



Swordfish Scalable Storage Management API User's Guide

Version 1.0.7

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

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

Technical Position

Last Updated 23 December 2018

USAGE

Copyright © 2018 SNIA. All rights reserved. All other trademarks or registered trademarks are the property of their respective owners.

The SNIA hereby grants permission for individuals to use this document for personal use only, and for corporations and other business entities to use this document for internal use only (including internal copying, distribution, and display) provided that:

1. Any text, diagram, chart, table or definition reproduced must be reproduced in its entirety with no alteration, and,
2. Any document, printed or electronic, in which material from this document (or any portion hereof) is reproduced must acknowledge the SNIA copyright on that material, and must credit the SNIA for granting permission for its reuse.

Other than as explicitly provided above, you may not make any commercial use of this document, or any portion thereof, or distribute this document to third parties. All rights not explicitly granted are expressly reserved to SNIA.

Permission to use this document for purposes other than those enumerated above may be requested by emailing tcmd@snia.org. Please include the identity of the requesting individual and/or company and a brief description of the purpose, nature, and scope of the requested use.

All code fragments, scripts, data tables, and sample code in this SNIA document are made available under the following license:

BSD 3-Clause Software License

Copyright © 2018, The Storage Networking Industry Association

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of The Storage Networking Industry Association (SNIA) nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

DISCLAIMER

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

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

Copyright © 2016-2018 Storage Networking Industry Association.

Revision History

Table 1: Revision history

Date	Revision	Notes
19 September 2016	1.0.0	Initial Release
12 October 2016	1.0.1	General clean up and formatting consistency A discussion of unused CoS and LoS entries in ServiceCatalog Improve purpose for many use cases
1 November 2016	1.0.2	Corrected XREF link formatting
24 January 2017	1.0.3	Additional use cases and new document section addressing client considerations
25 April 2017	1.0.4	Update cross-references.
3 October 2017	1.0.5	Minor updates and corrections
13 February 2018	1.0.6	Added on-demand replication use cases
12 October 2018	1.0.7	Editorial cleanup of JSON

Contact SNIA

SNIA Web Site

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

FEEDBACK AND INTERPRETATIONS

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

INTENDED AUDIENCE

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

VERSIONING POLICY

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

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

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

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

About SNIA

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

Acknowledgements

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

Table 2: Contributors

Member	Representatives
Broadcom, Inc.	Richelle Ahlvers
Dell Inc.	Patrick Boyd George Ericson Michael Raineri Rich Roscoe
Hitachi Data Systems	Eric Hibbard
Hewlett Packard Enterprise	Jeff Hilland Chris Lionetti John Mendonca Doug Voigt
Inova Development Inc.	Karl Schopmeyer
Intel Corporation	Slawek Putyrski Paul von Behren
Microsemi	Anand Nagarjan
Microsoft Corporation	Hector Linares Jim Pinkerton Michael Pizzo Scott Seligman
NetApp, Inc.	Don Deel Nilesh Maheshwari
ScienceLogic	Patrick Strick
VMware, Inc.	Murali Rajagopal

Table of Contents

USAGE	2
DISCLAIMER	3
Revision History	3
Contact SNIA	3
FEEDBACK AND INTERPRETATIONS	4
INTENDED AUDIENCE	4
VERSIONING POLICY	4
About SNIA	5
Acknowledgements	5
Table of Contents	7
1 Introduction	9
1.1 Audience	9
1.2 Documentation structure	9
1.3 Configuration assumptions	9
1.4 Knowledge assumptions	10
1.5 Related documents	10
2 General query syntax	10
2.1 OData conformance	10
2.2 Query method	10
2.3 Query Headers	11
2.4 Service root	12
2.5 Resource path	13
2.6 Query options	13
2.7 Filter expressions	13
2.8 HTTP status codes	14
3 Introduction	15
3.1 Audience	15
3.2 Documentation structure	15
3.3 Configuration assumptions	16
3.4 Knowledge assumptions	16
3.5 Related documents	16
4 General query syntax	16
4.1 OData conformance	16
4.2 Query method	17
4.3 Query Headers	17
4.4 Service root	19
4.5 Resource path	19
4.6 Query options	19
4.7 Filter expressions	20
4.8 HTTP status codes	20
5 Actors	20
5.1 Actors	20
5.2 CloudAdmin	21
5.3 DevOps	22
5.4 Storage Admin	22
6 Management Domains	23
6.1 Management Domain Overview	23
6.2 Application storage management domain	24
6.3 Block storage management domain	24
6.4 Service catalog management domain	25
6.5 File system storage management domain	27
7 User Guidance	27
7.1 Default Class of Service	27
8 Use Cases	28
8.1 Add Volume	28

8.2 Allocate VM space	29
8.3 Allocate Volume	30
8.4 Allocate Volume using Default Class of Service	32
8.5 Health updates	35
8.6 Create class of service	36
8.7 Create file share	38
8.8 Create file system	39
8.9 Create line of service	40
8.10 Create StorageGroup	42
8.11 Create storage pool	43
8.12 Discover Class of Service	45
8.13 Expand capacity of a storage volume	47
8.14 Find storage service	51
8.15 Get capacity by class of service	52
8.16 List supported line of service options	54
8.17 Protect Volumes of an application	61
8.18 Subscribe to Threshold Events	63
8.19 Set Capacity Usage Threshold	64

1 Introduction

1.1 Audience

This guide is intended to provide a common repository of best practices, common tasks and education for the users of the Swordfish API. Each use case includes an indication of the classes of API users who are most likely to find the case useful.

1.2 Documentation structure

This document begins with a set of information intended to provide a solid foundation for readers new to restful APIs in general and Swordfish in particular. While this material is no replacement for a thorough understanding of the Swordfish specification and the material that it references, it is intended as a stand alone document that can provide a solid introduction to Swordfish.

Based on that foundational material, this document then presents a set of Use Cases intended to capture common tasks and best practices that can be used to exercise the breadth and strength of the Swordfish API. In general, the guide is structured to provide more basic use cases first, and examine common refinements and options at the same time. More advanced tasks are handled later in the guide, and assume the prior skills and assumptions have been mastered.

For each use case, this guide will use a common template. [Table 3](#) lists each field of the template and its description.

Table 3: Guidelines for the Use Case Template

Name	Description
Title	A description of the high-level scope of the Use Case
Summary	A high-level summary of the use case
Purpose	The intended goals or motivations for the use case
Who	The <u>Actor(s)</u> who are likely to use this use case. The actor description also includes a list of other use cases of interest for a named actor
Management Domain	The <u>Management Domain(s)</u> applicable to this use case. The domain description also includes a list of other use cases of interest for the named management domains
Triggers	A description of likely business conditions or goals that would make this use case useful
Detailed Context	A detailed description of the operations environment and configuration assumptions for this use case
Preconditions	Pre-existing knowledge, configurations or capabilities
Inputs	A set of parameters and values that are used to adapt a generic use case to a specific business needs. Where appropriate, the parameter description will include a data type (e.g., {CAPACITY}: desired storage capacity (int64))
Basic Course of Events	A sequence of API requests, including required headers, the body of the request, and the expected reply
Configuration Impacts	Changes to the storage configurations caused by the use case
Failure Scenario	Common failure conditions encountered in this use case
See Also	Other Use Cases that may be of interest

1.3 Configuration assumptions

This document assumes that some fundamental configuration issues have been properly addressed, and will not need to be addressed in any detail. In particular, this document assumes:

- An appropriate security infrastructure (e.g., TLS 1.2)
- A functional Swordfish/Redfish installation, in either a standalone, aggregator, or distributed configuration
- Any required login credentials

1.4 Knowledge assumptions

The Swordfish API conforms to the standards defined in the [Redfish API \(http://redfish.dmtf.org\)](http://redfish.dmtf.org). More generally, it provides a RESTful interface. The reader is assumed to be familiar with common conventions for RESTful APIs. Those readers who are interested in additional background information are encouraged to refer to the following sources:

- For RESTful APIs: [Wikipedia \(https://en.wikipedia.org/wiki/Representational_state_transfer\)](https://en.wikipedia.org/wiki/Representational_state_transfer)
- For Redfish standards: [Redfish Specification \(http://redfish.dmtf.org\)](http://redfish.dmtf.org)
- For HTTP standards: [Wikipedia \(https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol\)](https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol)

1.5 Related documents

This User's Guide is part of the documentation suite for the Swordfish API. Readers are encouraged to refer to the following for additional information: - [Swordfish API Specification \(http://www.snia.org/swordfish\)](http://www.snia.org/swordfish) - [Swordfish Tutorials \(http://www.snia.org/swordfish\)](http://www.snia.org/swordfish)

2 General query syntax

2.1 OData conformance

Swordfish, like RedFish, conforms to the [OData Version 4 \(http://docs.oasis-open.org/odata/odata/v4.0/\)](http://docs.oasis-open.org/odata/odata/v4.0/). It conforms to the query syntax and URL structure that that standard defines. The high-level assumptions are summarized here. Readers who need more information should refer to the OData standard.

All Swordfish URLs follow a general form of:

- [QueryMethod ServiceRoot/ResourcePath/MetaData](#)], or
- [QueryMethod ServiceRoot/Resource Path\[?QueryParameters\]](#)

2.2 Query method

Swordfish queries support four query methods. Each query URL must include exactly one of the query methods listed in [Table 4](#).

Table 4: Query methods

Method	Action
GET	Retrieve the current state or settings of the named Resource Path as seen through the Service Root
POST	Create a new object under the named Resource Path
PUT	Replace the object referenced by the named Resource Path
DELETE	Delete the object referenced by the named Resource Path
PATCH	Update the object referenced by the named Resource Path

Method	Action
HEAD	Validates a GET request against the named <u>Resource Path</u> without returning the HTML headers for the response without the result of the query

2.3 Query Headers

All HTTP requests and responses in a compliant Swordfish implementation support the HTTP headers required by the Redfish Protocol Specification (http://redfish.dmtf.org/schemas/DSPo266_1.0.html). The supported headers are reproduced here for convenience.

2.3.1 Request headers

Table 5: Request headers

Header	Supported Values	Notes
Accept	<u>RFC 7231</u> (http://redfish.dmtf.org/schemas/DSPo266_1.0.html#RFC7231)	Indicates to the server what media type(s) this client is prepared to accept. Services shall support requests for resources with an Accept header including application/json or application/json;charset=utf-8. Services shall support requests for metadata with an Accept header including application/xml or application/xml;charset=utf-8.
Content-Type	<u>RFC 7231</u> (http://redfish.dmtf.org/schemas/DSPo266_1.0.html#RFC7231)	Describes the type of representation used in the message body. Content-Type shall be required in requests that include a request body. Services shall accept Content-Type values of application/json or application/json;charset=utf-8.
OData-Version	4.0	Services shall reject requests which specify an unsupported OData version. If a service encounters a version that it does not support, the service should reject the request with status code <u>412</u> . If client does not specify an Odata-Version header, the client is outside the boundaries of this specification.
Authorization	<u>RFC 7235</u> (http://redfish.dmtf.org/schemas/DSPo266_1.0.html#RFC7235), Section 4.2	Required for Basic Authentication
User-Agent	<u>RFC 7231</u> (http://redfish.dmtf.org/schemas/DSPo266_1.0.html#RFC7231)	Required for tracing product tokens and their version. Multiple product tokens may be listed.
Host	<u>RFC 7230</u> (http://redfish.dmtf.org/schemas/DSPo266_1.0.html#RFC7230)	Required to allow support of multiple origin hosts at a single IP address.
Origin	<u>W3C CORS, Section 5.7</u> (http://redfish.dmtf.org/schemas/DSPo266_1.0.html#W3C-CORS)	Used to allow web applications to consume Redfish Service while preventing CSRF attacks.
If-Match	<u>RFC 7232</u> (http://redfish.dmtf.org/schemas/DSPo266_1.0.html#RFC7232)	If-Match shall be supported on PUT and PATCH requests for resources for which the service returns ETags, to ensure clients are updating the resource from a known state.

