (null)*	The value shall be an ISO 8601 duration specifying the minimum sampling period over which average values are calculated.
MinSupportedIoOperationLatencyMicroseconds	integer (us) *read-write (null)*	The value shall be the minimum supported average IO latency in microseconds calculated over the SamplePeriod.
Name	string *read-only required*	This object represents the name of this resource or array member. The resource values shall comply with the Redfish Specification-described requirements. This string value shall be of the 'Name' reserved word format.
Oem {}	object	This property shall contain the OEM extensions. All values for properties that this object contains shall conform to the Redfish Specification-described requirements. See the *Resource* schema for details on this property.
SupportedIOWorkloads [{	array	The value shall be a collection of supported workloads.

Table 70: IOPerformanceLoSCapabilities 1.1.3 properties, cont.

Property	Type	Notes
Components [{	array	The value shall be an array of IO workload component descriptions.
AverageIOBytes	integer (By) *read-write (null)*	The value shall be the expected average I/O size.
Duration	string (s) *read-write (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) *read-write (null)*	The enumeration literal shall be the expected access pattern. *For the possible property values, see IOAccessPattern in Property details.*
PercentOfData	integer (%) *read-write (null)*	The value shall be the expected percent of the data referenced by the workload that is covered by this component.
PercentOfIOPS	integer (%) *read-write (null)*	The value shall be the expected percent of the total IOPS for this workload that is covered by this component.
Schedule {}	object	The value shall specifies when this workload component is applied to the overall workload. See the *v1_2_1.v1_2_1* schema for details on this property.
}]		

Table 70: IOPerformanceLoSCapabilities 1.1.3 properties, cont.

Property	Type	Notes
Name	string *read-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.
}]		
SupportedLinesOfService [{	array	The value shall be a collection supported IO performance service options.
@odata.id	string *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.
}]		

9.21.4 Property details

9.21.4.1 IOAccessPattern

The enumeration literal shall be the expected access pattern.

The values defined for the IOAccessPattern ENUM are summarized in Table 71.

Table 71: IOAccessPattern ENUM 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.

9.22 LineOfService 1.0.0

9.22.1 Decription

This service option is the abstract base class for other ClassOfService and concrete lines of service.

9.22.2 Properties

The properties defined for the LineOfService 1.0.0 object are summarized in Table 72.

Table 72: LineOfService 1.0.0 properties, cont.

Property	Type	Notes
@odata.etag	string *read-only*	The value of this property shall be a string that is defined by the ETag HTTP header definition in RFC7232.
Description	string *read-only* *(null)*	This object represents the description of this resource. The resource values shall comply with the Redfish Specification-described requirements.
Id	string *read-only* *required*	This property represents an identifier for the resource. The resource values shall comply with the Redfish Specification-described requirements.
Name	string *read-only* *required*	This object represents the name of this resource or array member. The resource values shall comply with the Redfish Specification-described requirements. This string value shall be of the 'Name' reserved word format.
Oem {}	object	This property shall contain the OEM extensions. All values for properties that this object contains shall conform to the Redfish Specification-described requirements. See the *Resource* schema for details on this property.

9.23 LineOfServiceCollection

9.23.1 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}*/IOPerformanceLinesOfService /redfish/v1/StorageServices/*{StorageServiceId}*/LinesOfService/DataProtectionLinesOfService /redfish/v1/StorageServices/*{StorageServiceId}*/LinesOfService/DataSecurityLinesOfService /redfish/v1/StorageServices/*{StorageServiceId}*/LinesOfService/DataStorageLinesOfService /redfish/v1/StorageServices/*{StorageServiceId}*/LinesOfService/IOConnectivityLinesOfService /redfish/v1/StorageServices/*{StorageServiceId}*/LinesOfService/IOPerformanceLinesOfService

9.23.2 Properties

The properties defined for the LineOfServiceCollection object are summarized in Table 73.

