



Swordfish Scalable Storage Management Error Message and Event Notification Guide

Version: 1.2.9

ABSTRACT:

The Swordfish Scalable Storage Management Message Guide provides a summary of the preferred handling of errors, error messages, and event notifications in a Swordfish implementation

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

SNIA Approved Publication

Table of Contents

USAGE	4
About SNIA	8
Acknowledgements	8
1 Introduction	9
1.1 Document Scope	9
1.2 Documentation structure	9
1.3 Base implementation assumptions	10
1.4 Knowledge assumptions	10
2 HTTP status codes	11
2.1 Overview	11
2.2 Related information	11
3 Message Registries	12
3.1 Base 1.21.0	12
3.2 ResourceEvent 1.4.2	92
3.3 Storage 1.0.0	110
3.4 StorageCoS 1.0.0	114
3.5 StorageConnection 1.0.0	116
3.6 StorageDevice 1.5.0	118
3.7 Swordfish 1.1.0	143
4 Error Messages	148
4.1 Overview	148
4.2 Action Errors	148
4.3 JSON Errors	149
4.4 Property Errors	149
4.5 Resource Errors	150
5 Alphabetic List of Error Cases	152
5.1 Overview	152
5.2 <i>Error Case: Action Not Supported</i>	152
5.3 <i>Error Case: Action Parameter Duplicate</i>	153
5.4 <i>Error Case: Action Parameter Missing</i>	154
5.5 <i>Error Case: Action Parameter Not Supported</i>	155
5.6 <i>Error Case: Action Parameter Unknown</i>	156
5.7 <i>Error Case: Action Parameter Value Error</i>	157
5.8 <i>Error Case: Action Parameter Value Not In List Error</i>	158
5.9 <i>Error Case: Action Parameter Value Type Error</i>	159
5.10 <i>Error Case: Empty JSON</i>	160
5.11 <i>Error Case: Invalid JSON</i>	162
5.12 <i>Error Case: Malformed JSON</i>	163
5.13 <i>Error Case: NoOperation</i>	164
5.14 <i>Error Case: Property Duplicate</i>	165
5.15 <i>Error Case: Property Missing</i>	166
5.16 <i>Error Case: Property Not Updated</i>	167
5.17 <i>Error Case: Property Not Writable</i>	168
5.18 <i>Error Case: Property Unknown</i>	169
5.19 <i>Error Case: Property Value Conflict</i>	170
5.20 <i>Error Case: Property Value Not In List</i>	171

5.21 Error Case: Property Value Out of Range	172
5.22 Error Case: Property Value Resource Conflict	173
5.23 Error Case: Property Value Type Error.....	175
5.24 Error Case: Resource Already Exists	176
5.25 Error Case: Resource At URI In Unknown Format	177
5.26 Error Case: Resource At URI Unauthorized – WIP	178
5.27 Error Case: Resource Cannot Be Deleted	179
5.28 Error Case: Resource Creation Conflict	180
5.29 Error Case: Resource Exhaustion	181
5.30 Error Case: Resource In Standby	183
5.31 Error Case: Resource In Use.	185
5.32 Error Case: Resource Missing At URI	186
5.33 Error Case: Resource Not Found.....	187
5.34 Error Case: Resource Type Incompatible	188
5.35 Error Case: Unrecognized Request Body	189
6 Event Notifications	191

List of Tables

Table 1: Revision History	7
Table 2: Contributors	8
Table 3: Guidelines for the Use Case Template	9
Table 4: Base messages	12
Table 5: ResourceEvent messages	92
Table 6: Storage messages	110
Table 7: StorageCoS messages	114
Table 8: StorageConnection messages	116
Table 9: StorageDevice messages	118
Table 10: Swordfish messages	143
Table 11: General Errors	148
Table 12: Action errors	148
Table 13: JSON Errors	149
Table 14: Property errors	149
Table 15: Resource errors	150

USAGE

Copyright (c) 2016 - 2026 Storage Networking Industry Association. All rights reserved. All other trademarks or registered trademarks are the property of their respective owners.

Storage Networking Industry Association (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 SNIA copyright on that material, and must credit 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 (c) 2026, 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 Storage Networking Industry Association nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

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. SNIA makes no warranty of any kind with regard to this publication, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The SNIA shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use.

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

Current Revision

SNIA is actively engaged in expanding and refining the Swordfish documentation. The most current revision can be found on the SNIA web site at

https://www.snia.org/tech_activities/standards/curr_standards/swordfish.

Contact SNIA

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

FEEDBACK AND INTERPRETATIONS

Requests for interpretation, suggestions for improvement and addenda, or defect reports are welcome. They should be sent via the SNIA Feedback Portal at <http://www.snia.org/feedback/> or by mail to SNIA, 5201 Great America Parkway, Suite 320, Santa Clara, CA 95054, USA.

INTENDED AUDIENCE

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

VERSIONING POLICY

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

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

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

Errata Number: Versioned material having version number ‘v’, a release number ‘r’, and an errata number ‘e’ should be backwards compatible with previous revisions of the material with the same version number and release number (“errata versions”). An errata revision of versioned material is limited to minor corrections or clarifications of existing versioned material. An errata revision may be backwards

incompatible, if the incompatibility is necessary for correct operation of implementations of the versioned material.

Revision History

The evolution of this document is summarized in Table 1.

Table 1: Revision History

Date	Revision	Notes
29 May 2020	1.2.0	Initial Release
18 August 2020	1.2.1	Minor editorial corrections
31 October 2020	1.2.1c	Released as SNIA Approved Publication
2 March 2021	1.2.2	Release as Working Draft
30 August 2021	1.2.3	Add entry for new message type (ResourceInUse)
5 December 2021	1.2.3	Release as SNIA Approved Publication
12 April 2022	1.2.4	Release as Working Draft
12 July 2022	1.2.4a	Release as SNIA Standard.
16 March 2023	1.2.5	Release as Working Draft
20 June 2023	1.2.5a	Release as SNIA Standard
22 January 2024	1.2.6	Release as Working Draft
9 April 2024	1.2.6	Release as SNIA Standard
21 May 2024	1.2.7	Release as Working Draft
13 August 2024	1.2.7	Release as SNIA Standard
28 January 2025	1.2.8	Release as Working Draft
22 May 2025	1.2.8	Release as SNIA Standard
26 January 2026	1.2.9	Release as Working Draft
		Updates:
		- Renamed to Swordfish Error Message and Event Notification Guide
		- Added message registry details
		- Updated message registry set and registry naming
		- Expanded message set to improve alignment with Redfish errors

Date	Revision	Notes
		- Added registry supplements to tie registries to usage examples

About SNIA

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

Acknowledgements

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

Table 2: Contributors

Member	Representatives
Broadcom, Inc.	Richelle Ahlvers
Cisco Systems, Inc.	Krishnakumar Gowravaram
Hewlett Packard Enterprise	Chris Lionetti
NetApp, Inc.	Don Deel

1 Introduction

1.1 Document Scope

This guide is intended to provide a common repository of best practices, common tasks and education for handling error conditions and event notifications in a Swordfish implementation. In particular, this document summarizes the Swordfish-related messages and message registries that, in conjunction with the information defined in Redfish, define the messages that can be returned in response to a RESTful request to the Swordfish API (i.e., a error message) or from the event service (i.e., an event notification).

1.2 Documentation structure

This document is intended to be a comprehensive source for the message registries used within Swordfish and the error messages and notifications that they define. The messages from each registry are explained in detail in Clause 3 (“Message Registries”). The document then provides a set of scenarios that capture common situations and best practices, using many (but not all) of the messages found in Clause 3. These usage examples are intended to promote the consistent handling of errors, error messaging, and event notification in a Swordfish implementations.

Each usage scenario uses 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 error
Summary	A high-level summary of the error
Message Group	One of GeneralErrors, ActionErrors, JSONErrors, PropertyErrors, ResourceErrors, or EventNotifications
Common Cause	A typical cause for generation of this message
Example	The specific example that will be used to illustrate proper usage of the message
Basic Course of Events	A sequence of API requests (including required headers, the body of the request, and the expected reply), or events that cause the message to be issued

Name	Description
Additional Context	Clarifying material, and additional detail intended to clarify subtleties of the error case or to highlight additional response options.

1.3 Base implementation assumptions

This document assumes that some fundamental configuration issues have been properly implemented, 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, including an EventService
- Any required login credentials

1.4 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 HTTP standards: [Wikipedia](#)
- For Redfish standards: [Redfish Specification](#)
- For Swordfish standards: [Swordfish Specification](#)
- For Swordfish API tutorials: [Swordfish Tutorials](#)

2 HTTP status codes

2.1 Overview

The HTTP status codes are defined by RFC2616 by W3.org, and are intended to address a broad range of HTTP implementations. Both the Redfish specification and the Swordfish specification provide information about usage for a subset of HTTP status codes. In addition, the server can return extended status information as a simple JSON object to further clarify the handling and outcome of a particular API request; guidance on when to use extended status and error information is also specified in the Redfish and Swordfish Specifications.

While Swordfish clients may receive any of the standard HTTP status codes, the Redfish and Swordfish Specifications include an explicit list that must be supported. In addition, as this subset of HTTP codes provides a detailed mapping from generic HTTP status codes to domain-specific situations and probable causes, they should be the most common and as the only required status codes, implementations should target their use exclusively as much as possible. This enables clients to implement with little to no vendor-specific instrumentation.

2.2 Related information

For more information, see:

- The [Swordfish Specification](#)
- The [Redfish Specification](#)
- The [HTTP Protocol definition of HTTP status codes](#).

3 Message Registries

3.1 Base 1.21.0

3.1.1 Overview

This registry defines the base messages for Redfish.

The messages defined in this version of the Base registry are summarized in Table 4.

Table 4: Base messages

Message	Severity	Description
AccessDenied	Critical	Indicates that while attempting to access, connect to, or transfer to or from another resource, the service denied access.
AccessForbidden	Critical	Indicates that the request contained valid credentials, but the user does not have the necessary privileges to perform the requested operation.
AccessUnauthorized	Critical	Indicates that the request did not contain valid credentials.
AccountForSessionNoLongerExists	OK	Indicates that the account for the session was removed, and so the session was removed as well.
AccountModified	OK	Indicates that the account was successfully modified.
AccountNotModified	Warning	Indicates that the modification requested for

Message	Severity	Description
		the account was not successful.
AccountRemoved	OK	Indicates that the account was successfully removed.
ActionDeprecated	Warning	Indicates the action is deprecated.
ActionNotSupported	Critical	Indicates that the action supplied with the POST operation is not supported by the resource.
ActionParameterDuplicate	Warning	Indicates that the action was supplied with a duplicated action parameter in the request body.
ActionParameterMissing	Critical	Indicates that the action requested was missing an action parameter that is required to process the action.
ActionParameterNotSupported	Warning	Indicates that the parameter supplied for the action is not supported on the resource.
ActionParameterUnknown	Warning	Indicates that an action was submitted but an action parameter supplied did not match any of the known parameters.
ActionParameterValueConflict	Warning	Indicates that the requested parameter value could not be completed because of a mismatch with other parameters or properties in the resource.
ActionParameterValueError	Warning	Indicates that a parameter was given an invalid value.
ActionParameterValueFormatError	Warning	Indicates that a parameter was given the correct value type but the format of the value is not supported.

Message	Severity	Description
ActionParameterValueNotInList	Warning	Indicates that a parameter was given the correct value type but the value of that parameter was not supported. The value is not in an enumeration.
ActionParameterValueOutOfRange	Warning	Indicates that a parameter was given the correct value type but the value of that parameter is outside the supported range.
ActionParameterValueTypeError	Warning	Indicates that a parameter was given the wrong value type, such as when a number is supplied for a parameter that requires a string.
ArraySizeTooLong	Warning	Indicates that the size of the array exceeded the maximum number of elements.
ArraySizeTooShort	Warning	Indicates that the size of the array is less than the minimum number of elements.
AuthenticationTokenRequired	Critical	Indicates that the request could not be performed because an authentication token was not provided.
ChassisPowerStateOffRequired	Warning	Indicates that the request requires a specified chassis to be powered off.
ChassisPowerStateOnRequired	Warning	Indicates that the request requires a specified chassis to be powered on.
ConditionInRelatedResource	Warning	Indicates that one or more fault or error conditions exist in a related resource.
CouldNotEstablishConnection	Critical	Indicates that the attempt to access the resource, file, or image at the URI was

Message	Severity	Description
		unsuccessful because a session could not be established.
CreateFailedMissingReqProperties (Deprecated)	Critical	Indicates that a create operation was attempted on a resource but that properties that are required for the create operation were missing from the request.
CreateLimitReachedForResource	Critical	Indicates that no more resources can be created on the resource as it has reached its create limit.
Created	OK	Indicates that all conditions of a successful create operation were met.
EmptyJSON	Warning	Indicates that the request body contained an empty JSON object when one or more properties are expected in the body.
EventBufferExceeded	Warning	Indicates undelivered events may have been lost due to a lack of buffer space in the service.
EventSubscriptionLimitExceeded	Critical	Indicates that an event subscription establishment was requested but the operation failed due to the number of simultaneous connections exceeding the limit of the implementation.
GeneralError	Critical	Indicates that a general error has occurred. Use in <code>@Message.ExtendedInfo</code> is discouraged. When used in <code>@Message.ExtendedInfo</code> , implementations are expected to include a

Message	Severity	Description
		Resolution property with this message and provide a service-defined resolution to indicate how to resolve the error.
GenerateSecretKeyRequired	Critical	Indicates that the Time-based One-Time Password (TOTP) secret key needs to be generated for the account before accessing the service. The secret key can be generated with a POST to the <code>GenerateSecretKey</code> action for the <code>ManagerAccount</code> resource instance.
HeaderInvalid	Critical	Indicates that a request header is invalid.
HeaderMissing	Critical	Indicates that a required request header is missing.
InsufficientPrivilege	Critical	Indicates that the credentials associated with the established session do not have sufficient privileges for the requested operation.
InsufficientStorage	Critical	Indicates that the operation could not be completed due to a lack of storage or memory available to the service.
InternalError	Critical	Indicates that the request failed for an unknown internal error but that the service is still operational.
InvalidIndex	Warning	The index is not valid.
InvalidJSON	Critical	Indicates that the request body contains invalid JSON.
InvalidObject	Critical	Indicates that the object in question is invalid according to the implementation. An

Message	Severity	Description
		example is a firmware update malformed URI.
InvalidURI	Critical	Indicates that the operation encountered a URI that does not correspond to a valid resource.
LicenseRequired	Critical	Indicates that a license is required to perform the requested operation.
MalformedJSON	Critical	Indicates that the request body was malformed JSON.
MaximumErrorsExceeded	Critical	Indicates that so many errors have occurred that the reporting service cannot return them all.
MissingOrMalformedPart	Critical	Indicates that a multipart request is missing a required part or contains malformed parts.
NetworkNameResolutionNotConfigured	Warning	Indicates that network-based name resolution is not configured on the service.
NetworkNameResolutionNotSupported	Warning	Indicates the service does not support network-based name resolution.
NoOperation	Warning	Indicates that the requested operation will not perform any changes on the service.
NoValidSession	Critical	Indicates that the operation failed because a valid session is required in order to access any resources.
OneTimePasscodeSent	OK	Indicates that a required one-time passcode was sent to the user. The code should be provided as the Token property in the request to create a session.
OperationFailed	Warning	Indicates that one of the

Message	Severity	Description
		internal operations necessary to complete the request failed. An example of this is when an internal service provider is unable to complete the request, such as in aggregation or RDE.
OperationNotAllowed	Critical	Indicates that the HTTP method in the request is not allowed on this resource.
OperationTimeout	Warning	Indicates that one of the internal operations necessary to complete the request timed out. An example of this is when an internal service provider is unable to complete the request, such as in aggregation or RDE.
PasswordChangeRequired	Critical	Indicates that the password for the account provided must be changed before accessing the service. The password can be changed with a PATCH to the Password property in the manager account resource instance. Implementations that provide a default password for an account may require a password change prior to first access to the service.
PasswordComplexityNotMet	Critical	Indicates that the password for the account provided does not meet the password complexity requirements of the service. The password for the account was not changed.
PasswordIncorrectLength	Critical	Indicates that the password

Message	Severity	Description
		for the account provided does not meet the password length requirements of the service. The password for the account was not changed.
PasswordReuseTooRecent	Critical	Indicates that the password for the account provided does not meet the password reuse requirements of the service. The password for the account was not changed.
PayloadTooLarge	Critical	Indicates that the supplied payload is too large to be accepted by the service.
PreconditionFailed	Critical	Indicates that the ETag supplied did not match the current ETag of the resource.
PreconditionRequired	Critical	Indicates that the request did not provide the required precondition such as an If-Match or If-None-Match header or @odata.etag annotations.
PropertyDeprecated	Warning	Indicates the property is deprecated.
PropertyDuplicate	Warning	Indicates that a duplicate property was included in the request body.
PropertyMissing	Warning	Indicates that a required property was not supplied as part of the request.
PropertyModified	OK	Indicates that all properties listed in RelatedProperties in the message were successfully modified.
PropertyNotUpdated	Critical	Indicates that a property was not updated due to an internal service error, but the service is still functional.

Message	Severity	Description
PropertyNotWritable	Warning	Indicates that a property was given a value in the request body, but the property is a read-only property.
PropertyUnknown	Warning	Indicates that an unknown property was included in the request body.
PropertyValueConflict	Warning	Indicates that the requested write of a property value could not be completed because of a conflict with another property value.
PropertyValueDeprecated	Warning	Indicates that a property was given a deprecated value.
PropertyValueError	Warning	Indicates that a property was given an invalid value.
PropertyValueExternalConflict	Warning	Indicates that the requested write of a property value could not be completed due to the current state or configuration of the resource. This can include configuration conflicts with other resources or parameters that are not exposed by this interface.
PropertyValueFormatError	Warning	Indicates that a property was given the correct value type but the format of the value is not supported.
PropertyValueIncorrect	Warning	Indicates that the requested write of a property value could not be completed because of an incorrect value of the property. Examples include values that do not match a regular expression requirement, passwords that do not match the implementation constraints, or specifying an invalid URI

Message	Severity	Description
		in a reference property.
PropertyValueModified	Warning	Indicates that a property was given the correct value type but the value of that property was modified. Examples are truncated or rounded values.
PropertyValueNotInList	Warning	Indicates that a property was given the correct value type but the value of that property was not supported. The value is not in an enumeration.
PropertyValueOutOfRange	Warning	Indicates that a property was given the correct value type but the value of that property is outside the supported range.
PropertyValueResourceConflict	Warning	Indicates that the requested write of a property value could not be completed due to the current state or configuration of another resource.
PropertyValueTypeError	Warning	Indicates that a property was given the wrong value type, such as when a number is supplied for a property that requires a string.
QueryCombinationInvalid	Warning	Indicates the request contains multiple query parameters and that two or more of them cannot be used together.
QueryNotSupported	Warning	Indicates that query is not supported on the implementation.
QueryNotSupportedOnOperation	Warning	Indicates that query is not supported with the given operation, such as when the

Message	Severity	Description
		\$expand query is attempted with a PATCH operation.
QueryNotSupportedOnResource	Warning	Indicates that query is not supported on the given resource, such as when the \$skip query is attempted on a resource that is not a collection.
QueryParameterOutOfRange	Warning	Indicates that a query parameter was provided that is out of range for the given resource. This can happen with values that are too low or that exceed what is possible for the supplied resource, such as when a page is requested that is beyond the last page.
QueryParameterUnsupported	Warning	Indicates that a query parameter is not supported.
QueryParameterValueError	Warning	Indicates that a query parameter was given an invalid value.
QueryParameterValueFormatError	Warning	Indicates that a query parameter was given the correct value type but the format of the value is not supported.
QueryParameterValueTypeError	Warning	Indicates that a query parameter was given the wrong value type, such as when a number is supplied for a query parameter that requires a string.
ResetRecommended	Warning	Indicates that a component reset is recommended for error recovery while unaffected applications can continue running without any effect on accuracy and

Message	Severity	Description
		performance.
ResetRequired	Warning	Indicates that a component reset is required for changes, error recovery, or operations to complete.
ResourceAlreadyExists	Critical	Indicates that a resource change or creation was attempted but that the operation cannot proceed because the resource already exists.
ResourceAtUriInUnknownFormat	Critical	Indicates that the URI was valid but the resource or image at that URI was in a format not supported by the service.
ResourceAtUriUnauthorized	Critical	Indicates that the attempt to access the resource, file, or image at the URI was unauthorized.
ResourceCannotBeDeleted	Critical	Indicates that a delete operation was attempted on a resource that cannot be deleted.
ResourceCreationConflict	Warning	Indicates that the requested resource creation could not be completed because the service has a resource that conflicts with the request.
ResourceDeprecated	Warning	Indicates the resource is deprecated.
ResourceExhaustion	Critical	Indicates that a resource could not satisfy the request due to some unavailability of resources. An example is that available capacity was allocated.
ResourceInStandby	Critical	Indicates that the request could not be performed because the resource is in

Message	Severity	Description
		standby.
ResourceInUse	Warning	Indicates that a change was requested to a resource but the change was rejected due to the resource being in use or transition.
ResourceMissingAtURI	Critical	Indicates that the operation expected an image or other resource at the provided URI but none was found. Examples of this are in requests that require URIs such as firmware updates.
ResourceNotFound	Critical	Indicates that the operation expected a resource identifier that corresponds to an existing resource but one was not found.
ResourceTypeIncompatible	Critical	Indicates that the resource type of the operation does not match that of the operation destination. Examples of this are a POST to a resource collection using the wrong resource type, an update where the <code>@odata.type</code> properties do not match, or a case of major version incompatibility.
RestrictedPrivilege	Warning	Indicates that the operation was not successful because a privilege is restricted.
RestrictedRole	Warning	Indicates that the operation was not successful because the role is restricted.
ServiceDisabled	Warning	Indicates that the operation failed because the service, such as the account service, is disabled and cannot accept requests.