Header	Supported Values	Notes
X-Auth-Token	Opaque encoded octet strings	Used for authentication of user sessions. The token value shall be indistinguishable from random.

2.3.2 Response headers

Table 6: Response headers

Header	Supported Values	Notes
OData-Version	4.0	Describes the OData version of the payload that the response conforms to.
Content-Type	RFC 7231 (http://redfish.dmtf.org/schemas/DSP0266_1.0.html#RFC7231)	Describes the type of representation used in the message body. Services shall specify a Content-Type of application/json when returning resources as JSON, and application/xml when returning metadata as XML. ;charset=utf-8 shall be appended to the Content-Type if specified in the chosen media-type in the Accept header for the request.
ETag	RFC 7232 (http://redfish.dmtf.org/schemas/DSP0266_1.0.html#RFC7232)	An identifier for a specific version of a resource, often a message digest. Etags shall be included on responses to GETs of ManagerAccount objects.
Server	RFC 7231 (http://redfish.dmtf.org/schemas/DSP0266_1.0.html#RFC7231)	Required to describe a product token and its version. Multiple product tokens may be listed.
Link		Link headers shall be returned as described in the clause on Link Headers (http://redfish.dmtf.org/schemas/DSP0266_1.0.html#link-header)
Location	RFC 7231 (http://redfish.dmtf.org/schemas/DSP0266_1.0.html#RFC7231)	Indicates a URI that can be used to request a representation of the resource. Shall be returned if a new resource was created. Location and X-Auth-Token shall be included on responses which create user sessions.
Cache-Control	RFC 7234 (http://redfish.dmtf.org/schemas/DSP0266_1.0.html#RFC7234)	This header shall be supported and is meant to indicate whether a response can be cached or not.
Access-Control-Allow-Origin	W3C CORS, Section 5.7 (http://redfish.dmtf.org/schemas/DSP0266_1.0.html#W3C-CORS)	Prevents or allows requests based on originating domain. Used to prevent CSRF attacks.
Allow	POST, PUT, PATCH, DELETE, GET, HEAD	Shall be returned with a 405 (Method Not Allowed) response to indicate the valid methods for the specified Request URI. Should be returned with any GET or HEAD operation to indicate the other allowable operations for this resource.
WWW-Authenticate	RFC 7234 (http://redfish.dmtf.org/schemas/DSP0266_1.0.html#RFC7234), Section 4.1	Required for Basic and other optional authentication mechanisms. See the Security (http://redfish.dmtf.org/schemas/DSP0266_1.0.html#security) clause for details.
X-Auth-Token	Opaque encoded octet strings	Used for authentication of user sessions. The token value shall be indistinguishable from random.

2.4 Service root

This is the base of all Swordfish URL's. A GET request to the Service Root will return an overview of the services provided by a given Swordfish service. In addition, the Service Root will include versioning information.

All Service Root URLs that are compliant with the Swordfish specification will be of the form `https://hostName/redfish/v1`, where **hostName** specifies the system (and optionally port number), of the Swordfish service provider.

2.5 Resource path

The Resource Path identifies the specific object (or collection of objects) that is the target of the Swordfish query. As with OData, Swordfish Resource Paths can identify: - A singleton object (e.g., a specific storage LUN or Volume) - A collection of objects (e.g., the list of all LUNs provided by a specific storage array)

At the highest level, Swordfish identifies two major collections: Storage Systems and Storage Services.

2.6 Query options

Swordfish queries can include arbitrary sets of query options to further refine the target of given query or the actions being requested of that target. These general query options are summarized in [Table 7: Query Options](#).

Note: Additional query options may be supported (or constrained) for a specific query target or resource path. These target-specific query options will be addressed in specific use case descriptions, as required.

Table 7: Query Options

Parameter Name	Arguments	Notes
<code>\$skip=<i>n</i></code>	Integer	Omit the first <i>n</i> entries in the collection from the returned set of objects (required by redfish (redfish.dmtf.org))
<code>\$top=<i>n</i></code>	Integer	Return, at most, the first <i>n</i> entries in the returned set of objects (required by redfish (redfish.dmtf.org))
<code>\$filter=<i>condition</i></code>	Filter Expression (filter expressions)	Returns only the members of the named collection that match the provided logical expression (required by swordfish (snia.org/swordfish))
<code>\$expand=<i>target</i></code>	Expand Expression (http://docs.oasis-open.org/odata/odata/v4.0/errata03/os/complete/part2-url-conventions/odata-v4.0-errata03-os-part2-url-conventions-complete.html#_Toc453752359)	Expand additional detail on the target property(s) in the returned result set (required by swordfish (snia.org/swordfish))
<code>\$select=<i>property list</i></code>	Comma-separated list of object properties	Return the named properties for each object in the result set, rather than the entire object (required by swordfish (snia.org/swordfish))
<code>\$orderby=<i>filter condition</i></code>	Filter Expression (filter expressions)	sort the result set by the output values from the filter expression (required by swordfish (snia.org/swordfish))

2.7 Filter expressions

1. Simple example `$filter=(age gt 30)` A group of people never to trust.

For more information see [Filter Expression \(http://docs.oasis-open.org/odata/odata/v4.0/errata03/os/complete/part2-url-conventions/odata-v4.0-errata03-os-part2-url-conventions-complete.html#_Toc453752356\)](http://docs.oasis-open.org/odata/odata/v4.0/errata03/os/complete/part2-url-conventions/odata-v4.0-errata03-os-part2-url-conventions-complete.html#_Toc453752356) in the OData specification.

2.8 HTTP status codes

Swordfish clients may receive any of the standard HTTP status codes. For status codes greater than 400, the server can also return extended status information as a simple JSON object (see [Redfish Specification \(http://redfish.dmtf.org\)](http://redfish.dmtf.org) for more information).

3 Introduction

3.1 Audience

This guide is intended to provide a common repository of best practices, common tasks and education for the users of the Swordfish API. Each use case includes an indication of the classes of API users who are most likely to find the case useful.

3.2 Documentation structure

This document begins with a set of information intended to provide a solid foundation for readers new to restful APIs in general and Swordfish in particular. While this material is no replacement for a thorough understanding of the Swordfish specification and the material that it references, it is intended as a stand alone document that can provide a solid introduction to Swordfish.

Based on that foundational material, this document then presents a set of Use Cases intended to capture common tasks and best practices that can be used to exercise the breadth and strength of the Swordfish API. In general, the guide is structured to provide more basic use cases first, and examine common refinements and options at the same time. More advanced tasks are handled later in the guide, and assume the prior skills and assumptions have been mastered.

For each use case, this guide will use a common template. [Table 3](#) lists each field of the template and its description.

Table 3: Guidelines for the Use Case Template

Name	Description
Title	A description of the high-level scope of the Use Case
Summary	A high-level summary of the use case
Purpose	The intended goals or motivations for the use case
Who	The Actor(s) who are likely to use this use case. The actor description also includes a list of other use cases of interest for a named actor
Management Domain	The Management Domain(s) applicable to this use case. The domain description also includes a list of other use cases of interest for the named management domains
Triggers	A description of likely business conditions or goals that would make this use case useful
Detailed Context	A detailed description of the operations environment and configuration assumptions for this use case
Preconditions	Pre-existing knowledge, configurations or capabilities
Inputs	A set of parameters and values that are used to adapt a generic use case to a specific business needs. Where appropriate, the parameter description will include a data type (e.g., {CAPACITY}: desired storage capacity (int64))
Basic Course of Events	A sequence of API requests, including required headers, the body of the request, and the expected reply

Name	Description
Configuration Impacts	Changes to the storage configurations caused by the use case
Failure Scenario	Common failure conditions encountered in this use case
See Also	Other Use Cases that may be of interest

3.3 Configuration assumptions

This document assumes that some fundamental configuration issues have been properly addressed, and will not need to be addressed in any detail. In particular, this document assumes:

- An appropriate security infrastructure (e.g., TLS 1.2)
- A functional Swordfish/Redfish installation, in either a standalone, aggregator, or distributed configuration
- Any required login credentials

3.4 Knowledge assumptions

The Swordfish API conforms to the standards defined in the [Redfish API](#). More generally, it provides a RESTful interface. The reader is assumed to be familiar with common conventions for RESTful APIs. Those readers who are interested in additional background information are encouraged to refer to the following sources:

- For RESTful APIs: [Wikipedia](#)
- For Redfish standards: [Redfish Specification](#)
- For HTTP standards: [Wikipedia](#)

3.5 Related documents

This User's Guide is part of the documentation suite for the Swordfish API. Readers are encouraged to refer to the following for additional information: - [Swordfish API Specification](#) - [Swordfish Tutorials](#)

4 General query syntax

4.1 OData conformance

Swordfish, like RedFish, conforms to the [OData Version 4](#). It conforms to the query syntax and URL structure that that standard defines. The high-level assumptions are summarized here. Readers who need more information should refer to the OData standard.

All Swordfish URLs follow a general form of:

- [QueryMethod ServiceRoot/ResourcePath/MetaData](#)], or
- [QueryMethod ServiceRoot/Resource Path\[?QueryParameters\]](#)

4.2 Query method

Swordfish queries support four query methods. Each query URL must include exactly one of the query methods listed in [Table 4](#).

Table 4: Query methods

Method	Action
GET	Retrieve the current state or settings of the named Resource Path as seen through the Service Root
POST	Create a new object under the named Resource Path
PUT	Replace the object referenced by the named Resource Path
DELETE	Delete the object referenced by the named Resource Path
PATCH	Update the object referenced by the named Resource Path
HEAD	Validates a GET request against the named Resource Path without returning the HTML headers for the response without the result of the query

4.3 Query Headers

All HTTP requests and responses in a compliant Swordfish implementation support the HTTP headers required by the [Redfish Protocol Specification](#). The supported headers are reproduced here for convenience.

4.3.1 Request headers

Table 5: Request headers

Header	Supported Values	Notes
Accept	RFC 7231	Indicates to the server what media type(s) this client is prepared to accept. Services shall support requests for resources with an Accept header including application/json or application/json;charset=utf-8. Services shall support requests for metadata with an Accept header including application/xml or application/xml;charset=utf-8.
Content-Type	RFC 7231	Describes the type of representation used in the message body. Content-Type shall be required in requests that include a request body. Services shall accept Content-Type values of application/json or application/json;charset=utf-8.

Header	Supported Values	Notes
OData-Version	4.0	Services shall reject requests which specify an unsupported OData version. If a service encounters a version that it does not support, the service should reject the request with status code 412 . If client does not specify an Odata-Version header, the client is outside the boundaries of this specification.
Authorization	RFC 7235 , Section 4.2	Required for Basic Authentication
User-Agent	RFC 7231	Required for tracing product tokens and their version. Multiple product tokens may be listed.
Host	RFC 7230	Required to allow support of multiple origin hosts at a single IP address.
Origin	W3C CORS , Section 5.7	Used to allow web applications to consume Redfish Service while preventing CSRF attacks.
If-Match	RFC 7232	If-Match shall be supported on PUT and PATCH requests for resources for which the service returns ETags, to ensure clients are updating the resource from a known state.
X-Auth-Token	Opaque encoded octet strings	Used for authentication of user sessions. The token value shall be indistinguishable from random.

4.3.2 Response headers

Table 6: Response headers

Header	Supported Values	Notes
OData-Version	4.0	Describes the OData version of the payload that the response conforms to.
Content-Type	RFC 7231	Describes the type of representation used in the message body. Services shall specify a Content-Type of application/json when returning resources as JSON, and application/xml when returning metadata as XML. ;charset=utf-8 shall be appended to the Content-Type if specified in the chosen media-type in the Accept header for the request.
ETag	RFC 7232	An identifier for a specific version of a resource, often a message digest. Etags shall be included on responses to GETs of ManagerAccount objects.
Server	RFC 7231	Required to describe a product token and its version. Multiple product tokens may be listed.
Link		Link headers shall be returned as described in the clause on Link Headers
Location	RFC 7231	Indicates a URI that can be used to request a representation of the resource. Shall be returned if a new resource was created. Location and X-Auth-Token shall be included on responses which create user sessions.
Cache-Control	RFC 7234	This header shall be supported and is meant to indicate whether a response can be cached or not.