Table 73: LineOfServiceCollection properties

Property	Type	Notes
@odata.etag	string *read-only*	The value of this property shall be a string that is defined by the ETag HTTP header definition in RFC7232.
Description	string *read-only* *(null)*	This object represents the description of this resource. The resource values shall comply with the Redfish Specification-described requirements.
Members [{	array	The value of each member entry shall reference a LineOfService resource.

Table 73: LineOfServiceCollection properties, cont.

Property	Type	Notes
@odata.id	string *read-only*	Link to a LineOfService resource. See the Links section and the *LineOfService* schema for details.
}]		
Members@odata.nextLink	string *read-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 *read-only*	This object represents the name of this resource or array member. The resource values shall comply with the Redfish Specification-described requirements. This string value shall be of the 'Name' reserved word format.
Oem {}	object	This property shall contain the OEM extensions. All values for properties contained in this object shall conform to the Redfish Specification-described requirements. See the *Resource* schema for details on this property.

9.24 SpareResourceSet 1.0.1

9.24.1 Decription

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

9.24.2 Properties

The properties defined for the SpareResourceSet 1.0.1 object are summarized in Table 74.

Table 74: SpareResourceSet 1.0.1 properties

Property	Type	Notes
@odata.etag	string *read-only*	The value of this property shall be a string that is defined by the ETag HTTP header definition in RFC7232.
Actions *(v1.0.1+)* {}	object	The Actions property shall contain the available actions for this resource.
Description	string *read-only (null)*	This object represents the description of this resource. The resource values shall comply with the Redfish Specification-described requirements.
Id	string *read-only required*	This property represents an identifier for the resource. The resource values shall comply with the Redfish Specification-described requirements.
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 Specification-described requirements. See the *Resource* schema for details on this property.
OnHandSpares [{	array	The type of resources in the set.
@odata.id	string *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.
}]		

Table 74: SpareResourceSet 1.0.1 properties, cont.

Property	Type	Notes
OnHandSpares@odata.count	integer *read-only*	The value of this property shall be an integer representing the number of items in a collection.
ReplacementSpareSets [{	array	Other spare sets that can be utilized to replenish this spare set.
@odata.id	string *read-only*	Link to another SpareResourceSet resource.
}]		
ReplacementSpareSets@odata.count	integer *read-only*	The value of this property shall be an integer representing the number of items in a collection.
}		
Name	string *read-only* *required*	This object represents the name of this resource or array member. The resource values shall comply with the Redfish Specification-described requirements. This string value shall be of the 'Name' reserved word format.
Oem {}	object	This property shall contain the OEM extensions. All values for properties that this object contains shall conform to the Redfish Specification-described requirements. See the *Resource* schema for details on this property.
OnHandLocation {}	object	The location where this set of spares is kept. See the *v1_5_0.v1_5_0* schema for details on this property.

Table 74: SpareResourceSet 1.0.1 properties, cont.

Property	Type	Notes
OnLine	boolean *read-write (null)*	This set shall be available online.
ResourceType	string *read-write (null)*	The type of resources in the set.
TimeToProvision	string *read-write (null)*	Amount of time needed to make an on-hand resource available as a spare. Pattern: -?P(D)?(T(H)?(M)?((.)?S)?)?
TimeToReplenish	string *read-write (null)*	Amount of time to needed replenish consumed on-hand resources. Pattern: -?P(D)?(T(H)?(M)?((.)?S)?)?

9.25 StorageGroup 1.2.1

9.25.1 Decription

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.

9.25.2 URIs

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

/redfish/v1/StorageServices/*{StorageServiceId}*/Volumes/*{VolumeId}*/StorageGroups/*{StorageGroupId}*

9.25.3 Properties

The properties defined for the StorageGroup 1.2.1 object are summarized in Table 75.