Message	Severity	Description
ServiceInUnknownState	Critical	Indicates that the operation failed because the service is in an unknown state and cannot accept additional requests.
ServiceShuttingDown	Critical	Indicates that the operation failed because the service is shutting down, such as when the service reboots.
ServiceTemporarilyUnavailable	Critical	Indicates the service is temporarily unavailable.
SessionLimitExceeded	Critical	Indicates that a session establishment was requested but the operation failed due to the number of simultaneous sessions exceeding the limit of the implementation.
SessionTerminated	OK	Indicates that the <code>DELETE</code> operation on the session resource resulted in the successful termination of the session.
SourceDoesNotSupportProtocol	Critical	Indicates that while attempting to access, connect to, or transfer a resource, file, or image from another location that the other end of the connection did not support the protocol.
StrictAccountTypes	Warning	Indicates the request failed because a set of <code>AccountTypes</code> or <code>OEMAccountTypes</code> was not accepted while <code>StrictAccountTypes</code> is set to <code>true</code> .
StringValueTooLong	Warning	Indicates that a string value passed to the given resource was longer than the

Message	Severity	Description
		maximum allowed length. An example is when an implementation has imposed a shorter maximum length than that allowed by the specification.
StringValueTooShort	Warning	Indicates that a string value passed to the given resource was shorter than the minimum required length. An example is when an implementation has imposed a greater minimum length than that required by the specification.
SubscriptionTerminated	OK	An event subscription was terminated by the service. No further events will be delivered.
Success	OK	Indicates that all conditions of a successful operation were met.
UndeterminedFault	Critical	Indicates that a fault or error condition exists but the source of the fault cannot be determined or is unknown to the service.
UnrecognizedRequestBody	Warning	Indicates that the service encountered an unrecognizable request body that could not even be interpreted as malformed JSON.

3.1.2 AccessDenied

Indicates that while attempting to access, connect to, or transfer to or from another resource, the service denied access.

- This message shall indicate that while attempting to access, connect to, or transfer to or from another resource, the service denied access.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Critical

Resolution: Attempt to ensure that the URI is correct and that the service has the appropriate credentials.

Message and Arguments: “While attempting to establish a connection to ”, the service denied access.”

1. *string*: The URI provided.
 - This argument shall contain the URI provided for this operation.

Sample Usage: None

3.1.3 AccessForbidden

Indicates that the request contained valid credentials, but the user does not have the necessary privileges to perform the requested operation.

- This message shall indicate that the request contained valid credentials, but the user does not have the necessary privileges to perform the requested operation.

Version Added: 1.20.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Critical

Resolution: Resubmit the request with credentials that contain the necessary privileges to perform the requested operation.

Message and Arguments: “Forbidden.”

Sample Usage: None

3.1.4 AccessUnauthorized

Indicates that the request did not contain valid credentials.

- This message shall indicate that the request did not contain valid credentials, such as those found in the `Authorization` and `X-Auth-Token` HTTP request headers.

Version Added: 1.20.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Critical

Resolution: Resubmit the request with valid credentials.

Message and Arguments: “Unauthorized.”

Sample Usage: None

3.1.5 AccountForSessionNoLongerExists

Indicates that the account for the session was removed, and so the session was removed as well.

- This message shall indicate that the account for the session was removed, and so the session was removed as well.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: OK

Resolution: Attempt to connect with a valid account.

Message and Arguments: “The account for the current session was removed, and so the current session was removed as well.”

Sample Usage: None

3.1.6 AccountModified

Indicates that the account was successfully modified.

- This message shall indicate that the account was successfully modified.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: OK

Resolution: None.

Message and Arguments: “The account was successfully modified.”

Sample Usage: None

3.1.7 AccountNotModified

Indicates that the modification requested for the account was not successful.

- This message shall indicate that the modification requested for the account was not successful.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: The modification may have failed due to permission issues or issues with the request body.

Message and Arguments: “The account modification request failed.”

Sample Usage: None

3.1.8 AccountRemoved

Indicates that the account was successfully removed.

- This message shall indicate that the account was successfully removed.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: OK

Resolution: None.

Message and Arguments: “The account was successfully removed.”

Sample Usage: None

3.1.9 ActionDeprecated

Indicates the action is deprecated.

- This message shall indicate that the action is deprecated.

Version Added: 1.10.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: Refer to the schema guide for more information.

Message and Arguments: “The action is deprecated.”

1. *string*: The name of the action.
 - This argument shall contain the name of the action that is deprecated.

Sample Usage: None

3.1.10 ActionNotSupported

Indicates that the action supplied with the `POST` operation is not supported by the resource.

- This message shall indicate that the action supplied with the `POST` operation is not supported by the resource.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Critical

Resolution: Check the Actions property in the resource for the supported actions.

Message and Arguments: “The action is not supported by the resource.”

1. *string*: The name of the action.
 - This argument shall contain the name of the action.

Sample Usage:

- [Error Case: Action Not Supported](#)

3.1.11 ActionParameterDuplicate

Indicates that the action was supplied with a duplicated action parameter in the request body.

- This message shall indicate that the action was supplied with a duplicated action parameter in the request body.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: Resubmit the action with only one instance of the action parameter in the request body if the operation failed.

Message and Arguments: “The action was submitted with more than one value for the parameter .”

1. *string*: The name of the action.
 - This argument shall contain the name of the action.
2. *string*: The name of the action parameter.
 - This argument shall contain the name of the action parameter.

Sample Usage:

- [Error Case: Action Parameter Duplicate](#)

3.1.12 ActionParameterMissing

Indicates that the action requested was missing an action parameter that is required to process the action.

- This message shall indicate that the action requested was missing an action parameter that is required to process the action.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Critical

Resolution: Supply the action with the required parameter in the request body when the request is

resubmitted.

Message and Arguments: “The action requires the parameter to be present in the request body.”

1. *string*: The name of the action.
 - This argument shall contain the name of the action.
2. *string*: The name of the action parameter.
 - This argument shall contain the name of the action parameter.

Sample Usage:

- [Error Case: Action Parameter Missing](#)

3.1.13 ActionParameterNotSupported

Indicates that the parameter supplied for the action is not supported on the resource.

- This message shall indicate that the parameter supplied for the action is not supported on the resource.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: Remove the parameter supplied and resubmit the request if the operation failed.

Message and Arguments: “The parameter for the action is not supported on the target resource.”

1. *string*: The name of the action parameter.
 - This argument shall contain the name of the action parameter.
2. *string*: The name of the action.
 - This argument shall contain the name of the action.

Sample Usage:

- [Error Case: Action Parameter Not Supported](#)

3.1.14 ActionParameterUnknown

Indicates that an action was submitted but an action parameter supplied did not match any of the known parameters.

- This message shall indicate that an action was submitted but an action parameter supplied did

not match any of the known parameters.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: Correct the invalid action parameter and resubmit the request if the operation failed.

Message and Arguments: “The action was submitted with the invalid parameter .”

1. *string*: The name of the action.
 - This argument shall contain the name of the action.
2. *string*: The name of the action parameter.
 - This argument shall contain the name of the action parameter.

Sample Usage:

- [Error Case: Action Parameter Unknown](#)

3.1.15 ActionParameterValueConflict

Indicates that the requested parameter value could not be completed because of a mismatch with other parameters or properties in the resource.

- This message shall indicate that the requested parameter value could not be completed because of a mismatch with other parameters or properties in the resource.

Version Added: 1.12.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: None.

Message and Arguments: “The parameter “ with the requested value of “ does not meet the constraints of the implementation.”

1. *string*: The name of the parameter for which a write was requested.
 - This argument shall contain the name of the parameter provided in the action request.
2. *string*: The value of the parameter with which there is a conflict.

- This argument shall contain the value of the parameter with which there is a conflict.

Sample Usage: None

3.1.16 ActionParameterValueError

Indicates that a parameter was given an invalid value.

- This message shall indicate that a parameter was given an invalid value, and the value cannot or should not be included in the message.

Version Added: 1.11.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: Correct the value for the parameter in the request body and resubmit the request if the operation failed.

Message and Arguments: “The value for the parameter in the action is invalid.”

1. *string*: The name of the action parameter.
 - This argument shall contain the name of the action parameter.
2. *string*: The name of the action.
 - This argument shall contain the name of the action.

Sample Usage:

- [Error Case: Action Parameter Value Error](#)

3.1.17 ActionParameterValueFormatError

Indicates that a parameter was given the correct value type but the format of the value is not supported.

- This message shall indicate that a parameter was given the correct value type but the format of the value is not supported.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: Correct the value for the parameter in the request body and resubmit the request if the operation failed.

Message and Arguments: “The value ” for the parameter in the action is not a format that the parameter can accept.”

1. *string*: The value provided for the action parameter.
 - This argument shall contain the value provided for the action parameter. Numeric values shall be converted to a string, and null values shall be represented by the string `null`.
2. *string*: The name of the action parameter.
 - This argument shall contain the name of the action parameter.
3. *string*: The name of the action.
 - This argument shall contain the name of the action.

Sample Usage: None

3.1.18 ActionParameterValueNotInList

Indicates that a parameter was given the correct value type but the value of that parameter was not supported. The value is not in an enumeration.

- This message shall indicate that a parameter was given the correct value type but the value of that parameter was not supported.

Version Added: 1.10.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: Choose a value from the enumeration list that the implementation can support and resubmit the request if the operation failed.

Message and Arguments: “The value ” for the parameter in the action is not in the list of acceptable values.”

1. *string*: The value provided for the action parameter.
 - This argument shall contain the value provided for the action parameter.
2. *string*: The name of the action parameter.
 - This argument shall contain the name of the action parameter.
3. *string*: The name of the action.
 - This argument shall contain the name of the action.

Sample Usage: None

3.1.19 ActionParameterValueOutOfRange

Indicates that a parameter was given the correct value type but the value of that parameter is outside the supported range.

- This value shall indicate that a parameter was given the correct value type but the value of that parameter is outside the supported range.

Version Added: 1.18.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: Correct the value for the parameter in the request body and resubmit the request if the operation failed.

Message and Arguments: “The value ” for the parameter in the action is not in the supported range of acceptable values.”

1. *string*: The value provided for the parameter.
 - This argument shall contain the value provided for the parameter.
2. *string*: The name of the parameter.
 - This argument shall contain the name of the parameter.
3. *string*: The name of the action.
 - This argument shall contain the name of the action.

Sample Usage: None

3.1.20 ActionParameterValueTypeError

Indicates that a parameter was given the wrong value type, such as when a number is supplied for a parameter that requires a string.

- This message shall indicate that a parameter was given the wrong value type, such as when a number is supplied for a parameter that requires a string.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: Correct the value for the parameter in the request body and resubmit the request if the operation failed.

Message and Arguments: “The value ” for the parameter in the action is not a type that the parameter can accept.”

1. *string*: The value provided for the action parameter.
 - This argument shall contain the value provided for the action parameter. Numeric values shall be converted to a string, and null values shall be represented by the string `null`.
2. *string*: The name of the action parameter.
 - This argument shall contain the name of the action parameter.
3. *string*: The name of the action.
 - This argument shall contain the name of the action.

Sample Usage:

- [Error Case: Action Parameter Value Type Error](#)

3.1.21 ArraySizeTooLong

Indicates that the size of the array exceeded the maximum number of elements.

- This message shall indicate that the size of the array exceeded the maximum number of elements.

Version Added: 1.16.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: Resubmit the request with an appropriate array size.

Message and Arguments: “The array provided for property exceeds the size limit .”

1. *string*: The name of the property.
 - This argument shall contain the name of the property.
2. *number*: The maximum number of elements supported for the array.
 - This argument shall contain the maximum number of elements supported for the array for this property.

Sample Usage: None

3.1.22 ArraySizeTooShort

Indicates that the size of the array is less than the minimum number of elements.

- This message shall indicate that the size of the array is less than the minimum number of elements.

Version Added: 1.16.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: Resubmit the request with an appropriate array size.

Message and Arguments: “The array provided for property is under the minimum size limit .”

1. *string*: The name of the property.
 - This argument shall contain the name of the property.
2. *number*: The minimum number of elements supported for the array.
 - This argument shall contain the minimum number of elements supported for the array for this property.

Sample Usage: None

3.1.23 AuthenticationTokenRequired

Indicates that the request could not be performed because an authentication token was not provided.

- This message shall indicate that the request could not be performed because the user provided in the request requires multifactor authentication and an authentication token was not provided.

Version Added: 1.15.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Critical

Resolution: Obtain an authentication token and resubmit the request.

Message and Arguments: “The request could not be performed because an authentication token was not provided.”

Sample Usage: None

3.1.24 ChassisPowerStateOffRequired

Indicates that the request requires a specified chassis to be powered off.

- This message shall indicate that the request requires the specified `Chassis` resource to be powered off as indicated by its `PowerState` property.

Version Added: 1.6.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: Power off the specified chassis and resubmit the request.

Message and Arguments: “The chassis with Id ” is required to be powered off to perform this request.”

1. *string*: The `Id` of the chassis that is required to be powered off.
 - This argument shall contain the value of the `Id` property of the `Chassis` resource that is required to be powered off.

Sample Usage: None

3.1.25 ChassisPowerStateOnRequired

Indicates that the request requires a specified chassis to be powered on.

- This message shall indicate that the request requires the specified `Chassis` resource to be powered on as indicated by its `PowerState` property.

Version Added: 1.6.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: Power on the specified chassis and resubmit the request.

Message and Arguments: “The chassis with Id ” is required to be powered on to perform this request.”

1. *string*: The `Id` of the chassis that is required to be powered on.

- This argument shall contain the value of the `Id` property of the `Chassis` resource that is required to be powered on.

Sample Usage: None

3.1.26 ConditionInRelatedResource

Indicates that one or more fault or error conditions exist in a related resource.

- This message shall indicate that one or more conditions that require attention exist in a resource that affects the `Health` or `HealthRollup` of this resource. This message can be used in place of multiple messages in a `Conditions` property to roll up conditions from a single resource. The resource can be located using the `OriginOfCondition` property. The value of the `MessageSeverity` property should match the highest severity of any conditions in the resource referenced by the `OriginOfCondition` property.

Version Added: 1.9.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: Check the `Conditions` array in the resource shown in the `OriginOfCondition` property to determine the conditions that need attention.

Message and Arguments: “One or more conditions exist in a related resource. See the `OriginOfCondition` property.”

Sample Usage:

- [Error Case: Condition In Related Resource](#)

3.1.27 CouldNotEstablishConnection

Indicates that the attempt to access the resource, file, or image at the URI was unsuccessful because a session could not be established.

- This message shall indicate that the attempt to access the resource, file, or image at the URI was unsuccessful because a session could not be established.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Critical

Resolution: Ensure that the URI contains a valid and reachable node name, protocol information, and other URI components.

Message and Arguments: “The service failed to establish a connection with the URI “.”

1. *string*: The URI provided.
 - This argument shall contain the URI provided for this operation.

Sample Usage: None

3.1.28 CreateFailedMissingReqProperties (Deprecated)

Indicates that a create operation was attempted on a resource but that properties that are required for the create operation were missing from the request.

- This message shall indicate that a create operation was attempted on a resource but that properties that are required for the create operation were missing from the request.

Version Added: 1.0.0

VersionDeprecated: 1.14.0

Deprecation Details: This message was deprecated in favor of `PropertyMissing`.

MessageSeverity: Critical

Resolution: Correct the body to include the required property with a valid value and resubmit the request if the operation failed.

Message and Arguments: “The create operation failed because the required property was missing from the request.”

1. *string*: The name of the required property.
 - This argument shall contain the name of the required property.

Sample Usage: None

3.1.29 CreateLimitReachedForResource

Indicates that no more resources can be created on the resource as it has reached its create limit.

- This message shall indicate that no more resources can be created on the resource as it has reached its create limit.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Critical

Resolution: Either delete resources and resubmit the request if the operation failed or do not resubmit the request.

Message and Arguments: “The create operation failed because the resource has reached the limit of possible resources.”

Sample Usage:

- [Error Case: Create Limit Reached For Resource](#)

3.1.30 Created

Indicates that all conditions of a successful create operation were met.

- This message shall indicate that all conditions of a successful create operation were met.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: OK

Resolution: None.

Message and Arguments: “The resource was created successfully.”

Sample Usage: None

3.1.31 EmptyJSON

Indicates that the request body contained an empty JSON object when one or more properties are expected in the body.

- This message shall indicate that the request body contained an empty JSON object when one or more properties are expected in the body.

Version Added: 1.2.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: Add properties in the JSON object and resubmit the request.

Message and Arguments: “The request body submitted contained an empty JSON object and the service is unable to process it.”

Sample Usage:

- [Error Case: Empty JSON](#)

3.1.32 EventBufferExceeded

Indicates undelivered events may have been lost due to a lack of buffer space in the service.

- This message shall indicate that event buffer space used by the service has been exceeded, and as a result one or more undelivered events may have been lost.

Version Added: 1.13.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: None.

Message and Arguments: “Undelivered events may have been lost due to exceeding the event buffer.”

Sample Usage: None

3.1.33 EventSubscriptionLimitExceeded

Indicates that an event subscription establishment was requested but the operation failed due to the number of simultaneous connections exceeding the limit of the implementation.

- This message shall indicate that an event subscription establishment was requested but the operation failed due to the number of simultaneous connections exceeding the limit of the implementation.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Critical

Resolution: Reduce the number of other subscriptions before trying to establish the event subscription or increase the limit of simultaneous subscriptions, if supported.