Header	Supported Values	Notes
Access-Control-Allow-Origin	W3C CORS, Section 5.7	Prevents or allows requests based on originating domain. Used to prevent CSRF attacks.
Allow	POST, PUT, PATCH, DELETE, GET, HEAD	Shall be returned with a 405 (Method Not Allowed) response to indicate the valid methods for the specified Request URI. Should be returned with any GET or HEAD operation to indicate the other allowable operations for this resource.
WWW-Authenticate	RFC 7234, Section 4.1	Required for Basic and other optional authentication mechanisms. See the Security clause for details.
X-Auth-Token	Opaque encoded octet strings	Used for authentication of user sessions. The token value shall be indistinguishable from random.

4.4 Service root

This is the base of all Swordfish URL's. A GET request to the Service Root will return an overview of the services provided by a given Swordfish service. In addition, the Service Root will include versioning information.

All Service Root URLs that are compliant with the Swordfish specification will be of the form `https://hostName/redfish/v1`, where **hostName** specifies the system (and optionally port number), of the Swordfish service provider.

4.5 Resource path

The Resource Path identifies the specific object (or collection of objects) that is the target of the Swordfish query. As with OData, Swordfish Resource Paths can identify: - A singleton object (e.g., a specific storage LUN or Volume) - A collection of objects (e.g., the list of all LUNs provided by a specific storage array)

At the highest level, Swordfish identifies two major collections: Storage Systems and Storage Services.

4.6 Query options

Swordfish queries can include arbitrary sets of query options to further refine the target of given query or the actions being requested of that target. These general query options are summarized in [Table 7: Query Options](#).

Note: Additional query options may be supported (or constrained) for a specific query target or resource path. These target-specific query options will be addressed in specific use case descriptions, as required.

Table 7: Query Options

Parameter Name	Arguments	Notes
<code>\$skip=<i>n</i></code>	Integer	Omit the first <i>n</i> entries in the collection from the returned set of objects (required by redfish)
<code>\$top=<i>n</i></code>	Integer	Return, at most, the first <i>n</i> entries in the returned set of objects (required by redfish)
<code>\$filter=<i>condition</i></code>	Filter Expression	Returns only the members of the named collection that match the provided logical expression (required by swordfish)
<code>\$expand=<i>target</i></code>	Expand Expression	Expand additional detail on the target property(s) in the returned result set (required by swordfish)
<code>\$select=<i>property list</i></code>	Comma-separated list of object properties	Return the named properties for each object in the result set, rather than the entire object (required by swordfish)
<code>\$orderby=<i>filter condition</i></code>	Filter Expression	sort the result set by the output values from the filter expression (required by swordfish)

4.7 Filter expressions

1. Simple example `$filter=(age gt 30)` A group of people never to trust.

For more information see [Filter Expression](#) in the OData specification.

4.8 HTTP status codes

Swordfish clients may receive any of the standard HTTP status codes. For status codes greater than 400, the server can also return extended status information as a simple JSON object (see [Redfish Specification](#) for more information).

5 Actors

5.1 Actors

This document covers a broad range of common use cases and storage management operations.

In an attempt to serve as both a introductory text and as a reference tool, the use cases have been grouped in a number of different ways:

- Alphabetically. Each use case is ordered according to a self-explanatory title, to make it easier for an experienced user to find a

specific piece of guidance.

- By management domain. While each organization will allocate responsibilities differently, there are general broad classes of storage management that share tasks. These are called [management domains](#).
- By actor. Similarly, the precise titles and responsibilities within storage management organizations will differ from one organization to the next, but this guide has identified three broad functional roles, and identified the use cases that are most applicable to those roles:
- [Cloud Admin](#): tasked with managing all aspects of cloud-based infrastructure including storage;
- [Storage Admin](#): tasked with operational storage tasks, such as storage lifecycle planning and day-to-day storage administration;
- [DevOps](#): tasked with leveraging storage configurations to automate and scale business operations.

5.2 CloudAdmin

A Cloud Administrator (CloudAdmin) is a converged infrastructure administrator, working with systems that are:

- Hyper-converged
- Rack-converged
- Hyper-scale

The CloudAdmin role in an enterprise or service provider is the individual or group primarily responsible for managing the operational lifecycle of a cloud, virtualization, converged environment that consists of the workloads, resource abstractions, storage, networking, and compute.

Also referred to as a "cloud architect", this role:

- designates a group of people to build and maintain a virtualized, converged, and/or cloud environment
- spans compute, storage, and networking disciplines as their job is to manage the operational lifecycle of the entire environment
- focuses on automation that involves scripting and potentially formal programming
- This role's value is to keep the environment that hosts applications available and responding. Deep subject matter expertise is less valuable unless it helps solve an issue.
- The focus for this role is to streamline the deployment of applications into the infrastructure including all the network and storage configuration.
- This role gets involved in the physical deployment of capacity in the datacenter.
- This role deals with software defined infrastructure deployment and management.
- Applications can span physical, virtual machine, container, PaaS building blocks. This role is expected to know how to best configure the "stack" to consume the underlying physical infrastructure.

5.2.1 Use Cases

- [Add volume](#)
- [Allocate volume](#)
- [Allocate volume using default class of service](#)
- [Create file share](#)
- [Create storage group](#)
- [Discover class of service](#)
- [Find storage service](#)

- [Get capacity by class of service](#)
- [List supported line of service options](#)
- [Protect volumes of an application](#)
- [Set capacity usage threshold](#)
- [Subscribe to threshold events](#)

5.3 DevOps

A member of the DevOps group:

- Consumes infrastructure capacity offered as a service/building blocks.
- Develops and deploys programmatic requests for capacity (fully automated, no ticketing or human intervention)
- Provisions storage as a virtual appliance on-premises or in the cloud as part of a larger application deployment
- Deploys 'cloud born apps' to consume object storage APIs (S3, Swift)

This role is typically aligned with the business unit. Their focus is the delivery, deployment, and maintenance of apps on IaaS and PaaS resources. This role is not typically a deep subject matter expert in compute, storage, or networking. From a development perspective, their desire is to treat the infrastructure as a programmable subsystem that presents resources on-demand.

This role:

- does not get involved in the physical deployment of capacity in the datacenter
- is responsible for automating infrastructure provisioning as part of the E2E deployment and management of an application with minimal or no human intervention
- consumes higher level services and protocols from the infrastructure; understanding how the device works is not interesting
- expects to consume the programmable interfaces of a device using their existing tools (REST, Python, Ansible, Puppet, Chef, Ruby etc.)

5.3.1 Use Cases

- [Allocate space for VM](#)
- [Allocate volume](#)
- [Allocate volume using default class of service](#)
- [Create file share](#)
- [Discover Class of Service](#)
- [Find storage service](#)
- [Get capacity by class of service](#)
- [Health updates](#)
- [Protect volumes of an application](#)
- [Review storage health report](#)
- [Set capacity usage threshold](#)
- [Subscribe to threshold events](#)

5.4 Storage Admin

A storage administrator designs storage solutions for modern environments, including:

- Virtualization (traditional virtualization VMware, Hyper-V)
- private cloud (self-service portal)
- hybrid cloud (can span private, colo, hosted, and public cloud)
- IaaS/PaaS stacks (modern app/devops)

The storage admin role in an enterprise or service provider can have responsibility for managing the operational lifecycle of storage in the datacenter. In particular:

- Organizations can have one or more people exclusively assigned to operating lifecycle of storage in the datacenter.
- The storage admin role is typically tasked with "figuring out" storage needs for the company
- The admin role can consist of level 2 operators and level 3 engineers and architects.
- The admin role deals with multiple storage devices from multiple vendors across one or more datacenters.
- Storage devices attached to multiple operating systems (Linux, Windows, Mainframe, Unix, virtualized, bare metal).
- The Admin role consists of deep storage subject matter expertise, may or may have general practitioners across all disciplines
- Members of this role are typically skilled at scripting and formal programming
- In a converged environment (storage, compute, and networking converged in to a rack), the storage admin role may choose to avoid/minimize involvement in the support of storage WITHIN the rack. (e.g. in virtualization, a single physical port can host 1000s of VMs).

5.4.1 Use Cases

- [Allocate volume](#)
- [Allocate volume using default class of service](#)
- [Create class of service](#)
- [Create file share](#)
- [Create file system](#)
- [Create line of service](#)
- [Create storage pool](#)
- [Discover class of Service](#)
- [Expand storage volume](#)
- [Find storage service](#)
- [Get capacity by class of service](#)
- [List supported line of service options](#)
- [Protect volumes of an application](#)
- [Set capacity usage threshold](#)
- [Subscribe to threshold events](#)

6 Management Domains

6.1 Management Domain Overview

This document covers a broad range of common use cases and storage management operations.

In an attempt to serve as both a introductory text and as a reference tool, the use cases have been grouped in a number of different ways:

- Alphabetically. Each use case is ordered according to a self-explanatory title, to make it easier for an experienced user to find a specific piece of guidance.
- By actor. While the precise titles and responsibilities within storage management organizations will differ from one organization to the next, this guide has identified three broad functional roles, known as [actors](#) and identified the use cases that are most applicable to those roles;
- By management domain. Similarly, while each organization will allocate responsibilities differently, there are general broad classes of storage management that share tasks. This guide has identified four management domains:
 - [Application Storage Management](#): which manages the interface between applications and the storage that they rely upon;
 - [Block Storage Management](#): focused on the management of resources that provide block-based access to storage;
 - [File System Management](#): which provides access to byte-accessible storage through file-based protocols;
 - [Service Catalog Management](#): which supports access to, and management of, a catalog of service options.

6.2 Application storage management domain

This domain manages the interface between applications and the storage that they rely upon.

StorageGroups provide a means to collectively manage the Volumes and FileShares utilized by an Application. The StorageGroup specifies whether the collected resources are managed so that storage is updated or replicated consistently across all members. Additionally, the StorageGroup provides the means to atomically expose (or hide) the collected resources to (or from) host endpoints.

6.2.1 Use cases

- [Discover Class of Service](#)
- [Find storage service](#)
- [Get capacity by class of service](#)
- [List supported line of service options](#)
- [Set capacity usage threshold](#)
- [Subscribe to threshold events](#)

6.3 Block storage management domain

Many devices and services provide their storage capacity to external devices and applications through block-based protocols to storage devices. This domain includes the management of resources that provide block-based access to storage.

Block-based storage is represented by a Volume. This domain provides for the discovery and provisioning of Volumes and for maintaining relationships to Device, Endpoint, StorageService, StorageGroup, StoragePool, and ComputerSystem resources.

6.3.1 Use Cases

- [Add volume](#)
- [Allocate VM space](#)
- [Allocate volume](#)
- [Allocate volume using default class of service](#)
- [Create class of service](#)
- [Create line of service](#)
- [Create storage group](#)
- [Create storage pool](#)
- [Discover class of service](#)
- [Expand storage volume](#)
- [Find storage service](#)
- [Get capacity by class of service](#)
- [List supported line of service options](#)
- [Protect volumes of an application](#)
- [Review volume metrics](#)
- [Set capacity usage threshold](#)
- [Subscribe to threshold events](#)

6.4 Service catalog management domain

Swordfish supports access to, and management of, a catalog of service options, (see [ITIL glossary and abbreviations](#)), supported by storage services.

The ClassOfService resource represents a service option that may be used to specify requirements for utility or warranty when provisioning a resource. Currently ClassOfService is defined for use in the Block Storage, File System, and ApplicationStorage domains.

The service catalog for each StorageService is represented by a collection of references to supported ClassOfService resources. Each ClassOfService is known minimally by a Name and a unique Identifier. When a ClassOfService is specified for a resource, the StorageService shall attempt to maintain that resource in compliance to the requirements of that ClassOfService. The requirements may be specified informally by text in the Description property or may be specified formally by the property values of embedded options related to specific lines of service.