Table 75: StorageGroup 1.2.1 properties

Property	Type	Notes
@odata.etag	string *read-only*	The value of this property shall be a string that is defined by the ETag HTTP header definition in RFC7232.
AccessState	string (enum) *read-write (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.
#StorageGroup.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.*
#StorageGroup.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.*
}		

Table 75: StorageGroup 1.2.1 properties, cont.

Property	Type	Notes
AuthenticationMethod *(v1.2+)*	string (enum) *read-write (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.*
ChapInfo *(v1.2+)* [{	array	The value of this property must reflect the authentication used by this specific endpoint. For example, 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.
InitiatorCHAPPassword	string *read-write (null)*	The value of this property shall be the shared secret for CHAP authentication.
InitiatorCHAPUser	string *read-write (null)*	If present, this property is the initiator CHAP username for authentication. For example, with an iSCSI scenario, use the initiator iQN.
TargetCHAPUser	string *read-write (null)*	The value of this property shall be the CHAP Username for 2-way CHAP authentication. For example, with an iSCSI scenario, use the target iQN. In a FC with DHCHAP, this value will be a FC WWN.
TargetPassword	string *read-write (null)*	The value of this property shall be the CHAP Secret for 2-way CHAP authentication.

Table 75: StorageGroup 1.2.1 properties, cont.

Property	Type	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 *read-only*	Link to a EndpointGroup resource. See the Links section and the *EndpointGroup* schema for details.
}]		
Description	string *read-only* *(null)*	This object represents the description of this resource. The resource values shall comply with the Redfish Specification-described requirements.
Id	string *read-only* *required*	This property represents an identifier for the resource. The resource values shall comply with the Redfish Specification-described requirements.
Identifier {}	object	The value shall be unique within the managed ecosystem. See the *v1_9_1.v1_9_1* schema for details on this property.

Table 75: StorageGroup 1.2.1 properties, cont.

Property	Type	Notes
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-only*	Link to another StorageGroup resource.
}]		
ChildStorageGroups@odata.count	integer *read-only*	The value of this property shall be an integer representing the number of items in a collection.
ClassOfService {	object	The ClassOfService that all storage in this StorageGroup conforms to.
@odata.id	string *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.
}		
Oem {}	object	This property shall contain the OEM extensions. All values for properties contained in this object shall conform to the Redfish Specification-described requirements. See the *Resource* schema for details on this property.
ParentStorageGroups [{	array	An array of references to StorageGroups that incorporate this StorageGroup.
@odata.id	string *read-only*	Link to another StorageGroup resource.

Table 75: StorageGroup 1.2.1 properties, cont.

Property	Type	Notes
}]		
ParentStorageGroups@odata.count	integer *read-only*	The value of this property shall be an integer representing the number of items in a collection.
}		
MappedVolumes *(v1.1+)* [{	array	An array of mapped volumes managed by this storage group.
LogicalUnitNumber	string *read-write (null)*	If present, the value is a SCSI Logical Unit Number for the Volume.
Volume {	object	The value shall reference a mapped Volume. See the *Volume* schema for details on this property.
@odata.id	string *read-only*	Link to a Volume resource. See the Links section and the *Volume* schema for details.
}		
}]		
MembersAreConsistent	boolean *read-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 *read-only required*	This object represents the name of this resource or array member. The resource values shall comply with the Redfish Specification-described requirements. This string value shall be of the 'Name' reserved word format.

Table 75: StorageGroup 1.2.1 properties, cont.

Property	Type	Notes
Oem {}	object	This property shall contain the OEM extensions. All values for properties that this object contains shall conform to the Redfish Specification-described requirements. See the *Resource* schema for details on this property.
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 *read-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 *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.
}]		
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 *read-only*	Link to a EndpointGroup resource. See the Links section and the *EndpointGroup* schema for details.
}]		

Table 75: StorageGroup 1.2.1 properties, cont.

Property	Type	Notes
Status {}	object	The property shall contain the status of the StorageGroup. See the *Resource* schema for details on this property.
Volumes [{	array	An array of references to volumes managed by this storage group.
@odata.id	string *read-only*	Link to a Volume resource. See the Links section and the *Volume* schema for details.
}]		
VolumesAreExposed	boolean *read-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.