Message and Arguments: “The event subscription failed due to the number of simultaneous subscriptions exceeding the limit of the implementation.”

Sample Usage: None

3.1.34 GeneralError

Indicates that a general error has occurred. Use in `@Message.ExtendedInfo` is discouraged. When used in `@Message.ExtendedInfo`, implementations are expected to include a `Resolution` property with this message and provide a service-defined resolution to indicate how to resolve the error.

- This message shall indicate that a general error has occurred. Implementations should not use this message in `@Message.ExtendedInfo`. If used in `@Message.ExtendedInfo`, implementations should include a `Resolution` property with this message and provide a service-defined resolution to indicate how to resolve the error.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Critical

Resolution: None.

Message and Arguments: “A general error has occurred. See Resolution for information on how to resolve the error, or [`@Message.ExtendedInfo`] if Resolution is not provided.”

Sample Usage: None

3.1.35 GenerateSecretKeyRequired

Indicates that the Time-based One-Time Password (TOTP) secret key needs to be generated for the account before accessing the service. The secret key can be generated with a `POST` to the `GenerateSecretKey` action for the `ManagerAccount` resource instance.

- This message shall indicate that the RFC6238-defined Time-based One-Time Password (TOTP) secret key needs to be generated for the account before accessing the service. The secret key can be generated with a `POST` to the `GenerateSecretKey` action for the `ManagerAccount` resource instance.

Version Added: 1.19.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Critical

Resolution: Generate secret key for this account by performing the `GenerateSecretKey` action on the referenced URI and retaining the secret key from the action response to produce a Time-based One-Time Password (TOTP) for the `Token` property in future session creation requests.

Message and Arguments: “The Time-based One-Time Password (TOTP) secret key for this account must be generated before access is granted. Perform the `GenerateSecretKey` action at URI “ and retain the secret key from the response.”

1. *string*: The URI for the `GenerateSecretKey` action of the account that requires a new Time-based One-Time Password (TOTP) secret key.
 - This argument shall contain the URI for the `GenerateSecretKey` action of the account that requires a new RFC6238-defined Time-based One-Time Password (TOTP) secret key.

Sample Usage: None

3.1.36 HeaderInvalid

Indicates that a request header is invalid.

- This message shall indicate that a request header is known but is invalid.

Version Added: 1.13.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Critical

Resolution: Resubmit the request with a valid request header.

Message and Arguments: “Header “ is invalid.”

1. *string*: The request header.
 - This argument shall contain the full request header, including the field name and field value.

Sample Usage: None

3.1.37 HeaderMissing

Indicates that a required request header is missing.

- This message shall indicate that a required request header is missing in the request.

Version Added: 1.13.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Critical

Resolution: Resubmit the request with the required request header.

Message and Arguments: “Required header “ is missing in the request.”

1. *string*: The missing required request header.
 - This argument shall contain the missing required request header.

Sample Usage: None

3.1.38 InsufficientPrivilege

Indicates that the credentials associated with the established session do not have sufficient privileges for the requested operation.

- This message shall indicate that the credentials associated with the established session do not have sufficient privileges for the requested operation.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Critical

Resolution: Either abandon the operation or change the associated access rights and resubmit the request if the operation failed.

Message and Arguments: “There are insufficient privileges for the account or credentials associated with the current session to perform the requested operation.”

Sample Usage: None

3.1.39 InsufficientStorage

Indicates that the operation could not be completed due to a lack of storage or memory available to the service.

- This message shall indicate that the request exceeds the storage or memory available to the service. For cases where the payload is larger than the maximum size supported, regardless of available storage, the message `PayloadTooLarge` shall be used.

Version Added: 1.13.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Critical

Resolution: Increase the free storage space available to the service and resubmit the request.

Message and Arguments: “Insufficient storage or memory available to complete the request.”

Sample Usage: None

3.1.40 InternalError

Indicates that the request failed for an unknown internal error but that the service is still operational.

- This message shall indicate that the request failed for an unknown internal error but that the service is still operational.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Critical

Resolution: Resubmit the request. If the problem persists, consider resetting the service.

Message and Arguments: “The request failed due to an internal service error. The service is still operational.”

Sample Usage: None

3.1.41 InvalidIndex

The index is not valid.

- This message shall indicate that the index is not valid.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: Verify the index value provided is within the bounds of the array.

Message and Arguments: “The index is not a valid offset into the array.”

1. *number*: The array index.
 - This argument shall contain the array index value provided.

Sample Usage: None

3.1.42 InvalidJSON

Indicates that the request body contains invalid JSON.

- This message shall indicate that the request body contains invalid JSON.

Version Added: 1.10.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Critical

Resolution: Ensure that the request body is valid JSON and resubmit the request.

Message and Arguments: “The request body submitted is invalid JSON starting at line and could not be parsed by the receiving service.”

1. *number*: The line number where the invalid JSON starts.
 - This argument shall contain the line number where the invalid JSON starts.

Sample Usage:

- [Error Case: Invalid JSON](#)

3.1.43 InvalidObject

Indicates that the object in question is invalid according to the implementation. An example is a firmware update malformed URI.

- This message shall indicate that the object in question is invalid according to the implementation.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Critical

Resolution: Either the object is malformed or the URI is not correct. Correct the condition and resubmit the request if it failed.

Message and Arguments: “The object at “ is invalid.”

1. *string*: The URI provided.
 - This argument shall contain the URI provided for this operation.

Sample Usage: None

3.1.44 InvalidURI

Indicates that the operation encountered a URI that does not correspond to a valid resource.

- This message shall indicate that the operation encountered a URI that does not correspond to a valid resource.

Version Added: 1.10.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Critical

Resolution: Provide a valid URI and resubmit the request.

Message and Arguments: “The URI was not found.”

1. *string*: The supplied URI.
 - This argument shall contain the invalid URI that was encountered.

Sample Usage: None

3.1.45 LicenseRequired

Indicates that a license is required to perform the requested operation.

- This message shall indicate that that a license is required to perform the requested operation.

Version Added: 1.16.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Critical

Resolution: Install the requested license and resubmit the request.

Message and Arguments: “A license is required for this operation: .”

1. *string*: The vendor description of the license required to perform the operation.
 - This argument shall contain the vendor description of the license required to perform the operation.

Sample Usage: None

3.1.46 MalformedJSON

Indicates that the request body was malformed JSON.

- This message shall indicate that the request body was malformed JSON.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Critical

Resolution: Ensure that the request body is valid JSON and resubmit the request.

Message and Arguments: “The request body submitted was malformed JSON and could not be parsed by the receiving service.”

Sample Usage:

- [Error Case: Malformed JSON](#)

3.1.47 MaximumErrorsExceeded

Indicates that so many errors have occurred that the reporting service cannot return them all.

- The maximum number of errors that the implementation can report was exceeded. This shall not be the only message reported by the service. The use of this message should be limited to RDE implementations.

Version Added: 1.8.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Critical

Resolution: Resolve other reported errors and retry the current operation.

Message and Arguments: “Too many errors have occurred to report them all.”

Sample Usage: None

3.1.48 MissingOrMalformedPart

Indicates that a multipart request is missing a required part or contains malformed parts.

- This message shall indicate that the request payload supplied as part of an HTTP multipart request is missing one or more required parts or contains malformed parts.

Version Added: 1.13.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Critical

Resolution: Add any missing required parts or correct the malformed parts and resubmit the request.

Message and Arguments: “The multipart request contains malformed parts or is missing required parts.”

Sample Usage: None

3.1.49 NetworkNameResolutionNotConfigured

Indicates that network-based name resolution is not configured on the service.

- This message shall indicate that network-based name resolution, such as DNS or WINS protocols, is not configured on the service.

Version Added: 1.11.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: Configure the network-based name resolution protocol support on this service, or update any URI values to include an IP address instead of a network name and resubmit the request.

Message and Arguments: "Network name resolution is not configured on this service."

Sample Usage: None

3.1.50 NetworkNameResolutionNotSupported

Indicates the service does not support network-based name resolution.

- This message shall indicate that the service does not support network-based name resolution, such as DNS or WINS protocols. URIs provided as property values must contain an IP address as the service cannot resolve a network name.

Version Added: 1.11.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: Update any URI values to include an IP address instead of a network name and resubmit the request.

Message and Arguments: "Resolution of network-based names is not supported by this service."

Sample Usage: None

3.1.51 NoOperation

Indicates that the requested operation will not perform any changes on the service.

- This message shall indicate that the requested operation will not perform any changes on the service.

Version Added: 1.4.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: Add properties in the JSON object and resubmit the request.

Message and Arguments: “The request body submitted contain no data to act upon and no changes to the resource took place.”

Sample Usage:

- [Error Case: NoOperation](#)

3.1.52 NoValidSession

Indicates that the operation failed because a valid session is required in order to access any resources.

- This message shall indicate that the operation failed because a valid session is required in order to access any resources.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Critical

Resolution: Establish a session before attempting any operations.

Message and Arguments: “There is no valid session established with the implementation.”

Sample Usage: None

3.1.53 OneTimePasscodeSent

Indicates that a required one-time passcode was sent to the user. The code should be provided as the `Token` property in the request to create a session.

- This message shall indicate that a one-time passcode was generated by the service as the result of a request to create a session. The one-time passcode shall be sent to the delivery address associated with account credentials supplied in the request.

Version Added: 1.17.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: OK

Resolution: Obtain the one-time passcode sent to the delivery address and resubmit the request using the passcode as the `Token` property.

Message and Arguments: “A one-time passcode was sent to: . Supply the passcode as the `Token` property in the request to create a session.”

1. *string*: The delivery address for the one-time passcode. The address is at least partially redacted.
 - This argument shall contain the delivery address for the one-time passcode. The value shall be partially or fully redacted to prevent receipt by unauthorized clients.

Sample Usage: None

3.1.54 OperationFailed

Indicates that one of the internal operations necessary to complete the request failed. An example of this is when an internal service provider is unable to complete the request, such as in aggregation or RDE.

- This message shall indicate that one of the internal operations necessary to complete the request failed. Partial results of the client operation may be returned.

Version Added: 1.8.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: Resubmit the request. If the problem persists, consider resetting the service or provider.

Message and Arguments: “An error occurred internal to the service as part of the overall request. Partial results may have been returned.”

Sample Usage: None

3.1.55 OperationNotAllowed

Indicates that the HTTP method in the request is not allowed on this resource.

- This message shall indicate that the HTTP method in the request is not allowed on this resource. This error message should be used together with HTTP status code 405 Method Not Allowed.

Version Added: 1.13.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Critical

Resolution: None.

Message and Arguments: “The HTTP method is not allowed on this resource.”

Sample Usage: None

3.1.56 OperationTimeout

Indicates that one of the internal operations necessary to complete the request timed out. An example of this is when an internal service provider is unable to complete the request, such as in aggregation or RDE.

- This message shall indicate that one of the internal operations necessary to complete the request timed out. Partial results of the client operation may be returned.

Version Added: 1.8.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: Resubmit the request. If the problem persists, consider resetting the service or provider.

Message and Arguments: “A timeout internal to the service occurred as part of the request. Partial results may have been returned.”

Sample Usage: None

3.1.57 PasswordChangeRequired

Indicates that the password for the account provided must be changed before accessing the service. The password can be changed with a `PATCH` to the `Password` property in the manager account resource instance. Implementations that provide a default password for an account may require a password change prior to first access to the service.

- This message shall indicate that the password for the account provided must be changed before accessing the service. The password can be changed with a `PATCH` to the `Password` property in the manager account resource instance. Implementations that provide a default password for an account may require a password change prior to first access to the service.

Version Added: 1.5.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Critical

Resolution: Change the password for this account using a `PATCH` to the `Password` property at the URI provided.

Message and Arguments: “The password provided for this account must be changed before access is granted. `PATCH` the `Password` property for this account located at the target URI ” to complete this process.”

1. *string*: The URI of the target resource.
 - This argument shall contain the URI of the target manager account resource instance.

Sample Usage: None

3.1.58 PasswordComplexityNotMet

Indicates that the password for the account provided does not meet the password complexity requirements of the service. The password for the account was not changed.

- This message shall indicate that the password for the account provided does not meet the password complexity requirements of the service.

Version Added: 1.21.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Critical

Resolution: Resubmit the request with a password that meets the password complexity requirements as specified by the `PasswordGuidanceMessage` property in the `AccountService` resource.

Message and Arguments: “The password provided for this account does not meet the password complexity requirements of the service.”

Sample Usage: None

3.1.59 PasswordIncorrectLength

Indicates that the password for the account provided does not meet the password length requirements of the service. The password for the account was not changed.

- This message shall indicate that the password for the account provided does not meet the length requirements of the service. The length of the supplied password is either too short or too long.

Version Added: 1.21.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Critical

Resolution: Resubmit the request with a password that meets the password length requirements as specified by the `MinPasswordLength` and `MaxPasswordLength` properties in the `AccountService` resource.

Message and Arguments: “The password provided for this account does not meet the password length requirements of the service.”

Sample Usage: None

3.1.60 PasswordReuseTooRecent

Indicates that the password for the account provided does not meet the password reuse requirements of the service. The password for the account was not changed.

- This message shall indicate that the password for the account provided does not meet the password reuse requirements of the service. The supplied password was used too recently.

Version Added: 1.21.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Critical

Resolution: Resubmit the request with a password that meets the password reuse requirements specified by the `EnforcePasswordHistoryCount` property in the `AccountService` resource.

Message and Arguments: “The password provided for this account does not meet the password reuse requirements of the service.”

Sample Usage: None

3.1.61 PayloadTooLarge

Indicates that the supplied payload is too large to be accepted by the service.

- This message shall indicate that the payload supplied exceeds the size supported by the service. For cases where the payload size is supported but there is insufficient available storage space, the message `InsufficientStorage` shall be used.

Version Added: 1.13.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Critical

Resolution: Check that the supplied payload is correct and supported by this service.

Message and Arguments: “The supplied payload exceeds the maximum size supported by the service.”

Sample Usage: None

3.1.62 PreconditionFailed

Indicates that the ETag supplied did not match the current ETag of the resource.

- Indicates that the ETag supplied either in the header or using that `@odata.etag` annotation in the resource did not match the one expected by the service and so the operation on this resource failed. The use of this message shall only be used in responses for deep operations, since HTTP status code 412 is typically used for this.

Version Added: 1.8.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Critical

Resolution: Try the operation again using the appropriate ETag.

Message and Arguments: “The ETag supplied did not match the ETag required to change this resource.”

Sample Usage: None

3.1.63 PreconditionRequired

Indicates that the request did not provide the required precondition such as an `If-Match` or `If-None-Match` header or `@odata.etag` annotations.

- Indicates that the request did not provide the required precondition such as an `If-Match` or `If-None-Match` header or `@odata.etag` annotations. The use of this message shall only be used in responses for deep operations, since HTTP status code 428 is typically used for this.

Version Added: 1.8.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Critical

Resolution: Try the operation again using an `If-Match` or `If-None-Match` header and appropriate ETag.

Message and Arguments: “A precondition header or annotation is required to change this resource.”

Sample Usage: None

3.1.64 PropertyDeprecated

Indicates the property is deprecated.

- This message shall indicate that the property is deprecated.

Version Added: 1.10.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: Refer to the schema guide for more information.

Message and Arguments: “The deprecated property was included in the request body.”

1. *string*: The name of the property.
 - This argument shall contain the name of the property that is deprecated.

Sample Usage: None

3.1.65 PropertyDuplicate

Indicates that a duplicate property was included in the request body.

- This message shall indicate that a duplicate property was included in the request body.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: Remove the duplicate property from the request body and resubmit the request if the operation failed.

Message and Arguments: “The property was duplicated in the request.”

1. *string*: The name of the duplicate property.
 - This argument shall contain the name of the duplicate property. The value should follow RFC6901-defined JSON pointer notation rules.

Sample Usage:

- [Error Case: Property Duplicate](#)

3.1.66 PropertyMissing

Indicates that a required property was not supplied as part of the request.

- This message shall indicate that a required property was not supplied as part of the request.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: Ensure that the property is in the request body and has a valid value and resubmit the request if the operation failed.

Message and Arguments: “The property is a required property and must be included in the request.”

1. *string*: The name of the property.
 - This argument shall contain the name of the property. The value should follow RFC6901-defined JSON pointer notation rules.

Sample Usage:

- [Error Case: Property Missing](#)

3.1.67 PropertyModified

Indicates that all properties listed in `RelatedProperties` in the message were successfully modified.

- This message shall indicate that the properties listed in `RelatedProperties` were successfully modified.

Version Added: 1.19.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: OK

Resolution: None.

Message and Arguments: “One or more properties were successfully modified.”

Sample Usage: None

3.1.68 PropertyNotUpdated

Indicates that a property was not updated due to an internal service error, but the service is still functional.

- This message shall indicate that a property was not updated due to an internal service error, but the service is still functional.

Version Added: 1.12.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Critical

Resolution: Resubmit the request. If the problem persists, check for additional messages and consider resetting the service.

Message and Arguments: “The property was not updated due to an internal service error. The service is still operational.”

1. *string*: The name of the property.
 - This argument shall contain the name of the property. The value should follow RFC6901-defined JSON pointer notation rules.

Sample Usage:

- [Error Case: Property Not Updated](#)

3.1.69 PropertyNotWritable

Indicates that a property was given a value in the request body, but the property is a read-only property.

- This message shall indicate that a property was given a value in the request body, but the property is a read-only property.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: Remove the property from the request body and resubmit the request if the operation failed.

Message and Arguments: “The property is a read-only property and cannot be assigned a value.”

1. *string*: The name of the property.
 - This argument shall contain the name of the property. The value should follow RFC6901-defined JSON pointer notation rules.

Sample Usage:

- [Error Case: Property Not Writable](#)

3.1.70 PropertyUnknown

Indicates that an unknown property was included in the request body.

- This message shall indicate that an unknown property was included in the request body.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: Remove the unknown property from the request body and resubmit the request if the operation failed.

Message and Arguments: “The property is not in the list of valid properties for the resource.”

1. *string*: The name of the unknown property.
 - This argument shall contain the name of the unknown property. The value should follow RFC6901-defined JSON pointer notation rules.

Sample Usage:

- [Error Case: Property Unknown](#)

3.1.71 PropertyValueConflict

Indicates that the requested write of a property value could not be completed because of a conflict with another property value.

- This message shall indicate that the requested write of a property could not be completed because of a conflict with the value of another property.

Version Added: 1.6.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: None.

Message and Arguments: “The property “ could not be written because its value would conflict with the value of the “ property.”

1. *string*: The name of the property for which a write was requested.
 - This argument shall contain the name of the property for which a write was requested.
2. *string*: The name of the property with which there is a conflict.
 - This argument shall contain the name of the property with which there is a conflict.

Sample Usage:

- [Error Case: Property Value Conflict](#)

3.1.72 PropertyValueDeprecated

Indicates that a property was given a deprecated value.

- This message shall indicate that a property was given a deprecated value.

Version Added: 1.10.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: Refer to the schema guide for more information.

Message and Arguments: “The value ” for the property is deprecated.”

1. *string*: The value provided for the property.
 - This argument shall contain the deprecated value provided for the property.
2. *string*: The name of the property.
 - This argument shall contain the name of the property.

Sample Usage: None

3.1.73 PropertyValueError

Indicates that a property was given an invalid value.

- This message shall indicate that a property was given an invalid value, and the value cannot or should not be included in the message text.

Version Added: 1.11.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: Correct the value for the property in the request body and resubmit the request if the operation failed.

Message and Arguments: “The value provided for the property is not valid.”