The embedded service options are described by values of complex types representing lines of service.

Over time, as the service catalog is continually updated to match evolving user needs and service provider offerings, it is expected that the catalog will contain entries (one or more ClassOfService or LineOfService instances) that are not currently active.

6.4.1 Data protection

The primary storage is described by a ClassOfService resource. That ClassOfService may aggregate any number of data protection service options. Each instance of a data protection service option describes the characteristics a replication session that shall be maintained for the containing primary storage resource.

For additional information, see the definitions for DataProtectionLineOfService and DataProtectionLoSCapabilities.

6.4.2 Data security

An instance of a data security service option describes an optional set of data security requirements. A data security Service option is typically aggregated into a ClassOfService resource that is associated with storage. At most one data security service option may be aggregated into a ClassOfService resource. When storage is provisioned with that ClassOfService, it will provide the currently specified data security characteristics.

A data security service option may specify data security characteristics that 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. In this case, the names of the standard or standards can usefully be included in the user/admin-visible name of the instance. With the notable exception of FIPS-140, compliance requires measures well beyond the means of a storage system to provide (e.g., both HIPAA and PCI impose significant requirements on administration and operation of the data center), so this approach promises that the storage system will do its part in supporting compliance, but does not (and cannot) promise that the storage system will deliver full compliance by itself.

The description attribute value may include human readable information including:

- Whether encryption keys are drive or array resident or externally managed (e.g., via KMIP).
- Information on how the array supports compliance to a standard identified in the name of the Service option. (e.g., specific algorithms employed that are FIPS-140 compliant, information about the validated cryptographic module and its validation certificate, relationship of the security functionality to specific PCI or HIPAA requirements).

NOTE For comparable cryptographic strengths, (see [NIST SP 800-57 part 1](#))

NOTE For symmetric encryption algorithm key sizes, 112 bits is the 3DES key size and 128, 192, and 256 bits are options for AES key sizes.

NOTE MediaEncryptionStrength includes the case where metadata about the data must be encrypted. (e.g. data presence vs. absence in a thin volume, array filesystem metadata) The implementation may be self-encrypting drives or encryption in the storage system's drive controller. Keys may be drive or array resident or externally managed (e.g., via KMIP).

For additional information, see the definitions for DataSecurityLineOfService and DataSecurityLoSCapabilities.

6.4.3 Data storage

Each data storage service option describes characteristics of the storage at a particular location. A class of service will have at most one data storage service option, which describes the storage specified by that class of Service.

For additional information, see the DataStorageLineOfService and DataProtectionLoSCapabilities.

6.4.4 IO connectivity

An IO connectivity service option specifies the characteristics of storage connectivity. For each value of AccessProtocol, at most one IO connectivity service option may be aggregated into a class of service.

NOTE: If used within a ClassOfService for Storage Provisioning, this value constrains the set of connections used to expose that storage.

For additional information, see the IOConnectivityLineOfService and IOConnectivityLoSCapabilities.

6.4.5 IO performance

An IO performance service option specifies a choice of performance characteristics as viewed through the data path to the storage. This is affected by choices of storage and connection technologies.

At most one IO performance service option may be aggregated into a `ClassOfService` for a storage resource. When storage is provisioned with that `ClassOfService`, it should provide at least the specified performance.

For additional information, see the `IOConnectivityLineOfService` and `IOConnectivityLoSCapabilities`.

6.4.6 Use cases

- [Create class of service](#)
- [Discover class of service](#)
- [Create line of service](#)
- [Find storage service](#)
- [Get capacity by class of service](#)
- [List supported line of service options](#)

6.5 File system storage management domain

`FileSystems` provide access to byte-accessible storage through file-based protocols. This domain includes the management of resources that provide file-based access to storage.

File-based storage is represented by a `FileSystem` resources. Remote access to portions of a `FileSystem` is provided by `FileShare` resources.

6.5.1 Use cases

- [Create File Share](#)
- [Create File System](#)
- [Discover Class of Service](#)
- [Find storage service](#)
- [Get capacity by class of service](#)
- [List supported line of service options](#)

7 User Guidance

7.1 Default Class of Service

If a pool, storage volume or other construct is created with no specified class of service when a class of service exists, the implementation will attempt to apply the `DefaultClassOfService`.

8 Use Cases

Title	Description
Add volume	Add a volume to a StorageGroup
Allocate VM Space	Allocate space for a new VM
Allocate volume	Allocate a new volume
Allocate volume using default class of service	Allocate a new volume without default class of service
Create class of service	Create a class of service
Create file share	Create a file share
Create file system	Create a files system
Create line of service	Create a line of service
Create StorageGroup	Create a StorageGroup
Create Storage Pool	Create a StoragePool
Discover class of service	Discover a class of service
Expand Volume	Expand capacity of a Volume
Find storage service	Find a services support a class of service
Get capacity by class of service	Summarized capacity by a class of service
Health Updates	Example of health status monitoring
List supported line of service options	List Supported Line Of Service Options
Set space thresholds	Set Capacity Usage Thresholds
Subscribe to thresholds events	Subscribe to Threshold Events
Protect Volumes of an application	Protect Volumes of an Application

8.1 Add Volume

Summary: Add Volumes to a StorageGroup

Purpose: Add Volumes to a StorageGroup

Who: [CloudAdmin](#)

Management Domain: [Block storage management](#)

Triggers: A need for expanded capacity in a StorageGroup

Detailed Context: A Volume has been created, and needs to be added to an existing StorageGroup.

Preconditions: None.

Inputs: - *{SGURL}*: URL for storage group - One or more URLs for Volumes to add [*{VURI1}*, *{VURI2}*, ...]

Basic Course of Events:

1. Add the Volume(s) the Members collection of the StorageGroup

- **Request:**

POST {SGURL}/Volumes

- **Headers:** No additional headers required.
- **Body:**

```
[{"VURI_1"}, {"VURI_2"}, ... ]
```

- **Response:** None defined.

Postconditions: The selected Volume(s) are added to the Members collection for the Storage Group.

Failure Scenario: None defined.

See also: None defined.

8.2 Allocate VM space

Summary: Allocate space for a VM (e.g., Docker) deployment

Purpose: Allocate space for a VM (e.g., Docker) deployment

Who: [DevOps](#)

Management domain: [Block storage management](#)

Triggers: DevOp needs to set up a new VM to host a Ceph OSD service.

Detailed context: Some cloud web services deploy a large number of VMs (or containers) that do not keep data in traditional files or block storage; instead, they store data in remote object stores. This use case relates to setting up systems hosting Ceph as an object store. Ceph is deployed as a set of collaborating systems providing fault tolerance; this use case addresses one VM dedicated as a Ceph Object store (OSD) system. The VM is allocated with a virtual disk for the OS and Ceph software; an external disk is allocated for Ceph data. For this use case, the DevOP prefers using an HDD rather than an SSD and needs at least 400 Gigabytes (400,000,000,000 bytes).

Preconditions: The catalog (see [JBOD discovery](#) use case) must exist and include at least one available HD

Inputs:

- {MEDIA}: The MediaType (in this case, HDD)
- {CAPACITY}: Capacity in bytes (in this case, at least 400,000,000,000)

Basic course of events:

- Devop searches for Drives in the catalog, looking for those with MediaType = HDD and CapacityBytes >= 400,000,000,000)
- A matching drive is selected. The admin notes the drive serial number and updates the catalog to mark the drive as allocated
- If necessary, the Devop makes the appropriate zoning changes to make the drive accessible to the server with the hypervisor

hosting the Ceph VM

- The Devop configures the hypervisor to make this drive available to the Ceph V

Postconditions: The Devop can now install and configure Ceph to use the selected drive

Failure Scenario: None defined.

See also: None defined.

8.3 Allocate Volume

Summary: Allocate a Volume

Purpose: Allocate a Volume with a known capacity and class of service.

Who: [CloudAdmin](#), [StorageAdmin](#), [DevOps](#)

Management Domain: [Block storage management](#)

Triggers: Need to allocate storage for a new application.

Detailed Context: The admin needs to satisfy a service request to provide a given amount of storage to an application, and to assure a given class of service.

Preconditions: None.

Inputs:

- **_URL for Storage Service:** `/redfish/v1/StorageServices(1)`
- **_Requested volume size:** `1 TiB` or `1099511627776 bytes`
- **_URL for requested class of service:** `/redfish/v1/StorageServices(1)/ClassesofService(BostonBunker)`
- **_Requested name of volume (string):** `Snapshot1`

Basic Course of Events:

1. Post the definition of the new volume to the Volumes resource collection.

This instructs the service to allocate a new volume of the requested size that meets the requirements of the specified class of service. Since additional details are not provided, the service is free to allocate the storage from any of its storage pools that can satisfy the request.

- **Request:** `POST /redfish/v1/StorageServices(1)/Volumes/Members`
- **Headers:** No additional headers required.
- **Body:**

```
{
  "Name": "Snapshot1",
  "CapacityBytes": 1099511627776,
  "Links": {
    "ClassOfService": {"odata.id": "/redfish/v1/StorageServices(1)/ClassesofService(BostonB
unker)"}
  }
}
```

- **Response:**

- **Headers:**

Location = /redfish/v1/StorageServices(1)/Volumes(3)

- **Body:**

```
{
"@SSM.Copyright": "Copyright (c) 2014-2016 SNIA. All rights reserved.",
"@odata.context": "/redfish/v1/$metadata#Volume.Volume",
"@odata.id": "/redfish/v1/StorageServices(1)/Volumes(3)",
"@odata.type": "#Volume_1_0_0.Volume",
"Name": "Snapshot1",
"Id": "3",
"Description": "",
"Identifiers": [
  {
    "DurableNameFormat": "NAA6",
    "DurableName": "65456765456761001234076100123487"
  }
],
"Manufacturer": "SuperDuperSSD",
"Model": "Drive Model string",
"Status": {
  "State": "Enabled",
  "Health": "OK"
},
"AccessCapabilities": [
  "Read",
  "Write",
  "Append",
  "Streaming"
],
"BlockSizeBytes": 512,
"CapacitySources": [ {
  "ConsumedBytes": 0,
  "AllocatedBytes": 10737418240,
  "GuaranteedBytes": 536870912,
  "ProvisionedBytes": 1099511627776,
  "Links": {
```

```
    "ClassOfService": {
      "@odata.id": "/redfish/v1/StorageServices(1)/Links/ClassesOfService(SilverBoston)"
    },
    "ProvidingPool": {
      "@odata.id": "/redfish/v1/StorageServices(1)/StoragePools(SpecialPool)"
    }
  }
}],
"CapacityBytes": 1099511627776,
"Capacity": {
  "Data": {
    "ConsumedBytes": 0,
    "AllocatedBytes": 10737418240,
    "GuaranteedBytes": 1099511627776,
    "ProvisionedBytes": 1099511627776
  },
  "Metadata": {
    "ConsumedBytes": 536870912,
    "AllocatedBytes": 536870912
  }
},
"Links": {
  "ClassofService": {
    "@odata.id": "/redfish/v1/StorageServices(1)/Links/ClassesofService/BostonBunker"
  }
}
}
```

Postconditions: The selected volumes are added to the collection for the Storage Group.

Failure Scenario: None defined

See also: None defined.

8.4 *Allocate Volume using Default Class of Service*

Summary: Allocate a Volume using the Default Class of Service

Purpose: Allocate a Volume with a known capacity and without specifying the class of service, invoking the DefaultClassOfService.

Who: [CloudAdmin](#), [StorageAdmin](#), [DevOps](#)

Management Domain: [Block storage management](#)

Triggers: Need to allocate storage for a new application.

Detailed Context: The admin needs to satisfy a service request to provide a given amount of storage to an application, but does

not specify a given class of service. The system will attempt to provision the system with the specified ClassOfService in DefaultClassOfService in either the given StoragePool or the StorageService as available.

Preconditions: The DefaultClassOfService property is set in either the StoragePool or StorageService.