9.25.4 Actions

9.25.4.1 ExposeVolumes

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.

URIs:

/redfish/v1/Storage/*{StorageId}*/StorageGroups/*{StorageGroupId}*/Actions/StorageGroup.ExposeVolumes
/redfish/v1/Storage/*{StorageId}*/Volumes/*{VolumeId}*/StorageGroups/*{StorageGroupId}*/Actions/StorageGroup.ExposeVolumes
/redfish/v1/StorageServices/*{StorageServiceId}*/StorageGroups/*{StorageGroupId}*/Actions/StorageGroup.ExposeVolumes
/redfish/v1/StorageServices/*{StorageServiceId}*/Volumes/*{VolumeId}*/StorageGroups/*{StorageGroupId}*/Actions/StorageGroup.ExposeVolumes

Parameters

(This action takes no parameters.)

9.25.4.2 HideVolumes

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.

URIs:

/redfish/v1/Storage/*{StorageId}*/StorageGroups/*{StorageGroupId}*/Actions/StorageGroup.HideVolumes
/redfish/v1/Storage/*{StorageId}*/Volumes/*{VolumeId}*/StorageGroups/*{StorageGroupId}*/Actions/StorageGroup.HideVolumes
/redfish/v1/StorageServices/*{StorageServiceId}*/StorageGroups/*{StorageGroupId}*/Actions/StorageGroup.HideVolumes
/redfish/v1/StorageServices/*{StorageServiceId}*/Volumes/*{VolumeId}*/StorageGroups/*{StorageGroupId}*/Actions/StorageGroup.HideVolumes

Parameters

(This action takes no parameters.)

9.25.5 Property details

9.25.5.1 AccessState

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.

The values defined for the AccessState ENUM are summarized in Table 76.

Table 76: AccessState ENUM values, cont.

string	Description
NonOptimized	In the context of this enumeration literal, each endpoint shall be in an Active/NonOptimized state.
Optimized	In the context of this enumeration literal, each endpoint shall be in an Active/Optimized state.
Standby	In the context of this enumeration literal, each endpoint shall be in a Standby state.
Transitioning	In the context of this enumeration literal, at least one endpoint shall be transitioning to a new AccessState.
Unavailable	In the context of this enumeration literal, each endpoint shall be in an unavailable state.

9.25.5.2 AuthenticationMethod

The value of this property must be what kind of authentication that the endpoints in this StorageGroup understands.

The values defined for the AuthenticationMethod ENUM are summarized in Table 77.

Table 77: AuthenticationMethod ENUM values

string	Description
CHAP	iSCSI Challenge Handshake Authentication Protocol (CHAP) authentication is used.
DHCHAP	Diffie-Hellman Challenge Handshake Authentication Protocol (DHCHAP) is an authentication protocol used in Fibre Channel. DHCHAP implies that only properties 'TargetCHAPUser' and 'TargetPassword' need to be present.
MutualCHAP	iSCSI Mutual Challenge Handshake Authentication Protocol (CHAP) authentication is used.
None	No authentication is used.

9.26 StorageGroupCollection

9.26.1 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

9.26.2 Properties

The properties defined for the StorageGroupCollection object are summarized in Table 78.

Table 78: StorageGroupCollection properties, cont.

Property	Type	Notes
@odata.etag	string *read-only*	The value of this property shall be a string that is defined by the ETag HTTP header definition in RFC7232.
Description	string *read-only (null)*	This object represents the description of this resource. The resource values shall comply with the Redfish Specification-described requirements.
Members [{	array	The value of each member entry shall reference a StorageGroup resource.
@odata.id	string *read-only*	Link to a StorageGroup resource. See the Links section and the *StorageGroup* schema for details.
}]		
Members@odata.nextLink	string *read-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 *read-only*	This object represents the name of this resource or array member. The resource values shall comply with the Redfish Specification-described requirements. This string value shall be of the 'Name' reserved word format.
Oem {}	object	This property shall contain the OEM extensions. All values for properties contained in this object shall conform to the Redfish Specification-described requirements. See the *Resource* schema for details on this property.