1. *string*: The name of the property.
 - This argument shall contain the name of the property. The value should follow RFC6901-defined JSON pointer notation rules.

Sample Usage: None

3.1.74 PropertyValueExternalConflict

Indicates that the requested write of a property value could not be completed due to the current state or configuration of the resource. This can include configuration conflicts with other resources or parameters that are not exposed by this interface.

- This message shall indicate that the requested write of a property could not be completed because the requested value of the property, which is a value supported by the implementation, cannot be accepted due to conflicts in the state or configuration of the resource.

Version Added: 1.10.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: None.

Message and Arguments: “The property “ with the requested value of “ could not be written because the value is not available due to a configuration conflict.”

1. *string*: The name of the property for which a write was requested.
 - This argument shall contain the name of the property for which a write was requested.
2. *string*: The value of the property that is in conflict.
 - This argument shall contain the value of the property that is in conflict.

Sample Usage: None

3.1.75 PropertyValueFormatError

Indicates that a property was given the correct value type but the format of the value is not supported.

- This message shall indicate that a property was given the correct value type but the format of the value is not supported.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: Correct the value for the property in the request body and resubmit the request if the operation failed.

Message and Arguments: “The value ” for the property is not a format that the property can accept.”

1. *string*: The value provided for the property.
 - This argument shall contain the value provided for the property. Numeric values shall be converted to a string, and null values shall be represented by the string `null`.
2. *string*: The name of the property.
 - This argument shall contain the name of the property. The value should follow RFC6901-defined JSON pointer notation rules.

Sample Usage:

- [Error Case: Property Value Format Error](#)

3.1.76 PropertyValueIncorrect

Indicates that the requested write of a property value could not be completed because of an incorrect value of the property. Examples include values that do not match a regular expression requirement, passwords that do not match the implementation constraints, or specifying an invalid URI in a reference property.

- This message shall indicate that the requested write of a property could not be completed because of an incorrect value for the property.

Version Added: 1.7.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: None.

Message and Arguments: “The property ” with the requested value of ” could not be written because the

value is not acceptable for the property.”

1. *string*: Name of the property for which a write was requested.
 - This argument shall contain the name of the property for which a write was requested.
2. *string*: Value of the property that is incorrect.
 - This argument shall contain the value of the property that is incorrect.

Sample Usage:

- [Error Case: Property Value Incorrect](#)

3.1.77 PropertyValueModified

Indicates that a property was given the correct value type but the value of that property was modified.
Examples are truncated or rounded values.

- This message shall indicate that a property was given the correct value type but the value of that property was modified.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: None.

Message and Arguments: “The property was assigned the value ” due to modification by the service.”

1. *string*: The name of the property.
 - This argument shall contain the name of the property.
2. *string*: The value assigned for property.
 - This argument shall contain the value assigned for property.

Sample Usage: None

3.1.78 PropertyValueNotInList

Indicates that a property was given the correct value type but the value of that property was not supported.
The value is not in an enumeration.

- This message shall indicate that a property was given the correct value type but the value of that property was not supported.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: Choose a value from the enumeration list that the implementation can support and resubmit the request if the operation failed.

Message and Arguments: “The value ” for the property is not in the list of acceptable values.”

1. *string*: The value provided for the property.
 - This argument shall contain the value provided for the property.
2. *string*: The name of the property.
 - This argument shall contain the name of the property. The value should follow RFC6901-defined JSON pointer notation rules.

Sample Usage:

- [Error Case: Property Value Not In List](#)

3.1.79 PropertyValueOutOfRange

Indicates that a property was given the correct value type but the value of that property is outside the supported range.

- This value shall indicate that a property was given the correct value type but the value of that property is outside the supported range.

Version Added: 1.3.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: Correct the value for the property in the request body and resubmit the request if the operation failed.

Message and Arguments: “The value ” for the property is not in the supported range of acceptable values.”

1. *string*: The value provided for the property.
 - This argument shall contain the value provided for the property.
2. *string*: The name of the property.

- This argument shall contain the name of the property. The value should follow RFC6901-defined JSON pointer notation rules.

Sample Usage:

- [Error Case: Property Value Out of Range](#)

3.1.80 PropertyValueResourceConflict

Indicates that the requested write of a property value could not be completed due to the current state or configuration of another resource.

- This message shall indicate that the requested write of a property could not be completed because the requested value of the property, which is a value supported by the implementation, cannot be accepted due to conflicts in the state or configuration of another resource.

Version Added: 1.10.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: None.

Message and Arguments: “The property “ with the requested value of “ could not be written because the value conflicts with the state or configuration of the resource at “.”

1. *string*: The name of the property for which a write was requested.
 - This argument shall contain the name of the property for which a write was requested.
2. *string*: The value of the property that is in conflict.
 - This argument shall contain the value of the property that is in conflict.
3. *string*: The URI of the resource in conflict.
 - This argument shall contain the URI of the resource that is in conflict with the value of the requested property.

Sample Usage:

- [Error Case: Property Value Resource Conflict](#)

3.1.81 PropertyValueTypeError

Indicates that a property was given the wrong value type, such as when a number is supplied for a property that requires a string.

- This message shall indicate that a property was given the wrong value type.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: Correct the value for the property in the request body and resubmit the request if the operation failed.

Message and Arguments: “The value ” for the property is not a type that the property can accept.”

1. *string*: The value provided for the property.
 - This argument shall contain the value provided for the property. Numeric values shall be converted to a string, and null values shall be represented by the string `null`.
2. *string*: The name of the property.
 - This argument shall contain the name of the property. The value should follow RFC6901-defined JSON pointer notation rules.

Sample Usage:

- [Error Case: Property Value Type Error](#)

3.1.82 QueryCombinationInvalid

Indicates the request contains multiple query parameters and that two or more of them cannot be used together.

- This message shall indicate that the request contains multiple query parameters and that two or more of them cannot be used together.

Version Added: 1.6.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: Remove one or more of the query parameters and resubmit the request if the operation failed.

Message and Arguments: “Two or more query parameters in the request cannot be used together.”

Sample Usage: None

3.1.83 QueryNotSupported

Indicates that query is not supported on the implementation.

- This message shall indicate that query is not supported on the implementation.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: Remove the query parameters and resubmit the request if the operation failed.

Message and Arguments: “Querying is not supported by the implementation.”

Sample Usage: None

3.1.84 QueryNotSupportedOnOperation

Indicates that query is not supported with the given operation, such as when the `$expand` query is attempted with a `PATCH` operation.

- This message shall indicate that query is not supported with the given operation.

Version Added: 1.6.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: Remove the query parameters and resubmit the request if the operation failed.

Message and Arguments: “Querying is not supported with the requested operation.”

Sample Usage: None

3.1.85 QueryNotSupportedOnResource

Indicates that query is not supported on the given resource, such as when the `$skip` query is attempted on a resource that is not a collection.

- This message shall indicate that query is not supported on the given resource.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: Remove the query parameters and resubmit the request if the operation failed.

Message and Arguments: “Querying is not supported on the requested resource.”

Sample Usage:

- [Error Case: Query Not Supported On Resource](#)

3.1.86 QueryParameterOutOfRange

Indicates that a query parameter was provided that is out of range for the given resource. This can happen with values that are too low or that exceed what is possible for the supplied resource, such as when a page is requested that is beyond the last page.

- This message shall indicate that a query parameter was provided that is out of range for the given resource.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: Reduce the value for the query parameter to a value that is within range, such as a start or count value that is within bounds of the number of resources in a collection or a page number that is within the range of valid pages.

Message and Arguments: “The value ” for the query parameter is out of range .”

1. *string*: The value provided for the query parameter.
 - This argument shall contain the value provided for the query parameter.
2. *string*: The query parameter.
 - This argument shall contain the name of the query parameter.
3. *string*: The valid range for the query parameter.
 - This argument shall contain a string representation of the valid range of values

supported for the query parameter.

Sample Usage: None

3.1.87 QueryParameterUnsupported

Indicates that a query parameter is not supported.

- This message shall indicate that a query parameter is not supported by this service.

Version Added: 1.15.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: Correct or remove the query parameter and resubmit the request.

Message and Arguments: “Query parameter “ is not supported.”

1. *string*: The query parameter.
 - This argument shall contain the name of the query parameter.

Sample Usage: None

3.1.88 QueryParameterValueError

Indicates that a query parameter was given an invalid value.

- This message shall indicate that a query parameter was given an invalid value that cannot or should not be included in the message.

Version Added: 1.11.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: Correct the value for the query parameter in the request and resubmit the request if the operation failed.

Message and Arguments: “The value for the parameter is invalid.”

1. *string*: The query parameter.
 - This argument shall contain the name of the query parameter.

Sample Usage: None

3.1.89 QueryParameterValueFormatError

Indicates that a query parameter was given the correct value type but the format of the value is not supported.

- This message shall indicate that a query parameter was given the correct value type but the format of the value is not supported.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: Correct the value for the query parameter in the request and resubmit the request if the operation failed.

Message and Arguments: “The value ” for the parameter is not a format that the parameter can accept.”

1. *string*: The value provided for the query parameter.
 - This argument shall contain the value provided for the query parameter.
2. *string*: The query parameter.
 - This argument shall contain the name of the query parameter.

Sample Usage: None

3.1.90 QueryParameterValueTypeError

Indicates that a query parameter was given the wrong value type, such as when a number is supplied for a query parameter that requires a string.

- This message shall indicate that a query parameter was given the wrong value type, such as when a number is supplied for a query parameter that requires a string.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: Correct the value for the query parameter in the request and resubmit the request if the operation failed.

Message and Arguments: “The value ” for the query parameter is not a type that the parameter can accept.”

1. *string*: The value provided for the query parameter.
 - This argument shall contain the value provided for the query parameter.
2. *string*: The query parameter.
 - This argument shall contain the name of the query parameter.

Sample Usage: None

3.1.91 ResetRecommended

Indicates that a component reset is recommended for error recovery while unaffected applications can continue running without any effect on accuracy and performance.

- This message shall indicate that a component reset is recommended for error recovery while unaffected applications can continue running without any effect on accuracy and performance.

Version Added: 1.12.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: Perform the recommended reset action on the specified component.

Message and Arguments: “In order to recover from errors, a component reset is recommended with the Reset action URI ” and ResetType ”.”

1. *string*: The URI for the reset action of the component that recommends a reset.
 - This argument shall contain the URI for the reset action that is recommended to be invoked for error recovery.
2. *string*: The `ResetType` required in a reset action on the component URI.
 - This argument shall contain the `ResetType` required in a reset action on the component URI for error recovery while unaffected applications can continue running without any effect on accuracy and performance. A reset action `POST` shall contain this `ResetType` parameter.

Sample Usage: None

3.1.92 ResetRequired

Indicates that a component reset is required for changes, error recovery, or operations to complete.

- This message shall indicate that a component reset is required for changes, error recovery, or operations to complete.

Version Added: 1.6.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: Perform the required reset action on the specified component.

Message and Arguments: “In order to apply changes, recover from errors, or complete the operation, a component reset is required with the Reset action URI ” and ResetType “.”

1. *string*: The URI for the reset action of the component that requires a reset.
 - This argument shall contain the URI for the reset action that is required to be invoked for changes, error recovery or operations to complete.
2. *string*: The `ResetType` required in a reset action on the component URI.
 - This argument shall contain the `ResetType` required in a reset action on the component URI for changes, error recovery or operations to complete. A reset action `POST` shall contain this `ResetType` parameter.

Sample Usage: None

3.1.93 ResourceAlreadyExists

Indicates that a resource change or creation was attempted but that the operation cannot proceed because the resource already exists.

- This message shall indicate that a resource change or creation was attempted but that the operation cannot proceed because the resource already exists.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Critical

Resolution: Do not repeat the create operation as the resource was already created.

Message and Arguments: “The requested resource of type with the property with the value ” already exists.”

1. *string*: The type of resource.
 - This argument shall contain the schema name of the resource.
2. *string*: The name of the property.
 - This argument shall contain the name of the property.
3. *string*: The value of the property.
 - This argument shall contain the value provided for the property. Numeric values shall be converted to a string, and null values shall be represented by the string `null`.

Sample Usage:

- [Error Case: Resource Already Exists](#)

3.1.94 ResourceAtUriInUnknownFormat

Indicates that the URI was valid but the resource or image at that URI was in a format not supported by the service.

- This message shall indicate that the URI was valid but the resource or image at that URI was in a format not supported by the service.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Critical

Resolution: Place an image or resource or file that is recognized by the service at the URI.

Message and Arguments: “The resource at ” is in a format not recognized by the service.”

1. *string*: The URI provided.
 - This argument shall contain the URI provided for this operation.

Sample Usage: None

3.1.95 ResourceAtUriUnauthorized

Indicates that the attempt to access the resource, file, or image at the URI was unauthorized.

- This message shall indicate that the attempt to access the resource, file, or image at the URI was unauthorized.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Critical

Resolution: Ensure that the appropriate access is provided for the service in order for it to access the URI.

Message and Arguments: “While accessing the resource at “, the service received an authorization error “.”

1. *string*: The URI provided.
 - This argument shall contain the URI provided for this operation.
2. *string*: The error message received from the authorization service.
 - This argument shall contain the error message received from the authorization service or software.

Sample Usage: None

3.1.96 ResourceCannotBeDeleted

Indicates that a delete operation was attempted on a resource that cannot be deleted.

- This message shall indicate that a delete operation was attempted on a resource that cannot be deleted.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Critical

Resolution: Do not attempt to delete a non-deletable resource.

Message and Arguments: “The delete request failed because the resource requested cannot be deleted.”

Sample Usage:

- [Error Case: Resource Cannot Be Deleted](#)

3.1.97 ResourceCreationConflict

Indicates that the requested resource creation could not be completed because the service has a resource that conflicts with the request.

- This message shall indicate that the requested creation of a resource could not be completed because the service has a resource that conflicts with the request, such as when the resource already exists or an existing resource has similar property values.

Version Added: 1.7.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: None.

Message and Arguments: “The resource could not be created. The service has a resource at URI ” that conflicts with the creation request.”

1. *string*: URI of the resource that conflicts with the creation request.
 - This argument shall contain the URI of the resource with which there is a conflict.

Sample Usage:

- [Error Case: Resource Creation Conflict](#)

3.1.98 ResourceDeprecated

Indicates the resource is deprecated.

- This message shall indicate that the resource is deprecated.

Version Added: 1.10.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: Refer to the schema guide for more information.

Message and Arguments: “The operation was performed on a deprecated resource “.”

1. *string*: The URI of the resource.
 - This argument shall contain the URI of the resource that is deprecated.

Sample Usage: None

3.1.99 ResourceExhaustion

Indicates that a resource could not satisfy the request due to some unavailability of resources. An example is that available capacity was allocated.

- This message shall indicate that a resource could not satisfy the request due to some unavailability of resources.

Version Added: 1.2.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Critical

Resolution: Ensure that the resources are available and resubmit the request.

Message and Arguments: “The resource “ was unable to satisfy the request due to unavailability of resources.”

1. *string*: The URI of the resource.
 - This argument shall contain the URI of the relevant resource.

Sample Usage:

- [Error Case: Resource Exhaustion](#)

3.1.100 ResourceInStandby

Indicates that the request could not be performed because the resource is in standby.

- This message shall indicate that the request could not be performed because the resource is in standby.

Version Added: 1.2.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Critical

Resolution: Ensure that the resource is in the correct power state and resubmit the request.

Message and Arguments: “The request could not be performed because the resource is in standby.”

Sample Usage:

- [Error Case: Resource In Standby](#)

3.1.101 ResourceInUse

Indicates that a change was requested to a resource but the change was rejected due to the resource being in use or transition.

- This message shall indicate that a change was requested to a resource but the change was rejected due to the resource being in use or transition.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: Remove the condition and resubmit the request if the operation failed.

Message and Arguments: “The change to the requested resource failed because the resource is in use or in transition.”

Sample Usage:

- [Error Case: Resource In Use](#)

3.1.102 ResourceMissingAtURI

Indicates that the operation expected an image or other resource at the provided URI but none was found. Examples of this are in requests that require URIs such as firmware updates.

- This message shall indicate that the operation expected an image or other resource at the provided URI but none was found.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Critical

Resolution: Place a valid resource at the URI or correct the URI and resubmit the request.

Message and Arguments: “The resource at the URI ” was not found.”

1. *string*: The URI provided.
 - This argument shall contain the URI provided for this operation.

Sample Usage:

- [Error Case: Resource Missing At URI](#)

3.1.103 ResourceNotFound

Indicates that the operation expected a resource identifier that corresponds to an existing resource but one was not found.

- This message shall indicate that the operation expected a resource identifier that corresponds to an existing resource but one was not found.

Version Added: 1.2.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Critical

Resolution: Provide a valid resource identifier and resubmit the request.

Message and Arguments: “The requested resource of type named ” was not found.”

1. *string*: The type of resource.
 - This argument shall contain the schema name of the resource.
2. *string*: The `Id` of the resource.
 - This argument shall contain the value of the `Id` property of the requested resource.

Sample Usage:

- [Error Case: Resource Not Found](#)

3.1.104 ResourceTypeIncompatible

Indicates that the resource type of the operation does not match that of the operation destination.

Examples of this are a `POST` to a resource collection using the wrong resource type, an update where the

@odata.type properties do not match, or a case of major version incompatibility.

- This message shall indicate that the resource type of the operation does not match that of the operation destination.

Version Added: 1.2.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Critical

Resolution: Resubmit the request with a payload compatible with the resource's schema.

Message and Arguments: "The [@odata.type] of the request body is incompatible with the [@odata.type] of the resource, which is ."

1. *string*: The @odata.type of the request.
 - This argument shall contain the value of the @odata.type property provided in the request.
2. *string*: The supported @odata.type value.
 - This argument shall contain the value of the @odata.type property supported by the resource.

Sample Usage:

- [Error Case: Resource Type Incompatible](#)

3.1.105 RestrictedPrivilege

Indicates that the operation was not successful because a privilege is restricted.

- This message shall indicate that the operation was not successful because a privilege is restricted.

Version Added: 1.9.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: Remove restricted privileges from the request body and resubmit the request.

Message and Arguments: "The operation was not successful because the privilege " is restricted."

1. *string*: The name of the restricted privilege.
 - This argument shall contain the name of the restricted privilege.

Sample Usage: None

3.1.106 RestrictedRole

Indicates that the operation was not successful because the role is restricted.

- This message shall indicate that the operation was not successful because the role is restricted.

Version Added: 1.9.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: No resolution is required. For standard roles, consider using the role specified in the AlternateRoleId property in the Role resource.

Message and Arguments: “The operation was not successful because the role “ is restricted.”

1. *string*: The name of the restricted role.
 - This argument shall contain the name of the restricted role.

Sample Usage: None

3.1.107 ServiceDisabled

Indicates that the operation failed because the service, such as the account service, is disabled and cannot accept requests.

- This message shall indicate that the operation failed because the service, such as the account service, is disabled and cannot accept requests.

Version Added: 1.11.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: Enable the service and resubmit the request if the operation failed.

Message and Arguments: “The operation failed because the service at is disabled and cannot accept requests.”

1. *string*: The URI of the disabled service.
 - This argument shall contain the URI of the disabled service.

Sample Usage: None

3.1.108 ServiceInUnknownState

Indicates that the operation failed because the service is in an unknown state and cannot accept additional requests.

- This message shall indicate that the operation failed because the service is in an unknown state and cannot accept additional requests.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Critical

Resolution: Restart the service and resubmit the request if the operation failed.

Message and Arguments: “The operation failed because the service is in an unknown state and can no longer take incoming requests.”

Sample Usage: None

3.1.109 ServiceShuttingDown

Indicates that the operation failed because the service is shutting down, such as when the service reboots.