Inputs:

- `_URL` for Storage Service: `/redfish/v1/StorageServices(1)`
- `_Requested volume size in TB (int64)`: `1099511627776`
- `_Requested name of volume (string)`: `Snapshot1`

Basic Course of Events:

1. Post the definition of the new volume to the Volumes resource collection.

This instructs the service to allocate a new volume of the requested size that meets the requirements of the specified class of service. Since additional details are not provided, the service is free to allocate the storage from any of its storage pools that can satisfy the request.

- **Request:** `POST /redfish/v1/StorageServices(1)/Volumes`
- **Headers:** No additional headers required.

- **Body:**

```
{
  "Name": "Snapshot1",
  "CapacityBytes": 1099511627776
}
```

- **Response:**

- **Headers:**

```
Location = /redfish/v1/StorageServices(1)/Volumes(3)
```

- **Body:**

```
{
  "@SSM.Copyright": "Copyright (c) 2014-2016 SNIA. All rights reserved.",
  "@odata.context": "/redfish/v1/$metadata#Volume.Volume",
  "@odata.id": "/redfish/v1/StorageServices(1)/Volumes(3)",
  "@odata.type": "#Volume_1_0_0.Volume",
  "Name": "Snapshot1",
  "Id": "3",
  "Description": "",
  "Identifiers": [
    {
      "DurableNameFormat": "NAA6",
      "DurableName": "65456765456761001234076100123487"
    }
  ],
  "Manufacturer": "SuperDuperSSD",
  "Model": "Drive Model string",
}
```

```
"Status": {
  "State": "Enabled",
  "Health": "OK"
},
"AccessCapabilities": [
  "Read",
  "Write",
  "Append",
  "Streaming"
],
"BlockSizeBytes": 512,
"CapacitySources": [ {
  "ConsumedBytes": 0,
  "AllocatedBytes": 10737418240,
  "GuaranteedBytes": 536870912,
  "ProvisionedBytes": 1099511627776,
  "Links": {
    "ClassOfService": {
      "@odata.id": "/redfish/v1/StorageServices(1)/Links/ClassesOfService(SilverBoston)"
    }
  }
},
  {
    "ProvidingPool": {
      "@odata.id": "/redfish/v1/StorageServices(1)/StoragePools(SpecialPool)"
    }
  }
}],
"Capacity": {
  "Data": {
    "ConsumedBytes": 0,
    "AllocatedBytes": 10737418240,
    "GuaranteedBytes": 1099511627776,
    "ProvisionedBytes": 1099511627776
  },
  "Metadata": {
    "ConsumedBytes": 536870912,
    "AllocatedBytes": 536870912
  }
},
"Links": {
  "ClassofService": {
    "@odata.id": "/redfish/v1/StorageServices(1)/Links/ClassesofService/BostonBunker"
  }
}
}
```

Postconditions: The selected volumes are added to the collection for the Storage Group.

Failure Scenario: None defined

See also: None defined.

8.5 Health updates

Summary: Retrieve storage metrics information for a storage volume.

Purpose:

- Application monitoring tool warns devops that the application is processing less requests based on a predefined threshold
- Devops has a simple script that queries Redfish capable infrastructure and pulls performance metrics
- Devops collect infrastructure performance metrics as part of an end-to-end diagnostics workflow to help identify potential bottlenecks

Who: [DevOps](#) at an enterprise

Management domain: [Block storage management](#)

Trigger: Lower than expected application requests completed per second

Detailed context: Devops creates diagnostics scripts that retrieve performance information from multiple layers in the application stack, including infrastructure, to help identify potential bottlenecks during production hours. Devops already understand that the issue is not in the application layers so now they have to dig deeper. Members of devops are typically not experts in infrastructure compute, storage, and networking so they need simple scripts that can provide the information they require without a deep understanding of underlying hardware. With the [storage administrator's](#) help, devops use a few simple GETs on storage objects related to their application. To help simplify the query, the storage administrator tags the volumes associated with the application. Devops can use stack wide performance metrics to quickly isolate potential bottlenecks that may be contributing to the slow down in the application.

Preconditions:

- Storage system has at least one storage pool with at least one storage volume
- Storage volume is exposed to at least one initiator
- Initiator is driving I/O to the volume

Inputs: DevOps knows the storage system name or IP address with the Swordfish service and the storage pool volume names.

Basic course of events:

1. uses GET operations to look at metrics of the storage volume
 - **Request:** GET /redfish/v1/StorageServices/1/Volumes/4/Metrics
 - **Headers:** No additional headers required.
 - **Body:** None defined.
 - **Response:**

```
{
  "@Redfish.Copyright": "Copyright 2015-2016 SNIA. All rights reserved.",
  "@odata.context": "/redfish/v1/$metadata#VolumeMetrics.VolumeMetrics",
  "@odata.id": "/redfish/v1/StorageServices/1/Volumes/4/Metrics",
  "@odata.type": "#VolumeMetrics.v1_0_0.VolumeMetrics",
  "Id": "Metrics",
  "Name": "Volume 1 Metrics",
  "CurrentPeriod": {
    "BlocksRead": 125534,
    "BlocksWritten": 542653
  },
  "Lifetime": {
    "BlocksRead": 125534,
    "BlocksWritten": 542653
  },
  "PerformanceData": {
    "AverageSecondsPerRead": 1,
    "AverageSecondsPerWrite": 1,
    "AverageSecondsPerTransfer": 1,
    "ReadsPerSecond": 2134,
    "WritesPerSecond": 4325,
    "TransfersPerSecond": 6459,
    "WriteBlocksPerSecond": 3085784,
    "ReadBlocksPerSecond": 9257350,
    "BlocksPerSecond": 12343134
  },
  "Actions": {
    "#Volume.ClearCurrentPeriod": {
      "target": "/redfish/v1/StorageServices/1/Volumes/4/Actions/Volume.ClearCurrentPeriod"
    }
  },
  "Oem": {}
},
{
  "Oem": {}
}
}
```

Postconditions: None defined.

See also: None defined.

8.6 *Create class of service*

Summary: Create a class of service

Purpose: Create a new class of service in the service catalog to match a newly available type of storage

Who: [StorageAdmin](#)

Management Domain: [Block storage management](#), [Service catalog management](#)

Triggers: The administrator has determined that a new class of service needs to be created to reflect a new class of SSD storage in the infrastructure.

Detailed Context: This is a simple scenario where the primary characteristic is the enhanced performance available of SSD drives.

Preconditions: None defined.

Inputs:

- URL for Storage Service: `/redfish/v1/StorageServices(1)`
- New class of service characteristics
- Name: "SSD"
- Description: "Minimal SSD class of service."
- IOPerformanceLineOfService
 - "Name": "SSDLoS"
 - "IOOperationsPerSecondIsLimited": `false`
 - "MaxIOOperationsPerSecondPerTerabyte": `100000`
 - "AverageIOOperationLatencyMicroseconds": `10`

Basic Course of Events:

1. Create ClassOfService

- **Request:** `POST /redfish/v1/StorageServices(1)/ClassesOfService/Members`
 - **Headers:** No additional headers required.
 - **Body:**

```
{
  "Name": "SSD",
  "Description": "Minimal SSD class of service.",
  "LinesOfService": {
    "IOPerformanceLinesOfService": [ {
      "Name": "SSDLoS",
      "IOOperationsPerSecondIsLimited": false,
      "MaxIOOperationsPerSecondPerTerabyte": 100000,
      "AverageIOOperationLatencyMicroseconds": 10
    }
  ]
}
```

2. **Response:**

- **Headers:**

Location= `/redfish/v1/StorageServices(1)/ClassesOfService(SSD)`

Postconditions: The requested class of service is added to the ClassesOfService collection.

Failure Scenario: None defined.

See also: None defined.

8.7 Create file share

Summary: Create a file share

Purpose: Share an existing file system as /Shares/MyShare

Who: [CloudAdmin](#), [StorageAdmin](#), [DevOps](#)

Management Domain: [File system storage management](#)

Triggers: None defined.

Detailed Context: Create a share starting at /Shares/MyShare.

Preconditions: None defined.

Inputs:

- URL for the filesystem: /redfish/v1/StorageServices(1)/FileSystems(QuickFiles)
- The path to the shared file: "/Shares/MyShare"
- Description: "Share of files under MyShare."

Basic Course of Events:

1. Create a file share

- **Request:** POST /redfish/v1/StorageServices(1)/FileSystems(QuickFiles)/ExportedShares
- **Headers:** No additional headers required.
- **Body:**

```
{  
    "Name": "MyShare",  
    "Description": "Share of files under MyShare.",  
    "SharedFilePath": "/Shares/MyShare"  
}
```

- **Response:**

- **Headers:**

Location= /redfish/v1/StorageServices(1)/FileSystems(QuickFiles)/ExportedShares(MyShare)

Postconditions: The requested file share is added to the ExportedShares collection for the file system.

Failure Scenario: None defined.

See also: None defined.

8.8 Create file system

Summary: Create a file system

Purpose: Create a file system with a given capacity and performance level.

Who: [StorageAdmin](#)

Management Domain: [File system storage management](#)

Triggers: None defined.

Detailed Context: Create a 100 TB file system based on SSD class storage.

Preconditions: None defined.

Inputs:

- URL for storage service: `/redfish/v1/StorageServices(1)`
- Name for the new file system: `"QuickFiles"`
- Description: `"100 TB FileSystem having SSD class storage."`
- URL for class of service: `/redfish/v1/StorageServices(1)/Links/ClassesOfService(SSD)`
- File system capacity:

```
json { "Data": { "ProvisionedBytes": 100000000000000; "IsThinProvisioned":
```

Basic Course of Events:

1. Create FileSystem

- **Request:** `POST /redfish/v1/StorageServices(1)/FileSystems`
- **Headers:** No additional headers required.
- **Body:**

```
{
  "Name": "QuickFiles",
  "Description": "100 TB FileSystem having SSD class storage.",
  "Capacity": {
    "Data": {
      "ProvisionedBytes": 100000000000000
    },
    "IsThinProvisioned": true
  },
  "Links": {
    "ClassOfService": { "odata.id":
      "/redfish/v1/StorageServices(1)/Links/ClassesOfService(SSD)" }
  }
}
```

- **Response:**

- **Headers:**

Location= /redfish/v1/StorageServices(1)/FileSystems(QuickFiles)

Postconditions: The requested file system is added to the FileSystems collection for the Storage Service.

Failure Scenario: None defined.

See also: None defined.

8.9 Create line of service

Summary: Create a line of service to reflect the performance characteristics of SSD storage

Purpose: The definition is created here in preparation of creating ClassOfService instances that include a requirement for SSD storage performance.

Who: [StorageAdmin](#)

Management Domain: [Block storage management](#), [Service catalog management](#)

Triggers: None defined.

Detailed Context: SSD storage is introduced and need a new performance line of service to reflect their capability.

Preconditions: None defined.

Inputs:

- URL for Storage Service: /redfish/v1/StorageServices(1)
- New IO performance line of service

```
json { "Name": "NewSSDLoS", "IoOperationsPerSecondIsLimitedBoolean": false, "MaxIoOperationsF
```


Basic Course of Events:

1. Get existing supported lines of service

- **Request:**

GET /redfish/v1/StorageServices(1)/Links/IOPerformanceLoSCapabilities/SupportedIOPerformanceLinesOfSe

- **Headers:** No additional headers required.