9.27 StoragePool 1.3.1

9.27.1 Decription

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.

9.27.2 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}*

9.27.3 Properties

The properties defined for the StoragePool 1.3.1 object are summarized in Table 79.

Table 79: StoragePool 1.3.1 properties

Property	Type	Notes
@odata.etag	string *read-only*	The value of this property shall be a string that is defined by the ETag HTTP header definition in RFC7232.
Actions *(v1.3+)* {}	object	The Actions property shall contain the available actions for this resource.

Table 79: StoragePool 1.3.1 properties, cont.

Property	Type	Notes
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 *read-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 *read-only*	Link to Collection of *Volume*. See the Volume schema for details.
}		
BlockSizeBytes	integer (By) *read-only (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. See the *CapacitySource.v1_0_0* schema for details on this property.
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-only*	Link to a CapacitySource resource. See the Links section and the *CapacitySource* schema for details.
}]		

Table 79: StoragePool 1.3.1 properties, cont.

Property	Type	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-only*	Link to Collection of *LineOfService*. See the LineOfService schema for details.
}		
Compressed *(v1.3+)*	boolean *read-write (null)*	This property shall contain a boolean indicator if the StoragePool is currently utilizing compression or not.
Deduplicated *(v1.3+)*	boolean *read-write (null)*	This property shall contain a boolean indicator if the StoragePool is currently utilizing deduplication or not.
DefaultClassOfService *(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.

Table 79: StoragePool 1.3.1 properties, cont.

Property	Type	Notes
@odata.id	string *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.
}		
Description	string *read-only (null)*	This object represents the description of this resource. The resource values shall comply with the Redfish Specification-described requirements.
Encrypted *(v1.3+)*	boolean *read-write (null)*	This property shall contain a boolean indicator if the StoragePool is currently utilizing encryption or not.
Id	string *read-only required*	This property represents an identifier for the resource. The resource values shall comply with the Redfish Specification-described requirements.
Identifier {}	object	The value identifies this resource. The value shall be unique within the managed ecosystem. See the *v1_9_1.v1_9_1* schema for details on this property.
IOStatistics *(v1.2+)* {}	object	The value shall represent IO statistics for this StoragePool. See the *IOStatistics* schema for details on this property.
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.

Table 79: StoragePool 1.3.1 properties, cont.

Property	Type	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 *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.
}]		
DedicatedSpareDrives@odata.count	integer *read-only*	The value of this property shall be an integer representing the number of items in a collection.
DefaultClassOfService {	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.
@odata.id	string *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.
}		

Table 79: StoragePool 1.3.1 properties, cont.

Property	Type	Notes
Oem {}	object	This property shall contain the OEM extensions. All values for properties contained in this object shall conform to the Redfish Specification-described requirements. See the *Resource* schema for details on this property.
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-only*	Link to a SpareResourceSet resource. See the Links section and the *SpareResourceSet* schema for details.
}]		
SpareResourceSets@odata.count	integer *read-only*	The value of this property shall be an integer representing the number of items in a collection.
}		
LowSpaceWarningThresholdPercents [ ]	array (%) (integer, null) *read-write*	Each time the following value is less than one of the values in the array the LOW_SPACE_THRESHOLD_WARNING event shall be triggered: Across all CapacitySources entries, percent = (SUM(AllocatedBytes) - SUM(ConsumedBytes))/SUM(AllocatedBytes).
MaxBlockSizeBytes *(v1.4.1+)*	integer (By) *read-only (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.
Name	string *read-only required*	This object represents the name of this resource or array member. The resource values shall comply with the Redfish Specification-described requirements. This string value shall be of the 'Name' reserved word format.