- This message shall indicate that the operation failed because the service is shutting down.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Critical

Resolution: When the service becomes available, resubmit the request if the operation failed.

Message and Arguments: “The operation failed because the service is shutting down and can no longer take incoming requests.”

Sample Usage: None

3.1.110 ServiceTemporarilyUnavailable

Indicates the service is temporarily unavailable.

- This message shall indicate that the service is temporarily unavailable.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Critical

Resolution: Wait for the indicated retry duration and retry the operation.

Message and Arguments: “The service is temporarily unavailable. Retry in seconds.”

1. *string*: The retry duration in seconds.
 - This argument shall contain the number of seconds the client should wait before attempting the operation again.

Sample Usage: None

3.1.111 SessionLimitExceeded

Indicates that a session establishment was requested but the operation failed due to the number of simultaneous sessions exceeding the limit of the implementation.

- This message shall indicate that a session establishment was requested but the operation failed due to the number of simultaneous sessions exceeding the limit of the implementation.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Critical

Resolution: Reduce the number of other sessions before trying to establish the session or increase the limit of simultaneous sessions, if supported.

Message and Arguments: “The session establishment failed due to the number of simultaneous sessions exceeding the limit of the implementation.”

Sample Usage: None

3.1.112 SessionTerminated

Indicates that the `DELETE` operation on the session resource resulted in the successful termination of the session.

- This message shall indicate that the `DELETE` operation on the session resource resulted in the successful termination of the session.

Version Added: 1.3.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: OK

Resolution: None.

Message and Arguments: “The session was successfully terminated.”

Sample Usage: None

3.1.113 SourceDoesNotSupportProtocol

Indicates that while attempting to access, connect to, or transfer a resource, file, or image from another location that the other end of the connection did not support the protocol.

- This message shall indicate that while attempting to access, connect to, or transfer a resource, file, or image from another location that the other end of the connection did not support the protocol.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Critical

Resolution: Change protocols or URIs.

Message and Arguments: “The other end of the connection at ” does not support the specified protocol .”

1. *string*: The URI provided.
 - This argument shall contain the URI provided for this operation.
2. *string*: The protocol specified.
 - This argument shall contain the name of the protocol requested for this operation.

Sample Usage: None

3.1.114 StrictAccountTypes

Indicates the request failed because a set of `AccountTypes` or `OEMAccountTypes` was not accepted while `StrictAccountTypes` is set to `true`.

- This message shall indicate that the request failed because a set of `AccountTypes` or `OEMAccountTypes` was not accepted while `StrictAccountTypes` is set to `true`.

Version Added: 1.9.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: Resubmit the request either with an acceptable set of `AccountTypes` and `OEMAccountTypes` or with `StrictAccountTypes` set to `false`.

Message and Arguments: “The request could not be fulfilled with the account types included in property “ because the property `StrictAccountTypes` is set to `true`.”

1. *string*: The name of the property.
 - This argument shall contain the name of the property `AccountTypes` or `OEMAccountTypes`.

Sample Usage: None

3.1.115 StringValueTooLong

Indicates that a string value passed to the given resource was longer than the maximum allowed length. An example is when an implementation has imposed a shorter maximum length than that allowed by the specification.

- This message shall indicate that a string value passed to the given resource was longer than the maximum allowed length.

Version Added: 1.2.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: Resubmit the request with an appropriate string length.

Message and Arguments: “The string “ exceeds the length limit .”

1. *string*: The string provided.
 - This argument shall contain the string value provided for this operation.
2. *number*: The maximum string length.
 - This argument shall contain the maximum supported string length for this property.

Sample Usage: None

3.1.116 StringValueTooShort

Indicates that a string value passed to the given resource was shorter than the minimum required length. An example is when an implementation has imposed a greater minimum length than that required by the specification.

- This message shall indicate that a string value passed to the given resource was shorter than the minimum required length.

Version Added: 1.16.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: Resubmit the request with an appropriate string length.

Message and Arguments: “The string “ was under the minimum required length .”

1. *string*: The string provided.
 - This argument shall contain the string value provided for this operation.
2. *number*: The minimum string length.
 - This argument shall contain the minimum supported string length for this property.

Sample Usage: None

3.1.117 SubscriptionTerminated

An event subscription was terminated by the service. No further events will be delivered.

- This message shall indicate that an event subscription was terminated by the service.

Version Added: 1.5.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: OK

Resolution: None.

Message and Arguments: "The event subscription was terminated."

Sample Usage: None

3.1.118 Success

Indicates that all conditions of a successful operation were met.

- This message shall indicate that all conditions of a successful operation were met.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: OK

Resolution: None.

Message and Arguments: "The request completed successfully."

Sample Usage: None

3.1.119 UndeterminedFault

Indicates that a fault or error condition exists but the source of the fault cannot be determined or is unknown to the service.

- This message shall indicate that a fault or error condition was reported to the service, but the

source of the fault cannot be determined or is unknown to the service.

Version Added: 1.9.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Critical

Resolution: None.

Message and Arguments: “An undetermined fault condition was reported by “.”

1. *string*: The software, provider, or service that reported the fault condition.
 - This argument shall contain the name of the entity that reported the fault condition.

Sample Usage: None

3.1.120 UnrecognizedRequestBody

Indicates that the service encountered an unrecognizable request body that could not even be interpreted as malformed JSON.

- This message shall indicate that the service encountered an unrecognizable request body that could not even be interpreted as malformed JSON.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: Correct the request body and resubmit the request if it failed.

Message and Arguments: “The service detected a malformed request body that it was unable to interpret.”

Sample Usage:

- [Error Case: Unrecognized Request Body](#)

3.2 ResourceEvent 1.4.2

3.2.1 Overview

This registry defines the messages to use for resource events.

The messages defined in this version of the ResourceEvent registry are summarized in Table 5.

Table 5: ResourceEvent messages

Message	Severity	Description
AggregationSourceDiscovered	OK	Indicates that a new aggregation source was discovered.
LicenseAdded (Deprecated)	OK	Indicates that a license was added.
LicenseChanged (Deprecated)	Warning	Indicates that a license has changed.
LicenseExpired (Deprecated)	Warning	Indicates that a license has expired.
PropertyValueModifiedByClient	OK	Indicates that the value of a property was modified by a client.
ResourceChanged	OK	Indicates that one or more resource properties have changed. This is not used whenever there is another event message for that specific change, such as when only the state has changed.
ResourceCreated	OK	Indicates that all conditions of a successful creation operation were met.
ResourceErrorThresholdCleared	OK	Indicates that a specified resource property has cleared its error threshold. Examples of this are drive I/O errors or network link errors.

Message	Severity	Description
ResourceErrorThresholdExceeded	Critical	Indicates that a specified resource property has exceeded its error threshold. Examples of this are drive I/O errors or network link errors.
ResourceErrorsCorrected	OK	Indicates that a specified resource property has corrected errors. Examples of this are drive I/O errors or network link errors.
ResourceErrorsDetected	Warning	Indicates that a specified resource property has detected errors. Examples of this are drive I/O errors or network link errors.
ResourcePaused	OK	Indicates that the power state of a resource has changed to paused.
ResourcePoweredOff	OK	Indicates that the power state of a resource has changed to powered off.
ResourcePoweredOn	OK	Indicates that the power state of a resource has changed to powered on.
ResourcePoweringOff	OK	Indicates that the power state of a resource has changed to powering off.
ResourcePoweringOn	OK	Indicates that the power state of a resource has changed to powering on.
ResourceRemoved	OK	Indicates that all conditions of a successful remove operation were met.
ResourceSelfTestCompleted	OK	Indicates that a self-test has completed.
ResourceSelfTestFailed	Critical	Indicates that a self-test has failed. Suggested resolution may be

Message	Severity	Description
		provided as OEM data.
ResourceStateChanged	OK	Indicates that the state of a resource has changed.
ResourceStatusChangedCritical	Critical	Indicates that the health of a resource has changed to Critical.
ResourceStatusChangedOK	OK	Indicates that the health of a resource has changed to OK.
ResourceStatusChangedWarning	Warning	Indicates that the health of a resource has changed to Warning.
ResourceVersionIncompatible	Warning	Indicates that an incompatible version of software was detected. Examples of this are after a component or system-level software update.
ResourceWarningThresholdCleared	OK	Indicates that a specified resource property has cleared its warning threshold. Examples of this are drive I/O errors or network link errors. Suggested resolution may be provided as OEM data.
ResourceWarningThresholdExceeded	Warning	Indicates that a specified resource property has exceeded its warning threshold. Examples of this are drive I/O errors or network link errors. Suggested resolution may be provided as OEM data.
TestMessage	OK	A test message used to validate event delivery mechanisms.
URIForResourceChanged	OK	Indicates that the URI for a resource has changed. Examples of this are

Message	Severity	Description
		physical component replacement or redistribution.

3.2.2 AggregationSourceDiscovered

Indicates that a new aggregation source was discovered.

- This message shall indicate an aggregation source was discovered in the network or host. The `OriginOfCondition` property shall contain a link to the matching `ConnectionMethod` resource, if known.

Version Added: 1.2.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: OK

Resolution: The aggregation source is available to the service and can be identified using the identified connection method.

Message and Arguments: “An aggregation source of connection method “ located at “ was discovered.”

1. *string*: The connection method of the discovered aggregation source.
 - This argument shall contain a value from the `ConnectionMethodType` enumeration defined in the `ConnectionMethod` schema that describes connection method of the discovered aggregation source.
2. *string*: The location of the discovered aggregation source, such as an IP address or service entry point.
 - This argument shall contain the location of the discovered aggregation source, such as an IP address or service entry point.

Sample Usage: None

3.2.3 LicenseAdded (Deprecated)

Indicates that a license was added.

- This message shall indicate that a license was added.

Version Added: 1.0.0

VersionDeprecated: 1.1.0

Deprecation Details: This message was deprecated in favor of `LicenseAdded` in the License Message Registry.

MessageSeverity: OK

Resolution: See vendor specific instructions for specific actions.

Message and Arguments: “A license for ” was added. The following message was returned: “.”

1. *string*: The relevant resource.
 - This argument shall contain the name of the relevant resource or service affected by the software license.
2. *string*: The message returned from the license validation process.
 - This argument shall contain the message returned from the license validation process or software.

Sample Usage: None

3.2.4 LicenseChanged (Deprecated)

Indicates that a license has changed.

- This message shall indicate that a license has changed.

Version Added: 1.0.0

VersionDeprecated: 1.1.0

Deprecation Details: This message was deprecated in favor of `ResourceChanged`.

MessageSeverity: Warning

Resolution: See vendor specific instructions for specific actions.

Message and Arguments: “A license for ” has changed. The following message was returned: “.”

1. *string*: The relevant resource.
 - This argument shall contain the name of the relevant resource or service affected by the software license.
2. *string*: The message returned from the license validation process.
 - This argument shall contain the message returned from the license validation process or software.

Sample Usage: None

3.2.5 LicenseExpired (Deprecated)

Indicates that a license has expired.

- This message shall indicate that a license has expired.

Version Added: 1.0.0

VersionDeprecated: 1.1.0

Deprecation Details: This message was deprecated in favor of `LicenseExpired` in the License Message Registry.

MessageSeverity: Warning

Resolution: See vendor specific instructions for specific actions.

Message and Arguments: “A license for ” has expired. The following message was returned: “.”

1. *string*: The relevant resource.
 - This argument shall contain the name of the relevant resource or service affected by the software license.
2. *string*: The message returned from the license validation process.
 - This argument shall contain the message returned from the license validation process or software.

Sample Usage: None

3.2.6 PropertyValueModifiedByClient

Indicates that the value of a property was modified by a client.

- This message shall indicate that the value of a property was modified by a client. For example, DHCPEnabled property was assigned the value `true` by a client.

Version Added: 1.4.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: OK

Resolution: No resolution is required.

Message and Arguments: “The property was assigned the value ” due to modification by a client.”

1. *string*: The name of the property.
 - This argument shall contain the name of the property. The value should follow RFC6901-defined JSON pointer notation rules.
2. *string*: The value assigned for property.
 - This argument shall contain the value assigned for property.

Sample Usage: None

3.2.7 ResourceChanged

Indicates that one or more resource properties have changed. This is not used whenever there is another event message for that specific change, such as when only the state has changed.

- This message shall indicate that one or more resource properties have changed.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: OK

Resolution: None.

Message and Arguments: “One or more resource properties have changed.”

Sample Usage: None

3.2.8 ResourceCreated

Indicates that all conditions of a successful creation operation were met.

- This message shall indicate that all conditions of a successful creation operation were met.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: OK

Resolution: None.

Message and Arguments: “The resource was created successfully.”

Sample Usage: None

3.2.9 ResourceErrorThresholdCleared

Indicates that a specified resource property has cleared its error threshold. Examples of this are drive I/O errors or network link errors.

- This message shall indicate that a specified resource property has cleared its error threshold.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: OK

Resolution: None.

Message and Arguments: “The resource property has cleared the error threshold of value .”

1. *string*: The name of the property.
 - This argument shall contain the name of the relevant property from a Redfish resource.
2. *number*: The value of the threshold.
 - This argument shall contain the value of the relevant error threshold.

Sample Usage: None

3.2.10 ResourceErrorThresholdExceeded

Indicates that a specified resource property has exceeded its error threshold. Examples of this are drive I/O errors or network link errors.

- This message shall indicate that a specified resource property has exceeded its error threshold.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Critical

Resolution: None.

Message and Arguments: “The resource property has exceeded error threshold of value .”

1. *string*: The name of the property.
 - This argument shall contain the name of the relevant property from a Redfish resource.
2. *number*: The value of the threshold.
 - This argument shall contain the value of the relevant error threshold.

Sample Usage: None

3.2.11 ResourceErrorsCorrected

Indicates that a specified resource property has corrected errors. Examples of this are drive I/O errors or network link errors.

- This message shall indicate that a specified resource property has corrected errors.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: OK

Resolution: None.

Message and Arguments: “The resource property has corrected errors of type “.”

1. *string*: The name of the property.
 - This argument shall contain the name of the relevant property from a Redfish resource.
2. *string*: The type of error encountered.
 - This argument shall contain a description of the type of error encountered.

Sample Usage: None

3.2.12 ResourceErrorsDetected

Indicates that a specified resource property has detected errors. Examples of this are drive I/O errors or network link errors.

- This message shall indicate that a specified resource property has detected errors.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: Resolution is dependent upon error type.

Message and Arguments: “The resource property has detected errors of type “.”

1. *string*: The name of the property.
 - This argument shall contain the name of the relevant property from a Redfish resource.
2. *string*: The type of error encountered.
 - This argument shall contain a description of the type of error encountered.

Sample Usage: None

3.2.13 ResourcePaused

Indicates that the power state of a resource has changed to paused.

- This message shall indicate that the value of the PowerState property of a resource has changed to *Paused*.

Version Added: 1.1.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: OK

Resolution: None.

Message and Arguments: “The resource “ was paused.”

1. *string*: The relevant resource.
 - This argument shall contain the name of the relevant Redfish resource.

Sample Usage: None

3.2.14 ResourcePoweredOff

Indicates that the power state of a resource has changed to powered off.

- This message shall indicate that the value of the PowerState property of a resource has changed to *Off*.

Version Added: 1.1.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: OK

Resolution: None.

Message and Arguments: “The resource ” has powered off.”

1. *string*: The relevant resource.
 - This argument shall contain the name of the relevant Redfish resource.

Sample Usage: None

3.2.15 ResourcePoweredOn

Indicates that the power state of a resource has changed to powered on.

- This message shall indicate that the value of the PowerState property of a resource has changed to On.

Version Added: 1.1.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: OK

Resolution: None.

Message and Arguments: “The resource ” has powered on.”

1. *string*: The relevant resource.
 - This argument shall contain the name of the relevant Redfish resource.

Sample Usage: None

3.2.16 ResourcePoweringOff

Indicates that the power state of a resource has changed to powering off.

- This message shall indicate that the value of the PowerState property of a resource has changed to PoweringOff. This message should not be used if the PowerState transition to Off will occur immediately.

Version Added: 1.1.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: OK

Resolution: None.

Message and Arguments: “The resource “ is powering off.”

1. *string*: The relevant resource.
 - This argument shall contain the name of the relevant Redfish resource.

Sample Usage: None

3.2.17 ResourcePoweringOn

Indicates that the power state of a resource has changed to powering on.

- This message shall indicate that the value of the PowerState property of a resource has changed to `PoweringOn`. This message should not be used if the PowerState transition to `On` will occur immediately.

Version Added: 1.1.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: OK

Resolution: None.

Message and Arguments: “The resource “ is powering on.”

1. *string*: The relevant resource.
 - This argument shall contain the name of the relevant Redfish resource.

Sample Usage: None

3.2.18 ResourceRemoved

Indicates that all conditions of a successful remove operation were met.

- This message shall indicate that all conditions of a successful remove operation were met.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: OK

Resolution: None.

Message and Arguments: “The resource was removed successfully.”

Sample Usage: None

3.2.19 ResourceSelfTestCompleted

Indicates that a self-test has completed.

- This message shall indicate that a self-test has completed.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: OK

Resolution: None.

Message and Arguments: “A self-test has completed.”

Sample Usage: None

3.2.20 ResourceSelfTestFailed

Indicates that a self-test has failed. Suggested resolution may be provided as OEM data.

- This message shall indicate that a self-test has failed.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Critical

Resolution: See vendor specific instructions for specific actions.

Message and Arguments: “A self-test has failed. The following message was returned: “.”

1. *string*: The self-test error message.
 - This argument shall contain the error message received as a result from the self-test.

Sample Usage: None

3.2.21 ResourceStateChanged

Indicates that the state of a resource has changed.

- This message shall indicate that the value of the State property of a resource has changed.

Version Added: 1.1.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: OK

Resolution: None.

Message and Arguments: “The state of resource ” has changed to .”

1. *string*: The relevant resource.
 - This argument shall contain the name of the relevant Redfish resource.
2. *string*: The state of the resource.
 - This argument shall contain the value of the State property for the relevant Redfish resource. The values shall be used from the definition of the State enumeration in the Resource schema.

Sample Usage: None

3.2.22 ResourceStatusChangedCritical

Indicates that the health of a resource has changed to Critical.

- This message shall indicate that the health of a resource has changed to Critical.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Critical

Resolution: None.

Message and Arguments: “The health of resource ” has changed to .”

1. *string*: The relevant resource.
 - This argument shall contain the name of the relevant Redfish resource.
2. *string*: The state of the resource.
 - This argument shall contain the value of the `Health` property for the relevant Redfish resource. The values shall be used from the definition of the `Health` enumeration in the `Resource` schema.

Sample Usage: None

3.2.23 ResourceStatusChangedOK

Indicates that the health of a resource has changed to OK.

- This message shall indicate that the health of a resource has changed to OK.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: OK

Resolution: None.

Message and Arguments: “The health of resource ” has changed to .”

1. *string*: The relevant resource.
 - This argument shall contain the name of the relevant Redfish resource.
2. *string*: The state of the resource.
 - This argument shall contain the value of the `Health` property for the relevant Redfish resource. The values shall be used from the definition of the `Health` enumeration in the `Resource` schema.

Sample Usage: None

3.2.24 ResourceStatusChangedWarning

Indicates that the health of a resource has changed to Warning.

- This message shall indicate that the health of a resource has changed to Warning.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: None.

Message and Arguments: “The health of resource ” has changed to .”

1. *string*: The relevant resource.
 - This argument shall contain the name of the relevant Redfish resource.
2. *string*: The state of the resource.
 - This argument shall contain the value of the `Health` property for the relevant Redfish resource. The values shall be used from the definition of the `Health` enumeration in the `Resource` schema.