- **Reply:**

- **Headers:** `ETag: "123-a"``

- **Body:**

```
{
  "Value": [{
    "Name": "LoS1",
    "IoOperationsPerSecondIsLimitedBoolean": false,
    "SamplePeriodSeconds": 60,
    "MaxIoOperationsPerSecondPerTerabyte": 83,
    "AverageIoOperationLatencyMicroseconds": 8000,
  },
  {
    "Name": "LoS2",
    "IoOperationsPerSecondIsLimitedBoolean": "false",
    "SamplePeriodSeconds": 60,
    "MaxIoOperationsPerSecondPerTerabyte": 133,
    "AverageIoOperationLatencyMicroseconds": 5000,
    "IOWorkload": {
      "Name": "Duplicon: OLTP"
    }
  }
  ]
}
```

2. Create new line of service

- **Request:**

PATCH /redfish/v1/StorageServices(1)/Links/IOPerformanceLoSCapabilities

- **Headers:**

`If-Match: "123-a"``

- **Body:**

```
{
  "SupportedIOPerformanceLinesOfService": [{
    "Name": "LoS1",
    "IoOperationsPerSecondIsLimitedBoolean": false,
    "SamplePeriodSeconds": 60,
    "MaxIoOperationsPerSecondPerTerabyte": 83,
    "AverageIoOperationLatencyMicroseconds": 8000
  },
  {
    "Name": "LoS2",
    "IoOperationsPerSecondIsLimitedBoolean": "false",
    "SamplePeriodSeconds": 60,
    "MaxIoOperationsPerSecondPerTerabyte": 133,
    "AverageIoOperationLatencyMicroseconds": 5000,
    "IOWorkload": {
      "Name": "Duplicon: OLTP"
    }
  },
  {
    "Name": "NewSSDLoS",
    "IoOperationsPerSecondIsLimitedBoolean": "false",
    "MaxIoOperationsPerSecondPerTerabyte": 100000,
    "AverageIoOperationLatencyMicroseconds": 10
  }
  ]
}
```

- **Response:**
- **Headers:** None defined.
- **Body:** None defined.

Postconditions: The requested line of service is added to the SupportedIOPerformanceLinesOfService of the Storage Service.

Failure Scenario: None defined.

See also: None defined.

8.10 Create StorageGroup

Summary: Create a StorageGroup

Purpose: Create a StorageGroup

Who: [CloudAdmin](#)

Management Domain: [Block storage management](#)

Triggers: None defined.

Detailed Context: Create a collection of application storage that is exposed to an application and managed as a unit.

Preconditions: None defined.

Inputs:

- *{SSURL}*: URL for storage service
- *{NAME}*: the proposed name of the new StorageGroup (string)
- *{DESC}*: a description of the new StorageGroup (string)
- *{CREQ}*: a consistency requirement for the new StorageGroup (boolean). When TRUE, all copies of data within a StorageGroup must be kept in sync.

Basic Course of Events:

1. Create StorageGroup

- **Request:** POST *{SSURL}/StorageGroups/Members*

- **Headers:** No additional headers required.

- **Body:**

```
json { Name: "{NAME}", Description: "{DESC}", MembersAreConsistent: {CREQ} }
```

- **Response:** None defined.

Postconditions: The requested StorageGroup is added to the Members collection for the Storage Service.

Failure Scenario: None defined.

See also: None defined.

8.11 *Create storage pool*

Summary: Create a StoragePool

Purpose: Create a StoragePool

Who: [StorageAdmin](#)

Management Domain: [Block storage management](#)

Triggers: Users need to allocate storage with characteristics satisfied by a class of service.

Detailed Context: Create a storage pool containing an amount of storage that can be used to create a requested class of service. The storage pool implementation will attempt to find and allocate enough storage that will satisfy the request. No metadata or snapshot storage is reserved.

Preconditions: None defined.

Inputs:

- URL for storage service: */redfish/v1/StorageServices(1)*

- Name for the new storage pool: "SSD"
- Description: "100 TB pool of SSD class storage."
- URL for class of service: /redfish/v1/StorageServices(1)/ClassesOfService(SSD)
- Storage pool capacity

```
{
  "Data": {
    "ProvisionedBytes": 1000000000000000,
    "IsThinProvisioned": false
  }
}
```

Basic Course of Events:

1. Create StoragePool

- **Request:** POST /redfish/v1/StorageServices(1)/StoragePools/Members
 - **Headers:** No additional headers required.
 - **Body:**

```
{
  "Name": "SSD",
  "Description": "100 TB pool of SSD class storage.",
  "Capacity": {
    "Data": {
      "ProvisionedBytes": 1000000000000000
    },
    "IsThinProvisioned": false
  },
  "ClassesOfService": {
    "Members": [{
      "@odata.id": "/redfish/v1/StorageServices/Members(1)/ClassesOfService(SSD)"
    }]
  }
}
```

- **Response:**

- **Headers:**

Location=/redfish/v1/StorageServices(1)/StoragePools(SSD)

Post-Conditions: The requested StoragePool is added to the collection for the Storage Service.

Failure Scenario: None defined.

See also: None defined.

8.12 Discover Class of Service

Summary: Discover a suitable class of service

Purpose: Identify an existing Class of Service that meets given requirements

Who: [CloudAdmin](#), [StorageAdmin](#), [DevOps](#)

Management Domain: [Block storage management](#), [File system storage management](#), [Application storage management](#), [Service catalog management](#)

Trigger: Need to provide a ClassOfService reference in another request (e.g., [Create Volume](#))

Detailed Context: Identify an existing class of service that would satisfy a set of characteristics (e.g., capacity, service level, throughput)

Preconditions: None.

Inputs: A set of storage attributes that must be satisfied, phrased as a [Filter Expression](#). For this example: we want at least 10000 IOPS per Terabyte as a desired minimum supported performance level

Basic Course of Events:

1. Use GET operation to retrieve possible ClassOfService entries

- **Request:**

GET /redfish/v1/StorageServices?\$expand=Members/ClassesOfService/Members(\$Filter=DataProtectionLi

- **Headers:** No additional headers required.
- **Body:** None.

- **Response:**

```
{
  "@SSM.Copyright": "Copyright © 2014-2016 SNIA. All rights reserved.",
  "@odata.context": "/redfish/v1/$metadata#ClassOfService.ClassOfService",
  "@odata.id": "/redfish/v1/StorageServices/FileService/ClassesOfService/BostonGoldJaw",
  "@odata.type": "#ClassOfService_1_0_0.ClassOfService",
  "Id": "BostonGoldJaw",
  "Name": "BostonGoldJaw",
  "Description": "Boston with Open Jaw config to Providence and Dallas",
  "ClassOfServiceVersion": "01.00.00",
  "IsDefault": false,
  "LinesOfService": {
    "IOConnectivityLineOfService": {
      "AccessProtocol": "FC",
      "MaxSupportedIoOperationsPerSecond": null
    },
    "IOPerformanceLineOfService": {
      "IoOperationsPerSecondIsLimitedBoolean": "false",
      "SamplePeriodSeconds": 60,
      "MaxIoOperationsPerSecondPerTerabyte": 10000,
      "MaxSupportedTerabytePerSecond": 10
    }
  }
}
```

```

    "AverageIoOperationLatencyMicroseconds":10,
    "IOWorkload":{
      "Name":"Duplicon:OLTP"
    }
  },
  "DataProtectionLineOfService":[{
    "RecoveryGeographicObjective":"AvailabilityZone",
    "RecoveryPointObjectiveSeconds":0,
    "RecoveryTimeObjective":"Online",
    "ProtectionType":"Mirror",
    "MinLifetimeSeconds":null,
    "IsIsolated":true,
    "Schedule":null,
    "ReplicaClassOfService":{
      "@odata.id":"/redfish/v1/StorageServices/FileService/ClassesOfService/Gold"
    },
    "ReplicaAccessLocation":{
      "Country":"","
      "Territory":"","
      "State":"","
      "City":"Providence",
      "Address1":"","
      "Address2":"","
      "Address3":"","
      "PostalCode":"","
      "Building":"","
      "Room":"","
      "Row":"","
      "Rack":"","
      "Shelf":"","
      "Item":"","
      "GPSCoords":"","
      "OtherLocationInfo":""
    }
  },
  {
    "RecoveryGeographicObjective":"AvailabilityZone",
    "RecoveryPointObjectiveSeconds":4,
    "RecoveryTimeObjective":"Online",
    "ProtectionType":"Continuous",
    "MinLifetimeSeconds":null,
    "IsIsolated":true,
    "Schedule":null,
    "ReplicaClassOfService":{
      "@odata.id":"/redfish/v1/StorageServices/FileService/ClassesOfService/Gold"
    },
    "ReplicaAccessLocation":{
      "Country":"","
      "Territory":"","
      "State":"","

```

```

        "City": "Dallas",
        "Address1": "",
        "Address2": "",
        "Address3": "",
        "PostalCode": "",
        "Building": "",
        "Room": "",
        "Row": "",
        "Rack": "",
        "Shelf": "",
        "Item": "",
        "GPSCoords": "",
        "OtherLocationInfo": ""
    }
}],
    "DataSecurityLineOfService": {
        "MediaEncryptionStrength": "Bits_256",
        "ChannelEncryptionStrength": "Bits_128",
        "HostAuthenticationType": "Ticket",
        "UserAuthenticationType": "Password",
        "SecureChannelProtocol": "TLS",
        "AntivirusScanPolicies": [
        ],
        "AntivirusEngineProvider": null,
        "DataSanitizationPolicy": "CryptographicErase"
    },
    "DataStorageLineOfService": {
        "RecoveryTimeObjective": 0,
        "ProvisioningPolicy": "Thin",
        "SpaceEfficient": true
    }
}
}
}

```

2. Select a Class of Service based on returned values.

Postcondition: The catalog now has entries for each new disk drive including class of service and capacity.

Failure Scenario:

Error Code	Cause
204	No existing Class of Service entries match the required criteria

See also: [Allocate Volume](#)

8.13 *Expand capacity of a storage volume*

Summary: Retrieve storage capacity information for a storage pool to check available capacity before expanding a storage volume.

Purpose:

- Application monitoring tool warns the admin that a storage volume used by a critical application is at 80% capacity
- Storage volumes can be expanded during the week without approval during non-business hours
- Administrator checks the available capacity on the storage pool based on the class of service required to ensure the storage volume expansion is possible
- During proper maintenance window, administrator expands storage volume based on predefined SLA established with application owner

Who: [StorageAdmin](#) at an enterprise

Management domain: [Block storage management](#)

Trigger: Low available capacity warning

Detailed context: The enterprise administrator offers managed storage services to business units in her organization. The administrator has established an SLA with application owners for storage volume expansion:

- Low storage capacity warning and critical alerts only apply to storage volumes with application data (the storage administrator is not responsible for the OS disks of the servers)
- Administrator can expand the storage volume by 25% up to 3 times without additional approvals for a maintenance window as long as the work is done during non-business hours (business hours are 7AM - 7PM local).
- Storage volumes for this application cannot exceed 2TB.
- Storage volume will be tagged with metadata to indicate the original size of the volume.
- Administrator is responsible for informing the application owner if the storage system is running low and overall capacity putting the SLA at risk. Administrator and application owner must plan for a migration of the application if waiting for additional storage disks/shelves to arrive is not feasible.

The application monitoring tool sends a warning alert to the administrator's datacenter monitoring tool indicating that a storage volume's capacity is at 80% and must be increased to ensure the application does not experience unplanned downtime. The administrator first checks the available capacity on the storage pool for a given storage class of service. Since the storage system is new, there is sufficient capacity available. Next the administrator figures out that the storage volume has never been expanded based on the original size tag. At the start of the maintenance window, non-business hours, the administrator initiates the expand action. For this particular storage system, the expand is a long running action so the administrator tracks the progress using the associated task.

Preconditions:

- Storage system has at least one storage pool with at least one storage volume
- Storage pool has enough available capacity to expand the storage volume by 25%

Inputs: Enterprise administrator knows the storage system name or IP address with the Swordfish service. Administrator knows the storage pool name and storage volume name.