Table 79: StoragePool 1.3.1 properties, cont.

Property	Type	Notes
Oem {}	object	This property shall contain the OEM extensions. All values for properties that this object contains shall conform to the Redfish Specification-described requirements. See the *Resource* schema for details on this property.
RecoverableCapacitySourceCount *(v1.2+)*	integer *read-write (null)*	The value is the number of available capacity source resources currently available in the event that an equivalent capacity source resource fails.
RemainingCapacityPercent * (v1.1+)*	integer *read-only (null)*	If present, this value shall return {[(SUM(AllocatedBytes) - SUM(ConsumedBytes)] / SUM(AllocatedBytes)} * 100 represented as an integer value.
Status {}	object	The property shall contain the status of the StoragePool. See the *Resource* schema for details on this property.
SupportedProvisioningPolicies *(v1.3+)* [ ]	array (string (enum)) *read-write (null)*	This collection shall specify all supported storage allocation policies for the Storage Pool. The enumeration literals may be used to specify space provisioning policy. *For the possible property values, see SupportedProvisioningPolicies in Property details.*
SupportedRAIDTypes *(v1.3+)* [ ]	array (string (enum)) *read-only (null)*	This collection shall contain all the RAIDType values supported by the storage pool. *For the possible property values, see SupportedRAIDTypes in Property details.*

9.27.4 Property details

9.27.4.1 SupportedProvisioningPolicies

This collection shall specify all supported storage allocation policies for the Storage Pool. The enumeration literals

may be used to specify space provisioning policy.

The values defined for the SupportedProvisioningPolicies ENUM are summarized in Table 80.

Table 80: SupportedProvisioningPolicies ENUM values

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

9.27.4.2 SupportedRAIDTypes

This collection shall contain all the RAIDType values supported by the storage pool.

The values defined for the SupportedRAIDTypes ENUM are summarized in Table 81.

Table 81: SupportedRAIDTypes ENUM values

string	Description
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.
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.

Table 81: SupportedRAIDTypes ENUM values, cont.

string	Description
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.
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.
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.

Table 81: SupportedRAIDTypes ENUM values, cont.

string	Description
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.
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.

9.28 StoragePoolCollection

9.28.1 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}*/ProvidingPools /redfish/v1/StorageServices/*{StorageServiceId}*/Volumes/*{VolumeId}*/AllocatedPools
/redfish/v1/StorageServices/*{StorageServiceId}*/Volumes/*{VolumeId}*/CapacitySources/*{CapacitySourceId}*/ProvidingPools

9.28.2 Properties

The properties defined for the StoragePoolCollection object are summarized in Table 82.

Table 82: StoragePoolCollection properties

Property	Type	Notes
@odata.etag	string *read-only*	The value of this property shall be a string that is defined by the ETag HTTP header definition in RFC7232.
Description	string *read-only (null)*	This object represents the description of this resource. The resource values shall comply with the Redfish Specification-described requirements.
Members [{	array	The value of each member entry shall reference a StoragePool resource.
@odata.id	string *read-only*	Link to a StoragePool resource. See the Links section and the *StoragePool* schema for details.
}]		
Members@odata.nextLink	string *read-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 *read-only*	This object represents the name of this resource or array member. The resource values shall comply with the Redfish Specification-described requirements. This string value shall be of the 'Name' reserved word format.
Oem {}	object	This property shall contain the OEM extensions. All values for properties contained in this object shall conform to the Redfish Specification-described requirements. See the *Resource* schema for details on this property.

9.29 StorageReplicaInfo 1.3.0

9.29.1 Decription

This entity shall define the characteristics of a replica.

9.29.2 Properties

The properties defined for the StorageReplicaInfo 1.3.0 object are summarized in Table 83.