Sample Usage: None

3.2.25 ResourceVersionIncompatible

Indicates that an incompatible version of software was detected. Examples of this are after a component or system-level software update.

- This message shall indicate that an incompatible version of software was detected.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: Compare the version of the resource with the compatible version of the software.

Message and Arguments: “An incompatible version of software ” was detected.”

1. *string*: The name of the software component.
 - This argument shall contain the name of the relevant software component or package. This might include both name and version information.

Sample Usage: None

3.2.26 ResourceWarningThresholdCleared

Indicates that a specified resource property has cleared its warning threshold. Examples of this are drive I/O errors or network link errors. Suggested resolution may be provided as OEM data.

- This message shall indicate that a specified resource property has cleared its warning threshold.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: OK

Resolution: None.

Message and Arguments: “The resource property has cleared the warning threshold of value .”

1. *string*: The name of the property.
 - This argument shall contain the name of the relevant property from a Redfish resource.
2. *number*: The value of the threshold.
 - This argument shall contain the value of the relevant error threshold.

Sample Usage: None

3.2.27 ResourceWarningThresholdExceeded

Indicates that a specified resource property has exceeded its warning threshold. Examples of this are drive I/O errors or network link errors. Suggested resolution may be provided as OEM data.

- This message shall indicate that a specified resource property has exceeded its warning threshold.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: None.

Message and Arguments: “The resource property has exceeded its warning threshold of value .”

1. *string*: The name of the property.
 - This argument shall contain the name of the relevant property from a Redfish resource.

2. *number*: The value of the threshold.
 - This argument shall contain the value of the relevant error threshold.

Sample Usage: None

3.2.28 TestMessage

A test message used to validate event delivery mechanisms.

- This message shall indicate a test of an event delivery mechanism, such as the creation of a LogEntry resource or an event transmission.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: OK

Resolution: None.

Message and Arguments: "Test message."

Sample Usage: None

3.2.29 URIForResourceChanged

Indicates that the URI for a resource has changed. Examples of this are physical component replacement or redistribution.

- This message shall indicate that the URI for a resource has changed.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: OK

Resolution: None.

Message and Arguments: "The URI for the resource has changed."

Sample Usage: None

3.3 Storage 1.0.0

3.3.1 Overview

This registry defines the event messages for Swordfish Storage devices. This replaces the Swordfish message registry.

The messages defined in this version of the Storage registry are summarized in Table 6.

Table 6: Storage messages

Message	Severity	Description
BackgroundOperationComplete	OK	Indicates that a background operation has completed.
BackgroundOperationFailed	Warning	Indicates that a background operation failed.
BackgroundOperationResumed	OK	Indicates that a background operation has resumed.
BackgroundOperationSuspended	Warning	Indicates that a background operation has been suspended.
LowSpaceWarningThresholdCleared	OK	Indicates that a LowSpaceWarningThreshold has been cleared.
LowSpaceWarningThresholdTriggered	Warning	Indicates that a LowSpaceWarningThreshold has been triggered.
ReservationsLostAcrossPowerCycle	OK	Indicates that persistent reservation key and holder information was lost during powercycle.

3.3.2 BackgroundOperationComplete

Indicates that a background operation has completed.

- Indicates that a background operation has completed.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: OK

Resolution: None

Message and Arguments: “The background operation <Arg1> has completed.”

1. *string*: The name of the background `Operation` that has completed.
 - This argument shall contain the name of the background task that has completed, typically defined by the `Operation` property or `OperationType` enum.

Sample Usage: None

3.3.3 BackgroundOperationFailed

Indicates that a background operation failed.

- Indicates that a background operation failed.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: Check device status.

Message and Arguments: “The background operation <Arg1> failed.”

1. *string*: The name of the background `Operation` that failed.
 - This argument shall contain the name of the background task that failed, typically defined by the `Operation` property or `OperationType` enum.

Sample Usage: None

3.3.4 BackgroundOperationResumed

Indicates that a background operation has resumed.

- Indicates that a background operation has resumed.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: OK

Resolution: None

Message and Arguments: “The background operation <Arg1> has resumed.”

1. *string*: The name of the background `Operation` that has resumed.
 - This argument shall contain the name of the background task that has resumed, typically defined by the `Operation` property or `OperationType` enum.

Sample Usage: None

3.3.5 BackgroundOperationSuspended

Indicates that a background operation has been suspended.

- Indicates that a background operation has been suspended.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: Resume or terminate the task

Message and Arguments: “The background operation <Arg1> has been suspended.”

1. *string*: The name of the background `Operation` that has been suspended.
 - This argument shall contain the name of the background task that has been suspended, typically defined by the `Operation` property or `OperationType` enum.

Sample Usage: None

3.3.6 LowSpaceWarningThresholdCleared

Indicates that a `LowSpaceWarningThreshold` has been cleared.

- Indicates that a `LowSpaceWarningThreshold` has been cleared.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: OK

Resolution: None.

Message and Arguments: “The LowSpaceWarningThreshold <Arg1> has been cleared on <Arg2>.”

1. *number*: threshold
 - This argument shall contain the threshold value that has been cleared.
2. *string*: resource
 - This argument shall contain the Resource reporting the threshold cleared.

Sample Usage: None

3.3.7 LowSpaceWarningThresholdTriggered

Indicates that a LowSpaceWarningThreshold has been triggered.

- Indicates that a LowSpaceWarningThreshold has been triggered. For some resource types, adding more capacity may resolve the low space warning. Other suggested resolutions may be provided as OEM data.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: Resolution dependent upon resource type.

Message and Arguments: “The LowSpaceWarningThreshold <Arg1> has been triggered on <Arg2>.”

1. *number*: Threshold
 - This argument shall contain the threshold value that has been exceeded.
2. *string*: Resource
 - This argument shall contain the Resource reporting the threshold trigger.

Sample Usage: None

3.3.8 ReservationsLostAcrossPowerCycle

Indicates that persistent reservation key and holder information was lost during powercycle.

- Indicates that persistent reservation key and holder information was lost during powercycle.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: OK

Resolution: Reservations must be reconfigured in-band.

Message and Arguments: “Persistent reservation key and holder information was lost during powercycle.”

Sample Usage: None

3.4 StorageCoS 1.0.0

3.4.1 Overview

This registry defines the event messages for Swordfish Storage Class of Service based implementations.

The messages defined in this version of the StorageCoS registry are summarized in Table 7.

Table 7: StorageCoS messages

Message	Severity	Description
ClassOfServiceOutOfCompliance	Warning	Indicates that the class of service is out of compliance.
LineOfServiceOutOfCompliance	Warning	Indicates that the line of service is out of compliance.
OutOfCompliance	Warning	Indicates that the service level requirements are out of compliance.

3.4.2 ClassOfServiceOutOfCompliance

Indicates that the class of service is out of compliance.

- Indicates that the class of service is out of compliance.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: Resolution dependent upon class of service and resource type.

Message and Arguments: “The service level requirements of the <Arg1> class of service are out of compliance.”

1. *string*: The name of the `ClassOfService` that is out of compliance.
 - This argument shall contain the name of the `ClassOfService` that is out of compliance.

Sample Usage: None

3.4.3 LineOfServiceOutOfCompliance

Indicates that the line of service is out of compliance.

- Indicates that the line of service is out of compliance.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: Resolution dependent upon line of service and resource type.

Message and Arguments: “The service level requirements of the <Arg1> line of service are out of compliance.”

1. *string*: The name of the `LineOfService` that is out of compliance.
 - This argument shall contain the name of the `LineOfService` that is out of compliance.

Sample Usage: None

3.4.4 OutOfCompliance

Indicates that the service level requirements are out of compliance.

- Indicates that the line of service is out of compliance.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: Resolution dependent upon service level and resource type.

Message and Arguments: “The service level requirements of <Arg1> are out of compliance.”

1. *string*: The name of the `LineOfService` that has service level requirements out of compliance.
 - This argument shall contain the name of the `LineOfService` that has service level requirements out of compliance.

Sample Usage: None

3.5 StorageConnection 1.0.0

3.5.1 Overview

This registry defines the event messages for Swordfish Connection usage.

The messages defined in this version of the StorageConnection registry are summarized in Table 8.

Table 8: StorageConnection messages

Message	Severity	Description
ConnectionMappingAdded	OK	Indicates that a new connection mapping (VolumeInfo entry) has been added to the Connection.
ConnectionMappingModified	OK	Indicates that a connection mapping (VolumeInfo entry) has been modified on the Connection.
ConnectionMappingRemoved	OK	Indicates that a connection mapping (VolumeInfo entry) has been removed from the Connection.

3.5.2 ConnectionMappingAdded

Indicates that a new connection mapping (VolumeInfo entry) has been added to the Connection.

- Indicates that a new connection mapping (VolumeInfo entry) has been added to the Connection.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: OK

Resolution: None.

Message and Arguments: "Connection mapping for Volume <Arg1> has been added."

1. *string*: The ID of the Volume related to the Connection.
 - This argument shall contain the ID of the Volume that has been added to the Connection.

Sample Usage: None

3.5.3 ConnectionMappingModified

Indicates that a connection mapping (VolumeInfo entry) has been modified on the Connection.

- Indicates that a connection mapping (VolumeInfo entry) has been modified on the Connection.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: OK

Resolution: None.

Message and Arguments: "Connection mapping for Volume <Arg1> has been modified."

1. *string*: The ID of the Volume related to the Connection.
 - This argument shall contain the ID of the Volume that has been modified on the Connection.

Sample Usage: None

3.5.4 ConnectionMappingRemoved

Indicates that a connection mapping (VolumeInfo entry) has been removed from the Connection.

- Indicates that a connection mapping (VolumeInfo entry) has been removed from the Connection.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: OK

Resolution: None.

Message and Arguments: “Connection mapping for Volume <Arg1> has been removed.”

1. *string*: The ID of the Volume related to the Connection.
 - This argument shall contain the ID of the Volume that has been removed from the Connection.

Sample Usage: None

3.6 StorageDevice 1.5.0

3.6.1 Overview

This registry defines the messages for storage devices.

The messages defined in this version of the StorageDevice registry are summarized in Table 9.

Table 9: StorageDevice messages

Message	Severity	Description
BatteryCharging	Warning	A battery charging condition was detected.
BatteryFailure	Critical	A battery failure condition was detected.
BatteryMissing	Critical	A battery missing condition was detected.
BatteryOK	OK	The health of a battery has changed to OK.

Message	Severity	Description
BlockSecurityIDFailed	Warning	The TCG-defined command to block establishment of a security ID failed.
BlockSecurityIDIssued	OK	The TCG-defined command to block establishment of a security ID was issued.
BlockSecurityIDRemoved	OK	TCG-defined commands to block establishment of a security ID are no longer being sent to a drive.
ControllerDegraded	Warning	A storage controller degraded condition was detected.
ControllerFailure	Critical	A storage controller failure was detected.
ControllerOK	OK	The storage controller health has changed to OK.
ControllerPasswordAccepted	OK	The storage controller password was entered.
ControllerPasswordRequired	Critical	The storage controller requires a password.
ControllerPortDegraded	Warning	A controller port degraded condition was detected.
ControllerPortFailure	Critical	A controller port failure condition was detected.
ControllerPortOK	OK	The health of a controller port has changed to OK.
ControllerPreviousError	Warning	A storage controller error was detected prior to reboot.
DriveFailure	Critical	A drive failure condition was detected.
DriveFailureCleared	OK	A previously detected failure condition on a drive was cleared.
DriveInserted	OK	A drive was inserted.

Message	Severity	Description
DriveMissing	Critical	A drive missing condition was detected.
DriveMissingCleared	OK	A previous drive missing condition was cleared.
DriveOK	OK	The health of a drive has changed to OK.
DriveOffline	Critical	A drive offline condition was detected.
DriveOfflineCleared	OK	A drive offline condition was cleared.
DrivePredictedMediaLifeLow	Warning	The drive media predicted life left is low.
DrivePredictiveFailure	Warning	A predictive drive failure condition was detected.
DrivePredictiveFailureCleared	OK	A previously detected predictive failure condition on a drive was cleared.
DriveRemoved	Critical	A drive was removed.
VolumeDegraded	Warning	The storage controller has detected a degraded volume condition.
VolumeFailure	Critical	The storage controller has detected a failed volume condition.
VolumeOK	OK	A volume health has changed to OK.
VolumeOffline (Deprecated)	Critical	The storage controller has detected an offline volume condition.
VolumeOffline	Critical	The storage controller has detected an offline volume condition.
VolumeOfflineCleared	OK	The storage controller has detected an online volume condition.
VolumeRebuilding	Warning	The storage controller is in the process of rebuilding the volume as part of a

Message	Severity	Description
		repair process. Performance might be degraded while the rebuild is in progress.
VolumeReconfiguring	Warning	The storage controller is in the process of reconfiguring the volume as part of adding new drives or changing RAID levels. Performance might be degraded while the reconfiguration is in progress.
WriteCacheDataLoss	Critical	The write cache is reporting loss of data in posted-writes memory.
WriteCacheDegraded	Critical	The write cache state is degraded.
WriteCacheProtected	OK	A storage controller write cache state is in protected mode.
WriteCacheTemporarilyDegraded	Warning	The write cache state is temporarily degraded.

3.6.2 BatteryCharging

A battery charging condition was detected.

- This message shall indicate a battery is below the required capacity and is charging.

Version Added: 1.1.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: None. There may be reduced performance and features while the battery is charging.

Message and Arguments: “A charging condition for the battery located in “ was detected.”

1. *string*: The location of the battery.

- This argument shall contain the battery location.

Sample Usage: None

3.6.3 BatteryFailure

A battery failure condition was detected.

- This message shall indicate a battery failure condition was detected.

Version Added: 1.1.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Critical

Resolution: Ensure all cables are properly and securely connected. Replace the failed battery.

Message and Arguments: “A failure condition for the battery located in “ was detected.”

1. *string*: The location of the battery.
 - This argument shall contain the battery location.

Sample Usage: None

3.6.4 BatteryMissing

A battery missing condition was detected.

- This message shall indicate a battery that is required by the configuration is missing.

Version Added: 1.1.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Critical

Resolution: Attach the battery. Ensure all cables are properly and securely connected.

Message and Arguments: “The battery located in “ is missing.”

1. *string*: The location of the battery.
 - This argument shall contain the battery location.

Sample Usage: None

3.6.5 BatteryOK

The health of a battery has changed to OK.

- This message shall indicate a battery health has changed to OK.

Version Added: 1.1.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: OK

Resolution: None.

Message and Arguments: “The health of the battery located in “ has changed to OK.”

1. *string*: The location of the battery.
 - This argument shall contain the battery location.

Sample Usage: None

3.6.6 BlockSecurityIDFailed

The TCG-defined command to block establishment of a security ID failed.

- This message shall indicate that the TCG-defined command to block establishment of a security ID failed when issued to a drive.

Version Added: 1.4.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: Update the drive firmware or replace the drive.

Message and Arguments: “The TCG-defined command to block establishment of a security ID failed when issued to the drive located in “.”

1. *string*: The location of the drive.
 - This argument shall contain the drive location.

Sample Usage: None

3.6.7 BlockSecurityIDIssued

The TCG-defined command to block establishment of a security ID was issued.

- This message shall indicate that the TCG-defined command to block establishment of a security ID was issued to a drive.

Version Added: 1.4.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: OK

Resolution: None.

Message and Arguments: “The TCG-defined command to block establishment of a security ID was issued to the drive located in ‘.’.”

1. *string*: The location of the drive.
 - This argument shall contain the drive location.

Sample Usage: None

3.6.8 BlockSecurityIDRemoved

TCG-defined commands to block establishment of a security ID are no longer being sent to a drive.

- This message shall indicate that the TCG-defined command to block establishment of a security ID are no longer sent to a drive.

Version Added: 1.4.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: OK

Resolution: None.

Message and Arguments: “The TCG-defined command to block establishment of a security ID are no longer sent to the drive located in ‘.’.”

1. *string*: The location of the drive.
 - This argument shall contain the drive location.

Sample Usage: None

3.6.9 ControllerDegraded

A storage controller degraded condition was detected.

- This message shall indicate that a storage controller degraded condition was detected.

Version Added: 1.1.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: Reseat the storage controller in the PCI slot. Update the controller to the latest firmware version. If the issue persists, replace the controller.

Message and Arguments: “A degraded condition for the storage controller located in “ was detected due to reason “.”

1. *string*: The location of the storage controller.
 - This argument shall contain the storage controller location.
2. *string*: The reason for the degraded controller.
 - This argument shall contain the reason for the degraded controller.

Sample Usage: None

3.6.10 ControllerFailure

A storage controller failure was detected.

- This message shall indicate that a storage controller failure condition was detected due to a hardware, communication, or firmware failure.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Critical

Resolution: Reseat the storage controller in the PCI slot. Update the controller to the latest firmware version. If the issue persists, replace the controller.

Message and Arguments: “A failure condition for the storage controller located in “ was detected.”

1. *string*: The location of the storage controller.
 - This argument shall contain the storage controller location.

Sample Usage: None

3.6.11 ControllerOK

The storage controller health has changed to OK.

- This message shall indicate that the storage controller health has changed to OK.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: OK

Resolution: None.

Message and Arguments: “The health of the storage controller located in “ has changed to OK.”

1. *string*: The location of the storage controller.
 - This argument shall contain the storage controller location.

Sample Usage: None

3.6.12 ControllerPasswordAccepted

The storage controller password was entered.

- This message shall indicate that the storage controller password was accepted. Normally this is used by storage controllers that require a controller password during boot to unlock encrypted volumes or drives.

Version Added: 1.1.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: OK

Resolution: None.

Message and Arguments: “A password was entered for the storage controller located in “.”

1. *string*: The location of the storage controller.
 - This argument shall contain the storage controller location.

Sample Usage: None

3.6.13 ControllerPasswordRequired

The storage controller requires a password.

- This message shall indicate that the storage controller requires a password. Normally this is used by storage controllers that require a controller password during boot to unlock encrypted volumes or drives.

Version Added: 1.1.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Critical

Resolution: Enter the controller password.

Message and Arguments: “The storage controller located in “ requires a password.”

1. *string*: The location of the storage controller.
 - This argument shall contain the storage controller location.

Sample Usage: None

3.6.14 ControllerPortDegraded

A controller port degraded condition was detected.

- This message shall indicate a controller port degraded condition was detected.

Version Added: 1.1.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: Ensure all cables are properly and securely connected. Replace faulty cables.

Message and Arguments: “A degraded condition for the controller port located in “ was detected due to reason “.”

1. *string*: The location of the controller port.
 - This argument shall contain the controller port location.
2. *string*: The reason for the degraded controller port.
 - This argument shall contain the reason for the degraded controller port.

Sample Usage: None

3.6.15 ControllerPortFailure

A controller port failure condition was detected.

- This message shall indicate a controller port failure condition was detected.

Version Added: 1.1.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Critical

Resolution: Ensure all cables are properly and securely connected. Replace faulty cables.

Message and Arguments: “A failure condition for the controller port located in “ was detected.”

1. *string*: The location of the controller port.
 - This argument shall contain the controller port location.

Sample Usage: None

3.6.16 ControllerPortOK

The health of a controller port has changed to OK.

- This message shall indicate a controller port health has changed to OK.

Version Added: 1.1.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: OK

Resolution: None.

Message and Arguments: “The health of the controller port located in “ has changed to OK.”

1. *string*: The location of the controller port.
 - This argument shall contain the controller port location.

Sample Usage: None

3.6.17 ControllerPreviousError

A storage controller error was detected prior to reboot.

- This message shall indicate that an unreported storage controller error condition was detected prior to reboot.

Version Added: 1.1.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: Update the controller to the latest firmware version. If the issue persists, replace the controller.