Basic Course of Events:

1. uses GET operations to look at information about storage volume and confirm low capacity and the current size is less than 2TB
- **Request:** GET /redfish/v1/StorageServices/1/Volumes/61001234876545676100123487654567

- **Response:**

```
{
  "@Redfish.Copyright": "Copyright 2015-2016 SNIA. All rights reserved.",
  "@odata.context": "/redfish/v1/$metadata#Volume.Volume",
  "@odata.id": "/redfish/v1/StorageServices/1/Volumes/61001234876545676100123487654567",
  "@odata.type": "#Volume_1_0_0.Volume",
  "Id": "61001234876545676100123487654567",
  "Name": "Volume 1",
  "Description": "application storage",
  "Identifiers": [
    {
      "DurableNameFormat": "NAA6",
      "DurableName": "61001234876545676100123487654567"
    }
  ],
  "Manufacturer": "SuperDuperStorageProvider",
  "Status": {
    "State": "Enabled",
    "Health": "OK"
  },
  "BlockSizeBytes": 512,
  "LowSpaceWarningThresholdPercent": [
    80,
    null,
    null,
    null,
    null
  ],
  "Capacity": {
    "Data": {
      "ConsumedBytes": 879609302221,
      "AllocatedBytes": 879609302221,
      "GuaranteedBytes": 549755813888,
      "ProvisionedBytes": 1099511627776
    }
  }
}
```

1. Uses GET operations to look at information about storage pool to confirm it has more than 25% available capacity

- **Request:** GET /redfish/v1/StorageServices/1/StoragePools/BasePool

- **Response:**

```
{
  "@Redfish.Copyright": "Copyright 2015-2016 SNIA. All rights reserved.",
  "@odata.context": "/redfish/v1/$metadata#StoragePool.StoragePool",
  "@odata.id": "/redfish/v1/StorageServices/1/StoragePools/BasePool",
  "@odata.type": "#StoragePool.1 0 0.StoragePool",
```

```
--
"Id": "BasePool",
"Name": "BasePool",
"Description": "Base storage pool for this storage service",
"Status": {
"State": "Enabled",
"Health": "OK",
"HealthRollUp": "Degraded"
},
"BlockSizeBytes": 8192,
"Capacity": {
"Data": {
"ConsumedBytes": 824633720832,
"AllocatedBytes": 1099511627776
},
"Metadata": null,
"Snapshot": null
},
"CapacitySources": [
{
"ProvidedCapacity": {
"ConsumedBytes": 70368744177664,
"AllocatedBytes": 140737488355328,
"GuaranteedBytes": 17592186044416,
"ProvisionedBytes": 562949953421312
},
"Links": {
"ClassOfService": {
"@odata.id": "/redfish/v1/StorageServices/1/ClassesOfService/GoldBoston"
},
"ProvidingPool": null,
"ProvidingVolume": null
}
}
],
"LowSpaceWarningThresholdPercent": [
70,
80,
90
],
"Links": {
"AllocatedPools": [],
"AllocatedVolumes": [
{
"@odata.id": "/redfish/v1/StorageServices/1/Volumes/61001234876545676100123487654567"
}
],
"SupportedClassesOfService": [
{
"@odata.id": "/redfish/v1/StorageServices/1/ClassesOfService/GoldBoston"
},

```

```
},
{
  "@odata.id": "/redfish/v1/StorageServices/1/ClassesOfService/SilverBoston"
}
]
}
}
```

2. Use expand action to increase size of storage volume by 25%

- **Request:** PATCH /redfish/v1/StorageServices/1/Volumes/61001234876545676100123487654567
 - **Headers:** No additional headers required
 - **Body:**

```
``json
{
  "Capacity": {
    "Data": {
      "AllocatedBytes": 1374389534720
    }
  }
}
...

```

- **Response:**

```
{
  "taskId": "ExpandTask123"
}
```

Postconditions: The volume's capacity expands by 25%. Administrator needs to track the associated task to know when the volume expansion completes.

See also: None defined.

8.14 Find storage service

Summary: Find storage services that support specified class of service characteristics.

Purpose: Identify storage services that support needed service options.

Who: [CloudAdmin](#), [StorageAdmin](#), [DevOps](#)

Management Domain: [Block storage management](#), [File system storage management](#), [Application storage management](#), [Service catalog management](#)

Trigger: Need to provide a Storage Service reference to support a more specific request (e.g., [Create Volume](#))

Detailed Context: Identify existing storage services that would satisfy a set of service characteristics (e.g., capacity, service level, throughput)

Preconditions: None.

Inputs: A set of storage attributes that must be satisfied, phrased as a [Filter Expression](#). For this example:

- {IOPS}: a desired minimum performance level

Basic Course of Events:

1. Use GET operation to retrieve possible StorageService entries

- **Request:**

```
GET /redfish/v1/StorageServices$expand=Members/Links/ClassesOfService ($filter=DataProtectionLineO:
```

- **Headers:** No additional headers required.
- **Body:** None.

- **Response:**

```
[{
  "@odata.context": "/redfish/v1/$metadata#StorageService.StorageService";
  "@odata.id": "/redfish/v1/StorageServices/1";
  "Name": "My Storage Service";
}]
```

2. Select a Class of Service URL from the returned values.

Postconditions: None defined.

Failure Scenario:

Error Code	Cause
204	No existing Class of Service entries match the required criteria

See also: None defined.

8.15 *Get capacity by class of service*

Summary: Get capacity by class of service

Purpose:

- Provide the amount of storage at a specified class of service.

Who: [CloudAdmin](#), [StorageAdmin](#), [DevOps](#)

Management domain: [Block Storage Management](#), [File system storage management](#), [Application storage management](#), [Service](#)

catalog management

Trigger: An alert indicating low available storage or simple monitoring of available storage.

Detailed context: Need to identify available storage capacity for a specified class of service. Input will be the StorageService URI and a class of service name. The use case will look across all storage pools.

Preconditions: None.

Inputs: The URI for the Storage Service that will provision storage is `/redfish/v1/StorageServices(1)` and the requested class of service name. In this example, `GoldBoston`.

Basic course of events:

1. Use GET operation to retrieve data protection capabilities and supported line of service options.

- **Request:**

GET `/redfish/v1/StorageServices(1)/StoragePools/Members?$expand=Links/ClassesOfService($filter=Nar`

- **Headers:** No additional headers required.
- **Body:** None.

- **Response:**

```
[ {
  "@odata.context": "/redfish/v1/$metadata#StoragePool.StoragePool",
  "@odata.Id": "/redfish/v1/$metadata#StoragePool.StoragePool/BasePool",
  "Name": "BasePool",
  "Capacity": {
    "Data": {
      "ConsumedBytes": 0,
      "AllocatedBytes": 0,
      "GuaranteedBytes": 0,
      "ProvisionedBytes": 0
    },
    "Metadata": null,
    "Snapshot": null
  }
},
{
  "@odata.context": "/redfish/v1/$metadata#StoragePool.StoragePool",
  "@odata.Id": "/redfish/v1/$metadata#StoragePool.StoragePool/SpecialPool",
  "Name": "SpecialPool",
  "Capacity": {
    "Data": {
      "ConsumedBytes": 549755813888,
      "AllocatedBytes": 1099511627776,
      "GuaranteedBytes": 70368744177664,
      "ProvisionedBytes": 140737488355328
    },
    "Metadata": null,
    "Snapshot": null
  }
}
]
```

1. For each returned StoragePool, the requester subtracts ConsumedBytes from ProvisionedBytes for each category , (i.e. Capacity/Data, Capacity/Metadata, and Capacity/Snapshot), and sums the results for each category to get the available capacity for the class of service.

Postconditions: None defined.

Failure Scenario:

Error Code	Cause
204	The addressed StorageService is not found.

See also: None defined.

8.16 List supported line of service options

Summary: List supported line of service options

Purpose:

- To provide the list of currently defined data protection capabilities and line of service options to an administrator with the responsibility of creating new ClassOfService options.

Who: [CloudAdmin](#), [StorageAdmin](#)

Management domain: [Block storage management](#), [File system storage management](#), [Application storage management](#), [Service catalog management](#)

Trigger: Need to define a new a ClassOfService for use in another request (e.g., [Create Volume](#))

Detailed context: The requester has determined that a new class of service is needed. This use case identifies existing service options for specified lines of service and the capabilities to create new line of service options. For this example, the assumption is that the administrator wants to see all Data Protection capabilities and line of service options.

Note: End users will not typically need to get this level of detail.

Preconditions: None.

Inputs: The URI for the Storage Service that will provision storage is `/redfish/v1/StorageServices(1)`.

Basic Course of Events:

1. Use GET operation to retrieve data protection capabilities and supported line of service options.

• **Request:**

GET `/redfish/v1/StorageServices(1)/Links/DataProtectionLoSCapabilities`

- **Headers:** No additional headers required.
- **Body:** None.

• **Response:**

```
{
  "@SSM.Copyright": "Copyright © 2014-2016 SNIA. All rights reserved.",
  "@odata.context": "/redfish/v1/$metadata#DataProtectionLoSCapabilites.DataProtectionLoSCapabilites",
  "@odata.id": "/redfish/v1/StorageServices/1/DataProtectionLoSCapabilites",
  "@odata.type": "#DataProtectionLoSCapabilites_1_0_0.DataProtectionLoSCapabilites",
  "Name": "DataProtectionLoSCapabilites",
  "SupportedRecoveryGeographicObjectives": ["Rack",
  "Row",
  "Server",
  "Facility",
  "AvailabilityZone",
  "Region"],
  "SupportedRecoveryPointObjectiveSeconds": [0,
  4,
  60,
  3600],
  "SupportedRecoveryTimeObjectives": ["Immediate",
```

```

"Online",
"Nearline",
"Offline"],
"SupportedReplicaTypes": ["Mirror",
  "Snapshot",
  "Clone"],
"MinimumLifetimeSeconds": 60,
"SupportsIsolated": true,
"SupportedReplicaOptions": [],
"SupportedLocations": [],
"SupportedDataProtectionLinesOfService": [{
"RecoveryGeographicObjective": "AvailabilityZone",
"RecoveryPointObjectiveSeconds": 604800,
"RecoveryTimeObjective": "Nearline",
"ProtectionType": "Delta",
"MinLifetimeSeconds": 7948800,
"IsIsolated": true,
"Schedule": {
  "EventStartTime": "03:00-05:00",
  "RecurrenceGranularity": "Weekly",
  "RecurrenceInterval": 1,
  "RecurrenceDayOfWeekMask": ["Saturday"]
},
"ReplicaClassOfService": {
  "@odata.id": "/redfish/v1/StorageServices/1/ClassesOfService/Silver_Providence"
},
"ReplicaAccessLocation": {
  "Country": "",
  "Territory": "",
  "State": "",
  "City": "Providence",
  "Address1": "",
  "Address2": "",
  "Address3": "",
  "PostalCode": "",
  "Building": "",
  "Room": "",
  "Row": "",
  "Rack": "",
  "Shelf": "",
  "Item": "",
  "GPSCoords": "",
  "OtherLocationInfo": ""
}
},
{
"RecoveryGeographicObjective": "AvailabilityZone",
"RecoveryPointObjectiveSeconds": 2592000,
"RecoveryTimeObjective": "Nearline",
"ProtectionType": "Full"
}