Table 83: StorageReplicaInfo 1.3.0 properties

Property	Type	Notes
@odata.etag	string *read-only*	The value of this property shall be a string that is defined by the ETag HTTP header definition in RFC7232.
Actions *(v1.2+)* {}	object	The Actions property shall contain the available actions for this resource.
Description	string *read-only (null)*	This object represents the description of this resource. The resource values shall comply with the Redfish Specification-described requirements.
Id	string *read-only required*	This property represents an identifier for the resource. The resource values shall comply with the Redfish Specification-described requirements.
Name	string *read-only required*	This object represents the name of this resource or array member. The resource values shall comply with the Redfish Specification-described requirements. This string value shall be of the 'Name' reserved word format.
Oem {}	object	This property shall contain the OEM extensions. All values for properties that this object contains shall conform to the Redfish Specification-described requirements. See the *Resource* schema for details on this property.

9.30 StorageService 1.4.0

9.30.1 Decription

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.

9.30.2 URIs

/redfish/v1/StorageServices/*{StorageServiceId}*

9.30.3 Properties

The properties defined for the StorageService 1.4.0 object are summarized in Table 84.

Table 84: StorageService 1.4.0 properties

Property	Type	Notes
@odata.etag	string *read-only*	The value of this property shall be a string that is defined by the ETag HTTP header definition in RFC7232.
Actions {	object	The Actions property shall contain the available actions for this resource.
#StorageService.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-only*	Link to Collection of *LineOfService*. See the LineOfService schema for details.
}		

Table 84: StorageService 1.4.0 properties, cont.

Property	Type	Notes
ClientEndpointGroups {	object	The value of each entry in the array shall reference an EndpointGroup. Contains a link to a resource.
@odata.id	string *read-only*	Link to Collection of *EndpointGroup*. See the EndpointGroup schema for details.
}		
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-only*	Link to Collection of *ConsistencyGroup*. See the ConsistencyGroup schema for details.
}		
DataProtectionLoSCapabilities *(v1.2+)* {	object	The value shall reference the data protection capabilities of this service. See the *DataProtectionLoSCapabilities* schema for details on this property.
@odata.id	string *read-only*	Link to a DataProtectionLoSCapabilities resource. See the Links section and the *DataProtectionLoSCapabilities* schema for details.
}		
DataSecurityLoSCapabilities *(v1.2+)* {	object	The value shall reference the data security capabilities of this service. See the *DataSecurityLoSCapabilities* schema for details on this property.
@odata.id	string *read-only*	Link to a DataSecurityLoSCapabilities resource. See the Links section and the *DataSecurityLoSCapabilities* schema for details.
}		

Table 84: StorageService 1.4.0 properties, cont.

Property	Type	Notes
DataStorageLoSCapabilities *(v1.2+)* {	object	The value shall reference the data storage capabilities of this service. See the *DataStorageLoSCapabilities* schema for details on this property.
@odata.id	string *read-only*	Link to a DataStorageLoSCapabilities resource. See the Links section and the *DataStorageLoSCapabilities* schema for details.
}		
DefaultClassOfService *(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.
@odata.id	string *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.
}		
Description	string *read-only* *(null)*	This object represents the description of this resource. The resource values shall comply with the Redfish Specification-described requirements.
Drives {	object	A collection that indicates all the drives managed by this storage service. Contains a link to a resource.
@odata.id	string *read-only*	Link to Collection of *Drive*. See the Drive schema for details.
}		
EndpointGroups {	object	The value of each entry in the array shall reference an EndpointGroup. Contains a link to a resource.
}		

Table 84: StorageService 1.4.0 properties, cont.