Message and Arguments: “A previous error condition for the storage controller located in “ was detected due to “.”

1. *string*: The location of the storage controller.
 - This argument shall contain the storage controller location.
2. *string*: The reason for error.
 - This argument shall contain the reason for failure.

Sample Usage: None

3.6.18 DriveFailure

A drive failure condition was detected.

- This message shall indicate a drive failure condition was detected due to a drive not installed correctly or failed.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Critical

Resolution: Ensure all cables are properly and securely connected. Ensure all drives are fully seated. Replace the defective cables, drive, or both.

Message and Arguments: “A failure condition for the drive located in “ was detected.”

1. *string*: The location of the drive.
 - This argument shall contain the drive location.

Sample Usage: None

3.6.19 DriveFailureCleared

A previously detected failure condition on a drive was cleared.

- This message shall indicate a failure condition on a drive was cleared, but any predictive failure or other conditions remain.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: OK

Resolution: None.

Message and Arguments: “A failure condition for the drive located in “ was cleared.”

1. *string*: The location of the drive.
 - This argument shall contain the drive location.

Sample Usage: None

3.6.20 DriveInserted

A drive was inserted.

- This message shall indicate a drive was inserted.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: OK

Resolution: If the drive is not properly displayed, attempt to refresh the cached data.

Message and Arguments: “The drive located in “ was inserted.”

1. *string*: The location of the drive.
 - This argument shall contain the drive location.

Sample Usage: None

3.6.21 DriveMissing

A drive missing condition was detected.

- This message shall indicate a drive that is required by the configuration is missing. This message shall exist any time a volume is degraded or failed due to a missing drive.

Version Added: 1.1.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Critical

Resolution: Ensure all cables are properly and securely connected. Ensure all drives are fully seated. Replace the defective cables, drive, or both. Delete the volume if it is no longer needed.

Message and Arguments: “A missing condition for the drive located in “ was detected.”

1. *string*: The location of the drive.
 - This argument shall contain the drive location.

Sample Usage: None

3.6.22 DriveMissingCleared

A previous drive missing condition was cleared.

- This message shall indicate a previous drive missing condition was cleared.

Version Added: 1.1.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: OK

Resolution: None.

Message and Arguments: “A missing condition for the drive located in “ was cleared.”

1. *string*: The location of the drive.
 - This argument shall contain the drive location.

Sample Usage: None

3.6.23 DriveOK

The health of a drive has changed to OK.

- This message shall indicate a drive health has changed to OK.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: OK

Resolution: None.

Message and Arguments: “The health of the drive located in “ has changed to OK.”

1. *string*: The location of the drive.
 - This argument shall contain the drive location.

Sample Usage: None

3.6.24 DriveOffline

A drive offline condition was detected.

- This message shall indicate a drive offline condition was detected. This may occur when a drive is unconfigured, encrypted by a foreign controller, offline during a background operation, or unsupported.

Version Added: 1.1.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Critical

Resolution: If the drive is unconfigured or needs an import, configure the drive. If the drive operation is in progress, wait for the operation to complete. If the drive is not supported, replace it.

Message and Arguments: “An offline condition for the drive located in “ was detected.”

1. *string*: The location of the drive.
 - This argument shall contain the drive location.

Sample Usage: None

3.6.25 DriveOfflineCleared

A drive offline condition was cleared.

- This message shall indicate a drive offline condition was cleared.

Version Added: 1.1.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: OK

Resolution: None.

Message and Arguments: “An offline condition for the drive located in “ was cleared.”

1. *string*: The location of the drive.
 - This argument shall contain the drive location.

Sample Usage: None

3.6.26 DrivePredictedMediaLifeLow

The drive media predicted life left is low.

- This message shall indicate a drive media predicted life left level is below a user or vendor-defined threshold.

Version Added: 1.3.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: Replace the drive before the predicted media life left reaches 0%.

Message and Arguments: “The drive located at <Arg1> has a predicted media life left of percent and is below the specified threshold of percent.”

1. *string*: The location of the drive.
 - This argument shall contain the drive location.
2. *number*: The percent of predicted media life left.
 - This argument shall contain the value of PredictedMediaLifeLeft for this drive.
3. *number*: The alert threshold.
 - This argument shall contain the value of the threshold that triggered this message.

Sample Usage: None

3.6.27 DrivePredictiveFailure

A predictive drive failure condition was detected.

- This message shall indicate a drive predictive failure condition was detected.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: If this drive is not part of a fault-tolerant volume, first back up all data, then replace the drive and restore all data afterward. If this drive is part of a fault-tolerant volume, replace this drive as soon as possible as long as the volume health is OK.

Message and Arguments: “A predictive failure condition for the drive located in “ was detected.”

1. *string*: The location of the drive.
 - This argument shall contain the drive location.

Sample Usage: None

3.6.28 DrivePredictiveFailureCleared

A previously detected predictive failure condition on a drive was cleared.

- This message shall indicate a predictive failure condition on a drive was cleared, but any other failure conditions remain.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: OK

Resolution: None.

Message and Arguments: “A predictive failure condition for the drive located in “ was cleared.”

1. *string*: The location of the drive.
 - This argument shall contain the drive location.

Sample Usage: None

3.6.29 DriveRemoved

A drive was removed.

- This message shall indicate a drive was removed.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Critical

Resolution: If the drive is still displayed, attempt to refresh the cached data.

Message and Arguments: “The drive located in “ was removed.”

1. *string*: The location of the drive.
 - This argument shall contain the drive location.

Sample Usage: None

3.6.30 VolumeDegraded

The storage controller has detected a degraded volume condition.

- This message shall indicate a volume degraded condition was detected due to one or more drives not installed correctly, missing, or failed.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: Ensure all cables are properly and securely connected. Replace failed drives.

Message and Arguments: “The volume ” attached to the storage controller located in ” is degraded.”

1. *string*: The identifier of the volume.
 - This argument shall contain an identifier for the volume. The identifier may contain `Id`, `Name`, `Identifiers`, `LogicalUnitNumber`, or other data used to uniquely identify the volume.
2. *string*: The location of the storage controller.
 - This argument shall contain the storage controller location.

Sample Usage: None

3.6.31 VolumeFailure

The storage controller has detected a failed volume condition.

- This message shall indicate a volume failure condition was detected due to one or more drives not installed correctly, missing, or failed.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Critical

Resolution: Ensure all cables are properly and securely connected. Ensure all drives are fully seated and operational.

Message and Arguments: “The volume ” attached to the storage controller located in “ has failed.”

1. *string*: The identifier of the volume.
 - This argument shall contain an identifier for the volume. The identifier may contain `Id`, `Name`, `Identifiers`, `LogicalUnitNumber`, or other data used to uniquely identify the volume.
2. *string*: The location of the storage controller.
 - This argument shall contain the storage controller location.

Sample Usage: None

3.6.32 VolumeOK

A volume health has changed to OK.

- This message shall indicate a volume health has changed to OK.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: OK

Resolution: None.

Message and Arguments: “The health of volume ” that is attached to the storage controller located in “ has changed to OK.”

1. *string*: The identifier of the volume.
 - This argument shall contain an identifier for the volume. The identifier may contain `Id`, `Name`, `Identifiers`, `LogicalUnitNumber`, or other data used to uniquely identify the volume.
2. *string*: The location of the storage controller.
 - This argument shall contain the storage controller location.

Sample Usage: None

3.6.33 VolumeOffline (Deprecated)

The storage controller has detected an offline volume condition.

- This message shall indicate a volume offline condition was detected. This may occur if the volume configuration is not supported or needs to be imported.

Version Added: 1.1.0

VersionDeprecated: 1.2.0

Deprecation Details: This message was deprecated in favor of `VolumeOffline` due to a typographic error in the `MessageId`.

MessageSeverity: Critical

Resolution: Use storage software to enable, repair, or import the volume. You may also delete or move volume back to the original controller.

Message and Arguments: “The volume “ attached to the storage controller located in “ is offline.”

1. *string*: The identifier of the volume.
 - This argument shall contain an identifier for the volume. The identifier may contain `Id`, `Name`, `Identifiers`, `LogicalUnitNumber`, or other data used to uniquely identify the volume.
2. *string*: The location of the storage controller.
 - This argument shall contain the storage controller location.

Sample Usage: None

3.6.34 VolumeOffline

The storage controller has detected an offline volume condition.

- This message shall indicate a volume offline condition was detected. This may occur if the volume configuration is not supported or needs to be imported.

Version Added: 1.2.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Critical

Resolution: Use storage software to enable, repair, or import the volume. You may also delete or move volume back to the original controller.

Message and Arguments: “The volume “ attached to the storage controller located in “ is offline.”

1. *string*: The identifier of the volume.
 - This argument shall contain an identifier for the volume. The identifier may contain `Id`, `Name`, `Identifiers`, `LogicalUnitNumber`, or other data used to uniquely identify the volume.

2. *string*: The location of the storage controller.
 - This argument shall contain the storage controller location.

Sample Usage: None

3.6.35 VolumeOfflineCleared

The storage controller has detected an online volume condition.

- This message shall indicate a volume online condition was detected.

Version Added: 1.1.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: OK

Resolution: None.

Message and Arguments: “The volume ” attached to the storage controller located in “ is online.”

1. *string*: The identifier of the volume.
 - This argument shall contain an identifier for the volume. The identifier may contain `Id`, `Name`, `Identifiers`, `LogicalUnitNumber`, or other data used to uniquely identify the volume.
2. *string*: The location of the storage controller.
 - This argument shall contain the storage controller location.

Sample Usage: None

3.6.36 VolumeRebuilding

The storage controller is in the process of rebuilding the volume as part of a repair process. Performance might be degraded while the rebuild is in progress.

- This message shall indicate that the storage controller is in the process of rebuilding the volume as part of a repair process. Performance might be degraded while the rebuild is in progress.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: Wait for the volume rebuild to complete.

Message and Arguments: “The volume ” attached to the storage controller located in ” is being rebuilt.”

1. *string*: The identifier of the volume.
 - This argument shall contain an identifier for the volume. The identifier may contain `Id`, `Name`, `Identifiers`, `LogicalUnitNumber`, or other data used to uniquely identify the volume.
2. *string*: The location of the storage controller.
 - This argument shall contain the storage controller location.

Sample Usage: None

3.6.37 VolumeReconfiguring

The storage controller is in the process of reconfiguring the volume as part of adding new drives or changing RAID levels. Performance might be degraded while the reconfiguration is in progress.

- This message shall indicate that the storage controller is in the process of reconfiguring the volume as part of adding new drives or changing RAID levels. Performance might be degraded while the reconfiguration is in progress.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: Wait for the volume reconfiguration to complete.

Message and Arguments: “The volume ” attached to the storage controller located in ” is being reconfigured.”

1. *string*: The identifier of the volume.
 - This argument shall contain an identifier for the volume. The identifier may contain `Id`, `Name`, `Identifiers`, `LogicalUnitNumber`, or other data used to uniquely identify the volume.
2. *string*: The location of the storage controller.
 - This argument shall contain the storage controller location.

Sample Usage: None

3.6.38 WriteCacheDataLoss

The write cache is reporting loss of data in posted-writes memory.

- This message shall indicate the write cache is reporting loss of data in posted-writes cache memory due to loss of battery power when the server lost power or a hardware error.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Critical

Resolution: Check the controller resource properties to determine the cause of the write cache data loss.

Message and Arguments: “The write cache on the storage controller located in “ has data loss.”

1. *string*: The location of the storage controller.
 - This argument shall contain the storage controller location.

Sample Usage: None

3.6.39 WriteCacheDegraded

The write cache state is degraded.

- This message shall indicate the write cache state degraded due to a missing battery or hardware failure.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Critical

Resolution: Check the controller to determine the cause of write cache degraded state, such as a missing battery or hardware failure.

Message and Arguments: “The write cache state on the storage controller located in “ is degraded.”

1. *string*: The location of the storage controller.
 - This argument shall contain the storage controller location.

Sample Usage: None

3.6.40 WriteCacheProtected

A storage controller write cache state is in protected mode.

- This message shall indicate a controller write cache state is in protected mode.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: OK

Resolution: None.

Message and Arguments: “The write cache state on the storage controller located in “ is in protected mode.”

1. *string*: The location of the storage controller.
 - This argument shall contain the storage controller location.

Sample Usage: None

3.6.41 WriteCacheTemporarilyDegraded

The write cache state is temporarily degraded.

- This message shall indicate write cache state temporarily degraded due to battery charging or data recovery rebuild operation pending.

Version Added: 1.0.0

VersionDeprecated: N/A

Deprecation Details: N/A

MessageSeverity: Warning

Resolution: Check the controller to determine the cause of write cache temporarily degraded state, such as a battery is charging or a data recovery rebuild operation is pending.

Message and Arguments: “The write cache state on the storage controller located in “ is temporarily degraded.”

1. *string*: The location of the storage controller.
 - This argument shall contain the storage controller location.

Sample Usage: None

3.7 Swordfish 1.1.0

3.7.1 Overview

This registry defines the event messages for Swordfish. This registry has been deprecated in favor of the Storage and StorageCoS registries.

The messages defined in this version of the Swordfish registry are summarized in Table 10.

Table 10: Swordfish messages

Message	Severity	Description
ClassOfServiceOutOfCompliance (Deprecated)	Warning	Indicates that the class of service is out of compliance.
LineOfServiceOutOfCompliance (Deprecated)	Warning	Indicates that the line of service is out of compliance.
LowSpaceWarningThresholdCleared (Deprecated)	OK	Indicates that a LowSpaceWarningThreshold has been cleared.
LowSpaceWarningThresholdTriggered (Deprecated)	Warning	Indicates that a LowSpaceWarningThreshold has been triggered.
NoDefaultValue (Deprecated)	Warning	Indicates that no default value was set.
OutOfCompliance (Deprecated)	Warning	Indicates that the service level requirements are out of compliance.
SearchFailed (Deprecated)	Warning	Indicates that a search failed.

3.7.2 ClassOfServiceOutOfCompliance (Deprecated)

Indicates that the class of service is out of compliance.

- The requirements for the named class of service have not been met.

Version Added: 1.0.0

VersionDeprecated: 1.1.0

Deprecation Details: This message was deprecated in favor of `ClassOfServiceOutOfCompliance` in the StorageCoS registry.

MessageSeverity: Warning

Resolution: Resolution dependent upon class of service and resource type.

Message and Arguments: “The service level requirements of the class of service are out of compliance.”

1. *string*: ClassOfService name
 - Name of the class of service that is out of compliance.

Sample Usage: None

3.7.3 LineOfServiceOutOfCompliance (Deprecated)

Indicates that the line of service is out of compliance.

- The requirements for the named line of service have not been met.

Version Added: 1.0.0

VersionDeprecated: 1.1.0

Deprecation Details: This message was deprecated in favor of `LineOfServiceOutOfCompliance` in the StorageCoS registry.

MessageSeverity: Warning

Resolution: Resolution dependent upon line of service and resource type.

Message and Arguments: “The service level requirements of the line of service are out of compliance.”

1. *string*: LineOfService name
 - Name of the line of service that is out of compliance.

Sample Usage: None

3.7.4 LowSpaceWarningThresholdCleared (Deprecated)

Indicates that a LowSpaceWarningThreshold has been cleared.

- A previously triggered low space condition has been resolved.

Version Added: 1.0.0

VersionDeprecated: 1.1.0

Deprecation Details: This message was deprecated in favor of the LowSpaceWarningThresholdCleared messages in Storage.

MessageSeverity: OK

Resolution: None.

Message and Arguments: “The LowSpaceWarningThreshold has been cleared on .”

1. *number*: Threshold value
 - The threshold value that has been exceeded.
2. *string*: Resource name
 - ResourceNameThe name of the resource for which the threshold value has been exceeded.

Sample Usage: None

3.7.5 LowSpaceWarningThresholdTriggered (Deprecated)

Indicates that a LowSpaceWarningThreshold has been triggered.

- Indicates that a LowSpaceWarningThreshold has been triggered. For some resource types, adding more capacity may resolve the low space warning. Other suggested resolutions may be provided as OEM data.

Version Added: 1.0.0

VersionDeprecated: 1.1.0

Deprecation Details: This message was deprecated in favor of the LowSpaceWarningThresholdTriggered messages in Storage.

MessageSeverity: Warning

Resolution: Resolution dependent upon resource type.

Message and Arguments: “The LowSpaceWarningThreshold has been triggered on .”

1. *number*: Threshold value
 - The threshold value that has been exceeded.
2. *string*: Resource name
 - The name of the resource for which the threshold value has been exceeded.

Sample Usage: None

3.7.6 NoDefaultValue (Deprecated)

Indicates that no default value was set.

- Indicates that no default value was set.

Version Added: 1.0.0

VersionDeprecated: 1.1.0

Deprecation Details: This message was deprecated in favor of `PropertyMissing`.

MessageSeverity: Warning

Resolution: Assign a default value for the property.

Message and Arguments: “No default value has been set for property <Arg1>.”

1. *string*: `PropertyName`
 - The name of the property that does not have a default property defined

Sample Usage: None

3.7.7 OutOfCompliance (Deprecated)

Indicates that the service level requirements are out of compliance.

- Indicates that the service level requirements are out of compliance.

Version Added: 1.0.0

VersionDeprecated: 1.1.0

Deprecation Details: This message was deprecated in favor of `OutOfCompliance` in the `StorageCoS` registry.

MessageSeverity: Warning

Resolution: Resolution dependent upon service level and resource type.

Message and Arguments: “The service level requirements of are out of compliance.”

1. *string*: resource name
 - Name of the resource that is out of compliance.

Sample Usage: None

3.7.8 SearchFailed (Deprecated)

Indicates that a search failed.

- Indicates that a search failed.

Version Added: 1.0.0

VersionDeprecated: 1.1.0

Deprecation Details: This message was deprecated in favor of the Query messages in Redfish base.

MessageSeverity: Warning

Resolution: Retry search with different parameters.

Message and Arguments: "Object <Arg1> not found that matched requested parameters."

1. *string*: Object type
 - The object type that was not found by the submitted query

Sample Usage: None

4 Error Messages

4.1 Overview

The tables in this clause summarize the errors defined in the Redfish base registry (version 1.19.0 for this release) that are most likely to be encountered in Swordfish implementations. ## General Errors

This group of error cases summarized in Table 11 deals with general error cases.

Errors in this group tend to be simple, non-specific mistakes or general notification that are handled through an error message, and do not reflect any problem with the storage system or its configuration.

Table 11: General Errors

Error Case	Common Error Cause(s)
NoOperation	The request will neither result in a change in the service, nor a change to a resource.

4.2 Action Errors

The group of error cases summarized in Table 12 deals with errors arising from actions.

Errors in this group tend to indicate invalid parameters, or a resource/parameter/action mismatch.

Table 12: Action errors

Error Case	Common Error Cause(s)
Action Not Supported	The requested action is not supported by the selected resource.
Action Parameter Duplicate	The body of the request contains duplicate parameter settings.
Action Parameter Missing	The action CreateReplicaTarget was submitted with the invalid parameter Capacity.
Action Parameter Not Supported	The parameter supplied for the action is not supported on the resource.
Action Parameter Unknown	The request contains unknown parameter settings.

Error Case	Common Error Cause(s)
Action Parameter Value Error	An invalid or unreachable URI or Resource is included as a parameter
Action Parameter Value Not In List Error	The request uses an ENUM value that is not defined for the target Resource.
Action Parameter Value Type Error	A parameter was given the wrong value type, such as when a number is supplied for a parameter that requires a string.

4.3 JSON Errors

The group of error cases summarized in Table 13 deals with mistakes in formatting the JSON required by an API request.