```



```
"ProtectionType": "Full",
"MinLifetimeSeconds": 10627200,
"IsIsolated": true,
"Schedule": {
  "EventStartTime": "04:00-05:00",
  "RecurrenceGranularity": "Monthly",
  "RecurrenceInterval": "1",
  "RecurrenceDayOfWeekMask": ["Sunday"],
  "RecurrenceDayOfMonthMask": [1,
2,
3,
4,
5,
6,
7],
  "RecurrenceMonthOfYearMask": []
},
"ReplicaClassOfService": {
  "@odata.id": "/redfish/v1/StorageServices/1/ClassesOfService/Storage/Silver_Providence"
},
"ReplicaAccessLocation": {
  "Country": "",
  "Territory": "",
  "State": "",
  "City": "Providence",
  "Address1": "",
  "Address2": "",
  "Address3": "",
  "PostalCode": "",
  "Building": "",
  "Room": "",
  "Row": "",
  "Rack": "",
  "Shelf": "",
  "Item": "",
  "GPSCoords": "",
  "OtherLocationInfo": ""
}
},
{
  "RecoveryGeographicObjective": "AvailabilityZone",
  "RecoveryPointObjectiveSeconds": 0,
  "RecoveryTimeObjective": "Online",
  "ProtectionType": "Continuous",
  "MinLifetimeSeconds": null,
  "IsIsolated": true,
  "Schedule": null,
  "ReplicaClassOfService": {
    "@odata.id": "/redfish/v1/StorageServices/1/ClassesOfService/Storage/Gold_Providence"
  },
}
```

```
"ReplicaAccessLocation": {
  "Country": "",
  "Territory": "",
  "State": "",
  "City": "Providence",
  "Address1": "",
  "Address2": "",
  "Address3": "",
  "PostalCode": "",
  "Building": "",
  "Room": "",
  "Row": "",
  "Rack": "",
  "Shelf": "",
  "Item": "",
  "GPSCoords": "",
  "OtherLocationInfo": ""
},
"Schedule": null
},
{
  "RecoveryGeographicObjective": "AvailabilityZone",
  "RecoveryPointObjectiveSeconds": 4,
  "RecoveryTimeObjective": "Online",
  "ProtectionType": "Continuous",
  "ReplicaAccessLocation": {
    "Country": "",
    "Territory": "",
    "State": "",
    "City": "Dallas",
    "Address1": "",
    "Address2": "",
    "Address3": "",
    "PostalCode": "",
    "Building": "",
    "Room": "",
    "Row": "",
    "Rack": "",
    "Shelf": "",
    "Item": "",
    "GPSCoords": "",
    "OtherLocationInfo": ""
  },
  "Schedule": null,
  "Links": {
    "ReplicaOption": {
      "@odata.id": "/redfish/v1/StorageServices/1/ClassesOfService/Gold_Dallas"
    }
  }
},
```

```
{
  "RecoveryGeographicObjective": "AvailabilityZone",
  "RecoveryPointObjectiveSeconds": 0,
  "RecoveryTimeObjective": "Online",
  "ProtectionType": "Mirror",
  "ReplicaAccessLocation": {
    "Country": "",
    "Territory": "",
    "State": "",
    "City": "Providence",
    "Address1": "",
    "Address2": "",
    "Address3": "",
    "PostalCode": "",
    "Building": "",
    "Room": "",
    "Row": "",
    "Rack": "",
    "Shelf": "",
    "Item": "",
    "GPSCoords": "",
    "OtherLocationInfo": ""
  },
  "Schedule": null,
  "Links": {
    "ReplicaOption": {
      "@odata.id": "/redfish/v1/StorageServices/1/ClassesOfService/Gold_Providence"
    }
  }
},
{
  "RecoveryGeographicObjective": "AvailabilityZone",
  "RecoveryPointObjectiveSeconds": 4,
  "RecoveryTimeObjective": "Online",
  "ProtectionType": "Continuous",
  "ReplicaAccessLocation": {
    "Country": "",
    "Territory": "",
    "State": "",
    "City": "Dallas",
    "Address1": "",
    "Address2": "",
    "Address3": "",
    "PostalCode": "",
    "Building": "",
    "Room": "",
    "Row": "",
    "Rack": "",
    "Shelf": "",
    "Item": ""
  }
}
```

```

    "GPSCoords": "",
    "OtherLocationInfo": ""
  },
  "Schedule": null,
  "Links": {
    "ReplicaOption": {
      "@odata.id": "/redfish/v1/StorageServices/1/ClassesOfService/Gold_Dallas"
    }
  }
},
{
  "RecoveryGeographicObjective": "AvailabilityZone",
  "RecoveryPointObjectiveSeconds": 0,
  "RecoveryTimeObjective": "Online",
  "ProtectionType": "Continuous",
  "MinLifetimeSeconds": null,
  "IsIsolated": true,
  "Schedule": null,
  "ReplicaClassOfService": {
    "@odata.id": "/redfish/v1/StorageServices/1/ClassesOfService/Gold_Cambridge"
  },
  "ReplicaAccessLocation": {
    "Country": "",
    "Territory": "",
    "State": "",
    "City": "Cambridge",
    "Address1": "",
    "Address2": "",
    "Address3": "",
    "PostalCode": "",
    "Building": "",
    "Room": "",
    "Row": "",
    "Rack": "",
    "Shelf": "",
    "Item": "",
    "GPSCoords": "",
    "OtherLocationInfo": ""
  }
},
{
  "RecoveryGeographicObjective": "AvailabilityZone",
  "RecoveryPointObjectiveSeconds": 4,
  "RecoveryTimeObjective": "Nearline",
  "ProtectionType": "Continuous",
  "MinLifetimeSeconds": null,
  "IsIsolated": true,
  "Schedule": null,
  "ReplicaClassOfService": {
    "@odata.id": "/redfish/v1/StorageServices/1/ClassesOfService/Gold_Dallas"
  }
}

```

```

    "odata.id": "/redfish/v1/StorageServices/1/ClassesOfService/Gold_Dallas"
  },
  "ReplicaAccessLocation": {
    "Country": "",
    "Territory": "",
    "State": "",
    "City": "Dallas",
    "Address1": "",
    "Address2": "",
    "Address3": "",
    "PostalCode": "",
    "Building": "",
    "Room": "",
    "Row": "",
    "Rack": "",
    "Shelf": "",
    "Item": "",
    "GPSCoords": "",
    "OtherLocationInfo": ""
  }
}]
}

```

2. Repeat as needed for other lines of service using the service properties:

- DataSecurityLoSCapabilities
- DataStorageLoSCapabilities
- IOConnectivityLoSCapabilities
- IOPerformanceLoSCapabilities.

Postconditions: None defined.

Failure Scenario:

Error Code	Cause
204	The addressed StorageService is not found.

See also: None defined.

8.17 *Protect Volumes of an application*

Summary: Protect Application Volumes

Purpose: Protect the Volumes of an application represented by a storage group with daily clones and hourly snapshots.

Who: [CloudAdmin](#), [StorageAdmin](#), [DevOps](#)

Management Domain: [Block storage management](#)

Triggers: Need to protect a set of volumes.

Detailed Context: A storage group collects a set of related storage entities (volumes or file systems). The primary purpose of the collection is to govern access to the members by clients or to add service requirements for the members of the collection. It is expected that the collected storage is used by a set of related client applications.

In this use case, the goal is to add data protection services to all of the storage members, in the form of hourly snapshots and daily clones. This is done by adding a ClassOfService to the storage group or by updating an existing ClassOfService for the storage group. In this use case, the assumption is that the storage group does not already have a class of service, (but this is checked). The added class of service is additive to the class of service requirements of each member storage entity. If the added requirement are in conflict with the requirements of any member, then the request to add the new class of service shall be rejected.

Preconditions: None.

Inputs:

- **_URL** for storage service: /redfish/v1/StorageServices(1)
- **_URL** for the storage group: /redfish/v1/StorageServices(1)/StorageGroups(1)

Basic Course of Events:

1. The first step is to check to see that the desired ClassOfService for the storage group has not been assigned.
2. **Request:** GET /redfish/v1/StorageServices(1)/StorageGroups(1)/Links/ClassOfService
 - **Headers:** No additional headers required.
 - **Body:**
3. **Response:** 204 No Content
 - **Headers:**
 - **Body:**
4. Add a ClassOfService that meets the requirements to create hourly snapshots and daily clones. This is added to the classes of service supported by the storage service. If the storage service is not able to provide the requested service, this request shall be rejected.
5. **Request:** POST /redfish/v1/StorageServices(1)/Links/ClassesOfService
 - **Headers:** No additional headers required.
 - **Body:** ` "Name": "HourlySnapshotsDailyClone", "Description": "Hourly local snapshots and daily local clone.", "LinesOfService": {

"DataProtectionLineOfService":
[{
"Name": "HourlyLocalSnapshot",
"RecoveryGeographicObjective": "Server",
"RecoveryPointObjective": "PT1H",
"RecoveryTimeObjective": "Nearline",
"ReplicaType": "Snapshot",
"Schedule": {
"Name": "Hourly"
"Lifetime": "P2D"
"MaxOccurrences": 50,

```
    "RecurrenceInterval": "PT1H"
  }
},
{
  "Name": "DailyLocalClone",
  "RecoveryGeographicObjective": "Server",
  "RecoveryPointObjective": "P1D",
  "RecoveryTimeObjective": "Nearline",
  "ReplicaType": "Clone",
  "Schedule": {
    "Name": "Daily"
    "Lifetime": "P2W"
    "MaxOccurrences": 25,
    "RecurrenceInterval": "P1D"
  }
}]
}`
```

6. **Response:**

- **Headers:** Location = /redfish/v1/StorageServices(1)/Links/ClassesOfService(9)
- **Body:**

7. Add the new Class of Service to the storage group. In the process of adding this new class of service the storage service shall check that the new requirements can be met for each member of the storage group. If not, this request shall be rejected.

8. **Request:**

PATCH /redfish/v1/StorageServices(1)/StorageGroups(1)/Links

- **Headers:** No additional headers required.
- **Body:**

```
"ClassOfService": {"OData.Id": "/redfish/v1/StorageServices(1)/Links/ClassesOfService(9)"}
```

9. **Response:**

- **Body:**

Postconditions: The updated storage group is now creating hourly snapshots and daily clones for all volumes or filesystems that are members of that group.

Failure Scenario: None defined.

See also: None defined.

8.18 *Subscribe to Threshold Events*

Summary: Subscribe to Trigger/Clear events for LowSpaceWarningThresholds for a named Volume.

Purpose: Provide an event stream to support utilization management. This is used in conjunction with `LowSpaceWarningThresholds` to provide a means for on-going monitoring of resource consumption.

Who: [CloudAdmin](#), [StorageAdmin](#), [DevOps](#)

Management domain: [Block storage management](#), [Application storage management](#)

Triggers: None defined.

Detailed Context: This provides the basis for monitoring capacity consumption.

Preconditions: None defined.

Inputs:

- `{volume}`: The Volume to be monitored
- `{destination}`: The subscription destination
- `{events}`: An array of events to be subscribed

Basic course of events:

1. Retrieve the event destination

- **Request:** `POST /redfish/v1/EventService/Subscriptions/Members`

- **Headers:** No additional headers required.

- **Body:**

```
json { "@Redfish.Copyright": "Copyright 2016-2017 Storage Networking Industry Association (SN:"
```

- **Response:** 201 - Created

```
{  
  "Location": "/redfish/v1/EventService/Subscriptions/1/Members/1e7da"  
}
```

Postconditions: Newly-created event subscription is added to the `EventService`.

Failure Scenario: None defined.

See also: None defined.

8.19 Set Capacity Usage Threshold

Summary: Set a low space warning threshold on a Volume.

Purpose: Define capacity threshold(s) that will trigger the creation of a `LowSpaceWarningThresholdTriggered` event and/or `LowSpaceWarningThresholdCleared` event against a given Volume.

Who: [CloudAdmin](#), [StorageAdmin](#), [DevOps](#)

Management domain: [Block storage management](#), [Application storage management](#)

Triggers: None defined.

Detailed Context: This provides the basis for monitoring capacity consumption.

Preconditions: Volume exists.

Inputs:

- {volume}: Volume to be monitored
- {threshold}: A array of percentages value(s)

Basic course of events:

1. Get the LowSpaceWarningThresholdPercents array for the selected Volume.

- **Request:**

```
GET /redfish/v1/StorageServices/Simple1/Volumes/V1?$select=LowSpaceWarningThresholdPercents
```

- **Headers:** No additional headers required.
- **Body:** None defined.

2. **Response:**

```
json { "@odata.metadata": "/redfish/v1/$metadata#Volume.Volume/v1?$select=LowSpaceWarningThresh
```

3. Update the values in the array of thresholds

- **Request:** PATCH /redfish/v1/StorageServices/Simple1/Volumes/V1

- **Headers:** ETag: 1abdc7.
- **Body:**

```
json { "value": [ { "LowSpaceWarningThresholdPercents": [0,5,10,35,50] } ] }
```

- **Response:** 200 - OK

- **Body:**

```
json { "@Redfish.Copyright": "Copyright 2015-2016 SNIA. All rights reserved.", "@odata.conte
```

Postconditions: The new threshold values are set for the identified volume.

Failure Scenario: None defined.

See also: None defined.