Property	Type	Notes
@odata.id	string *read-only*	Link to Collection of *EndpointGroup*. See the EndpointGroup schema for details.
Endpoints {	object	The value of each entry in the array shall reference an Endpoint managed by this service.
@odata.id	string *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.
}		
FileSystems {	object	An array of references to FileSystems managed by this storage service. Contains a link to a resource.
@odata.id	string *read-only*	Link to Collection of *FileSystem*. See the FileSystem schema for details.
}		
Id	string *read-only* *required*	This property represents an identifier for the resource. The resource values shall comply with the Redfish Specification-described requirements.
Identifier {}	object	The value identifies this resource. The value shall be unique within the managed ecosystem. See the *v1_9_1.v1_9_1* schema for details on this property.
IOConnectivityLoSCapabilities *(v1.2+)* {	object	The value shall reference the IO connectivity capabilities of this service. See the *IOConnectivityLoSCapabilities* schema for details on this property.
@odata.id	string *read-only*	Link to a IOConnectivityLoSCapabilities resource. See the Links section and the *IOConnectivityLoSCapabilities* schema for details.

Table 84: StorageService 1.4.0 properties, cont.

Property	Type	Notes
}		
IOPerformanceLoSCapabilities *(v1.2+)* {	object	The value shall reference the IO performance capabilities of this service. See the *IOPerformanceLoSCapabilities* schema for details on this property.
@odata.id	string *read-only*	Link to a IOPerformanceLoSCapabilities resource. See the Links section and the *IOPerformanceLoSCapabilities* schema for details.
}		
IOStatistics *(v1.2+)* {}	object	The value shall represent IO statistics for this StorageService. See the *IOStatistics* schema for details on this property.
LinesOfService *(v1.4+)* [{	array	The value of each entry shall reference a LineOfService collection defined for this service.
@odata.id	string *read-only*	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.
DataProtectionLoSCapabilities {	object	The value shall reference the data protection capabilities of this service. See the *DataProtectionLoSCapabilities* schema for details on this property.
@odata.id	string *read-only*	Link to a DataProtectionLoSCapabilities resource. See the Links section and the *DataProtectionLoSCapabilities* schema for details.
}		

Table 84: StorageService 1.4.0 properties, cont.

Property	Type	Notes
DataSecurityLoSCapabilities {	object	The value shall reference the data security capabilities of this service. See the *DataSecurityLoSCapabilities* schema for details on this property.
@odata.id	string *read-only*	Link to a DataSecurityLoSCapabilities resource. See the Links section and the *DataSecurityLoSCapabilities* schema for details.
}		
DataStorageLoSCapabilities {	object	The value shall reference the data storage capabilities of this service. See the *DataStorageLoSCapabilities* schema for details on this property.
@odata.id	string *read-only*	Link to a DataStorageLoSCapabilities resource. See the Links section and the *DataStorageLoSCapabilities* schema for details.
}		
DefaultClassOfService {	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.
@odata.id	string *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.
}		
HostingSystem	 *read-write*	The value shall reference the ComputerSystem or StorageController that hosts this service.
IOConnectivityLoSCapabilities {	object	The value shall reference the IO connectivity capabilities of this service. See the *IOConnectivityLoSCapabilities* schema for details on this property.

Table 84: StorageService 1.4.0 properties, cont.

Property	Type	Notes
@odata.id	string *read-only*	Link to a IOConnectivityLoSCapabilities resource. See the Links section and the *IOConnectivityLoSCapabilities* schema for details.
}		
IOPerformanceLoSCapabilities {	object	The value shall reference the IO performance capabilities of this service. See the *IOPerformanceLoSCapabilities* schema for details on this property.
@odata.id	string *read-only*	Link to a IOPerformanceLoSCapabilities resource. See the Links section and the *IOPerformanceLoSCapabilities* 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 Specification-described requirements. See the *Resource* schema for details on this property.
}		
Name	string *read-only* *required*	This object represents the name of this resource or array member. The resource values shall comply with the Redfish Specification-described requirements. This string value shall be of the 'Name' reserved word format.
Oem {}	object	This property shall contain the OEM extensions. All values for properties that this object contains shall conform to the Redfish Specification-described requirements. See the *Resource* schema for details on this property.

Table 84: StorageService 1.4.0 properties, cont.

Property	Type	Notes