Errors in this group are simple syntactic mistakes or omissions in the API request, and do not reflect any problem with the storage system or its configuration.

Table 13: JSON Errors

Error Message	Common Error Cause(s)
Empty JSON	The request requires a JSON body, and none was included.
Invalid JSON	The JSON body could not be properly parsed.
Malformed JSON	The JSON body included in the request can be parsed, but contains invalid parameter values.
Unrecognized Request Body	Invalid request body content, such as a binary file

4.4 Property Errors

The group of error cases summarized in Table 14 deals with error in the selection or handling of property values.

Errors in this group tend to arise from property-level constraints in the schema or a particular implementation.

Table 14: Property errors

Error Message	Common Error Cause(s)
Property Duplicate	A property is inadvertently duplicated in the request body.
Property Missing	The request does not include all of the properties

Error Message	Common Error Cause(s)
	required to process it.
Property Not Updated	Internal error do to configuration conflict.
Property Not Writable	The request to change a single property, references a property that is read-only.
Property Unknown	An incorrect property name is used in the request body.
Property Value Conflict	The request uses an invalid combination of (valid) properties.
Property Value Not In List	The request uses a valid value type for a given property according to the Swordfish specification, but the implementation does not support that value.
Property Value Out of Range	An integer property value is too small or too large
Property Value Resource Conflict	Secondary or assoiated resources impacted by a query cannot be updated.
Property Value Type Error	A string value is provided when a numeric value is required.

4.5 Resource Errors

The group of error cases summarized in Table 15 deals with general error cases.

Errors in this group tend to be simple, non-specific mistakes or general notification that are handled through an error message, and do not reflect any problem with the storage system or its configuration.

Table 15: Resource errors

Error Message	Common Error Cause(s)
Resource Already Exists	A CREATE the implementation cannot be accepted, because the resource already exists.
Resource At URI In Unknown Format	A malformed or incorrect URI
Resource At URI Unauthorized – WIP	Add common cause of error
Resource Cannot Be Deleted	The named resource cannot be deleted.
Resource Creation Conflict	Recreation of an existing Resource
Resource Exhaustion	An attempt to expand a Volume beyond the underlying storage capacity.
Resource In Standby	Attempting to use a Resource that has been quiesced, or is not available for some other reason.
Resource In Use	The requested change cannot be completed

Error Message	Common Error Cause(s)
	because the resource is in use or in transition.
Resource Missing At URI	A malformed or incorrect URI
Resource Not Found	The DELETE request references a resource that cannot be found.
Resource Type Incompatible	Attempting to add to a <code>ResourceCollection</code> with an instance of the wrong <code>ResourceType</code>

5 Alphabetic List of Error Cases

5.1 Overview

The error cases summarized in this clause are a subset of the errors defined in the Redfish base registry (version 1.19.0 for this release) that are most likely to be encountered in Swordfish implementations.

5.2 *Error Case: Action Not Supported*

Summary: When a client sends a request to a Swordfish implementation and the action supplied with the POST operation is defined within the schema, but is not supported by the implementation, the `ActionNotSupported` message shall be returned.

Common Cause: The requested action is not supported by the selected resource.

Example: User tries to suspend replication on a Volume.

Basic Course of Events:

1. The user attempts to suspend replication on a Volume.

Request: POST /redfish/v1/StorageServices/ISC/Volumes/1/
Volume.SuspendReplication

Headers: No additional headers required.

Body:

```
{
  "TargetVolume": "/redfish/v1/Storage/1/Volumes/650973452245"
}
```

HTTP Status Code Returned: 400

Headers: No additional headers required.

Body:

```
{
  "error": {
    "code": "Base.1.21.ActionNotSupported",
    "message": "The action SuspendReplication is not supported by the resource."
  }
}
```

AdditionalContext: None.

5.3 Error Case: Action Parameter Duplicate

Summary: When a client sends a request to a Swordfish implementation and more than one value is provided for a parameter to the action, the `ActionParameterDuplicate` message shall be returned.

Common Cause: The body of the request contains duplicate parameter settings.

Example: User attempts to create a new volume to serve as a target replica for an existing source volume, but provides duplicate or conflicting values for `ReplicaUpdateMode`.

Basic Course of Events:

1. Post (as an Action) the request on the source Volume.

This instructs the service to use the identified Volume as the source Volume for the specified replication relationship. For any additional details required, the service will rely on default values.

Request: POST /redfish/v1/Storage/1/Volumes/1/Volume.CreateReplicaTarget

Headers: No additional headers required.

Body:

```
{
  "VolumeName" : "Mirror of Volume 65",
  "ReplicaUpdateMode" : "Synchronous",
  "ReplicaUpdateMode" : "Asynchronous",
  "TargetStoragePool" : "/redfish/v1/Storage/1/StoragePools/PrimaryPool",
  "ReplicaType" : "Mirror"
}
```

HTTP Status Code Returned: 400

Body:

```
{
  "error": {
    "code": "Base.1.21.ActionParameterDuplicate",
    "message": "The action CreateReplicaTarget was submitted with more than one
value for the parameter ReplicaUpdateMode."
  }
}
```

AdditionalContext: None.

5.4 Error Case: Action Parameter Missing

Summary: When a client sends a request to a Swordfish implementation, but omits one or more of the required parameters for the action, the `ActionParameterMissing` message shall be returned.

Common Cause: The action `CreateReplicaTarget` was submitted with the invalid parameter `Capacity`.

Example: User attempts to create a new volume to serve as a target replica for an existing source volume, but fails to specify the `TargetStoragePool` for the new Volume.

Basic Course of Events:

1. Post (as an Action) the request on the source Volume.

This instructs the service to use the identified Volume as the source Volume for the specified replication relationship. For any additional details required, the service will rely on default values.

Request: POST /redfish/v1/Storage/1/Volumes/1/Volume.CreateReplicaTarget

Headers: No additional headers required.

Body:

```
{
  "ReplicaUpdateMode" : "Synchronous",
  "ReplicaType" : "Mirror"
}
```

HTTP Status Code Returned: 400

Headers: No additional headers required.

Body:

```
{
  "error": {
    "code": "Base.1.21.ActionParameterMissing",
    "message": "The action CreateReplicaTarget requires the parameter
    TargetStoragePool to be present in the request body."
  }
}
```

AdditionalContext: None.

5.5 Error Case: Action Parameter Not Supported

Summary: When a client invokes an action against a Resource, but the body of the request include a parameter that is not supported by the current implementation, the `ActionParameterNotSupported` message shall be returned.

Common Cause: The parameter supplied for the action is not supported on the resource.

Example: The user attempts to remove a replica relationship, and includes the `DeleteTargetVolume` property, which is not supported in the current implementation.

Basic Course of Events:

1. The user attempts to delete the replication relationship.

Request: POST /redfish/v1/Storage/Volumes/1/Volume.RemoveReplicaRelationship

Headers: No additional headers required.

Body:

```
{
  "TargetVolume" : "/redfish/v1/Storage/1/Volumes/42524988",
  "DeleteTargetVolume" : true
}
```

HTTP Status Code Returned: 400

Headers: No additional headers required.

Body:

```
{
```

```
"error": {
  "code": "Base.1.21.ActionParameterNotSupported",
  "message": "The parameter DeleteTargetVolume for the action
  RemoveReplicaRelationship is not supported on the target resource."
}
```

AdditionalContext: None.

5.6 Error Case: Action Parameter Unknown

Summary: When a client sends a request to a Swordfish implementation, but includes an unknown parameter in the action, the `ActionParameterUnknown` message shall be returned.

Common Cause: The request contains unknown parameter settings.

Example: User attempts to create a new volume to serve as a target replica for an existing source volume, but includes an unknown parameter (“capacity”) in the body of the request.

Basic Course of Events:

1. Post (as an Action) the request on the source Volume.

Request: POST /redfish/v1/Storage/1/Volumes/1/Volume.CreateReplicaTarget

Headers: No additional headers required.

Body:

```
{
  "Capacity": 100002334153,
  "ReplicaUpdateMode" : "Synchronous",
  "TargetStoragePool" : "/redfish/v1/Storage/1/StoragePools/PrimaryPool",
  "ReplicaType" : "Mirror"
}
```

HTTP Status Code Returned: 400

Headers: No additional headers required.

Body:

```
{
```

```
"error": {
  "code": "Base.1.21.ActionParameterUnknown",
  "message": "The action CreateReplicaTarget was submitted with the invalid
parameter Capacity."
}
```

AdditionalContext: None.

5.7 Error Case: Action Parameter Value Error

Summary: When a client invokes an action, but an included parameter contains an invalid value, the `ActionParameterValueError` message shall be returned.

Common Cause: An invalid or unreachable URI or `Resource` is included as a parameter

Error Group: ActionErrors

Example: User tries to update firmware using an unreachable URI for the new image.

Basic Course of Events:

1. Call the FirmwareUpdate service

Request:

```
POST /redfish/v1/Systems/Sys-1/Storage/SimplestNVMeSSD/Controllers/
NVMeIOController/Actions/UpdateService.SimpleUpdate
```

- **Headers:** Content-type : application/json
- **Body:**

```
{
  "ImageURI": "http://invalid-server.local/firmware.bin"
}
```

HTTP Status Code Returned: 400

Headers: No additional headers required.

Body:

```

{
  "error": {
    "code": "Base.1.21.ActionParameterValueError",
    "Message": "The value for the parameter %21 in the action
UpdateService.SimpleUpdate is invalid.",
  }
}

```

AdditionalContext: None.

5.8 Error Case: Action Parameter Value Not In List Error

Summary: When a client invokes an action using a parameter of the correct type, but the implementation does not support the selected value, the ActionParameterValueNotInList message shall be returned.

Common Cause: The request uses an ENUM value that is not defined for the target Resource.

Error Group: ActionErrors

Example: User requests an unsupported replication type on AssignReplicaTarget

Inputs:

- URL for target volume: /redfish/v1/Storage/1/Volumes/650973452245
- Requested replica type: TokenizedClone
- ReplicaUpdateMode: Synchronous

Basic Course of Events:

1. Post (as an Action) the request on the source Volume.

Request: POST /redfish/v1/Storage/1/Volumes POST /redfish/v1/Storage/1/Volumes/1/Volume.AssignReplicaTarget

Headers: No additional headers required.

Body:

```

{
  "ReplicaUpdateMode": "Synchronous",
  "TargetVolume": "/redfish/v1/Storage/1/Volumes/650973452245",
}

```

```
"ReplicaType": "TokenizedClone"
}
```

HTTP Status Code Returned: 501 Not Implemented

Headers: None

Body:

```
{
  "error": {
    "code": "Base.1.10.ActionParameterValueNotInList",
    "message": "The value 'TokenizedClone' for the parameter ReplicaType in the action
               AssignReplicaType is not in the list of acceptable values."
  }
}
```

Additional Context: None.

5.9 Error Case: Action Parameter Value Type Error

Summary: When a client sends a request to a Swordfish implementation, but uses a wrong value type for one or more parameter(s), the `ActionParameterValueTypeError` message shall be returned.

Common Cause: A parameter was given the wrong value type, such as when a number is supplied for a parameter that requires a string.

Example: User attempts to create a new volume to serve as a target replica for an existing source volume, but specifies an integer value for the `VolumeName`.

Basic Course of Events:

1. Post (as an Action) the request on the source Volume.

This instructs the service to use the identified Volume as the source Volume for the specified replication relationship. For any additional details required, the service will rely on default values.

Request: POST `/redfish/v1/Storage/1/Volumes/1/Volume.CreateReplicaTarget`

Headers: No additional headers required.

Body:

```
{
  "VolumeName": 123456,
  "ReplicaUpdateMode" : "Synchronous",
  "TargetStoragePool" : "/redfish/v1/Storage/1/StoragePools/PrimaryPool",
  "ReplicaType" : "Mirror"
}
```

HTTP Status Code Returned: 400

Headers: No additional headers required.

Body:

```
{
  "error": {
    "code": "Base.1.21.ActionParameterValueTypeError",
    "message": "The value 123456 for the parameter VolumeName in the action
      CreateReplicaTarget is of a different type than the parameter can accept."
  }
}
```

AdditionalContext: None.

5.10 Error Case: Empty JSON

Summary: When a client sends a request to a Swordfish implementation, but fails to include any properties required to process the request when one or more properties are expected, the `EmptyJSON` message shall be returned.

Common Cause: The request requires a JSON body, and none was included.

Example: User tries to create a Volume, but omits any properties.

Basic Course of Events:

1. Post the definition of the new volume to the Volumes resource collection with no Body.

Request: POST /redfish/v1/Storage/1/Volumes

Headers: No additional headers required.

Body:

```
{  
  
}
```

HTTP Status Code Returned: 400

Headers: No additional headers required.

Body:

```
{  
  "error": {  
    "code": "Base.1.6.EmptyJSON",  
    "Message": "The request body submitted contained an empty JSON object and the  
               service is unable to process it."  
  }  
}
```

AdditionalContext: Note: The EmptyJSON case is technically a subset case of the PropertyMissing case. This may end up deprecated from the Redfish Message Registry in time and replaced by pointers to Property Missing instead.

If desired, the implementation can return an ExtendedInfo structure that includes the information about the specific issue (in this case, a pointer to the duplicate CapacityBytes property).

```
{  
  "error": {  
    "code": "Base.1.21.EmptyJSON",  
    "Message": "The request body submitted contained an empty JSON object and the  
               service is unable to process it.",  
    "@Message.ExtendedInfo": [  
      {  
        "@odata.type": "#Message.v1_0_0.Message",  
        "MessageId": "Base.1.21.PropertyMissing",  
        "RelatedProperties": [  
          "#/CapacityBytes"  
        ],  
        "Message": "The property CapacityBytes is a required property and must be  
                   included in the request.",  
        "MessageArgs": [  
          "CapacityBytes"  
        ],  
        "Severity": "Warning",  
        "Resolution": "Ensure that the property is in the request body and has a  
                      valid value and resubmit the request if the operation failed."  
      }  
    ]  
  }  
}
```

```
}  
]  
}  
}
```

AdditionalContext: None.

5.11 Error Case: Invalid JSON

Summary: When a client sends a request to a Swordfish implementation, but the request does not contain valid JSON, `InvalidJSON` message shall be returned.

Common Cause: The JSON body could not be property parsed.

Error Group: JSONErrors

Example: User tries to create a Volume, but includes an array of RAID types, rather than a single string value.

Basic Course of Events:

1. Post the definition of the new volume to the Volumes resource collection with no Body.

Request: POST /redfish/v1/Storage/1/Volumes

Headers: No additional headers required.

Body:

```
{  
  "Name" : MyVolume",  
  "RAIDType" : "RAID1",  
  "CapacityBytes": 34576345685  
}
```

HTTP Status Code Returned: 400

Headers: No additional headers required.

Body:

```
{  
  "error": {
```

```
    "code": "Base.1.21.InvalidJSON",
    "Message": "The request body submitted is invalid JSON starting at line 2 and
could not be parsed by the receiving service."
  }
}
```

5.12 *Error Case: Malformed JSON*

Summary: When a client sends a request to a Swordfish implementation, but the request contains JSON that is valid (i.e., can be parsed) but fails to meet other requirements, the `MalformedJSON` message shall be returned.

Common Cause: The JSON body included in the request can be parse, but contains invalid paramter values.

Example: User tries to create a Volume, but includes a collection of RAID types, rather than a single RAID type.

Basic Course of Events:

1. Post the definition of the new volume to the Volumes resource collection without a valid RAIDType value.

Request: POST /redfish/v1/Storage/1/Volumes

Headers: No additional headers required.

Body:

```
{
  "Name" : "MyVolume",
  "RAIDType" : ["RAID1", "RAID10"],
  "CapacityBytes": 34576345685
}
```

HTTP Status Code Returned: 400

Headers: No additional headers required.

Body:

```
{
  "error": {
```

```
    "code": "Base.1.21.MalformedJSON",
    "Message": "The request body submitted was malformed JSON and could not be
    parsed by the receiving service."
  }
}
```

AdditionalContext: None.

5.13 *Error Case: NoOperation*

Summary: When a client sends a valid request to a Swordfish implementation, but that request will neither result in a change in the service, nor a change to the resource, the `NoOperation` message should be returned.

Common Cause: The request will neither result in a change in the service, nor a change to a resource.

Example: User tries to expand an existing Volume, but provides a new value equal to the existing size.

Basic Course of Events:

1. Post the definition of the new volume to the Volumes resource collection with no Body.

Request: PATCH /redfish/v1/Storage/1/Volumes

Headers: No additional headers required.

Body:

```
{
  "Name" : "MyVolume",
  "RAIDType" : "RAID1",
  "CapacityBytes": 23049823948
}
```

HTTP Status Code Returned: 400

Headers: No additional headers required.

Body:

```
{
  "error": {
    "code": "Base.1.21.NoOperation",
```

```
    "Message": "The request body submitted contain no data to act upon and no  
    changes to the resource took place."  
  }  
}
```

AdditionalContext: None.

5.14 *Error Case: Property Duplicate*

Summary: When a client sends a request to a Swordfish implementation, but includes multiple values for a required property, the `PropertyDuplicate` message shall be returned.

Common Cause: A property is inadvertently duplicated in the request body.

Error Group: `PropertyErrors`

Example: User tries to create a Volume, but provides two different capacity values.

Basic Course of Events:

1. Post the definition of the new volume to the `AllocatedVolumes` resource collection.

Request:

POST `/redfish/v1/Storage/1/StoragePools/PrimaryPool/AllocatedVolumes`

- **Headers:** `Content-type : application/json`
- **Body:**

```
{  
  "Name" : "MyVolume",  
  "CapacityBytes" : 1099511627776,  
  "CapacityBytes" : 1000000000000,  
  "RAIDType" : "RAID1"  
}
```

HTTP Status Code Returned: 400

Headers: No additional headers required.

Body:

```
{
  "error": {
    "code": "Base.1.21.PropertyDuplicate",
    "Message": "The property CapacityBytes was duplicated in the request."
  }
}
```

AdditionalContext: : None.

5.15 *Error Case: Property Missing*

Summary: When a client sends a request to a Swordfish implementation, but fails to include all of the properties required to process the request, the `PropertyMissing` message shall be returned.

Common Cause: The request does not include all of the properties required to process it.

Example: User tries to create a Volume, but omits the desired RAID level for the volume, which has no default level set.

Basic Course of Events:

1. Post the definition of the new volume to the Volumes resource collection but omit the capacity property.

Request: POST /redfish/v1/Storage/1/Volumes

Headers: No additional headers required.

Body:

```
{
  "Name" : "MyVolume",
  "Capacity": {
    "Data": {
      "Allocated": 1000000000
    }
  }
}
```

HTTP Status Code Returned: 400

Headers: No additional headers required.

Body:

```

{
  "error": {
    "code": "Base.1.21.PropertyMissing",
    "Message": "The property CapacityBytes is a required property and must be
    included in the request."
  }
}

```

AdditionalContext: In this example, we have chosen to specify a “required in context” property. This requirement is not explicitly noted in the Volume schema; it is instead noted in the general requirements for implementations, in the user’s guide examples, and in profile definitions.

5.16 *Error Case: Property Not Updated*

Summary: When a client update request fails to modify the target property, but the underlying service is otherwise functional, the `PropertyNotUpdated` message shall be returned.

Common Cause: Internal error do to configuration conflict.

Error Group: PropertyErrors

Example: User tries to enable compression on a `StoragePool` that is already in the process of being compressed.

Basic Course of Events:

1. Use the `SetCompressionState` action on the selected `StoragePool`.

Request: POST /redfish/v1/Storage/1/StoragePools/1/Actions/SetCompressionState

Headers: No additional headers required.

Body: ```json {
 “Enabled”: true }

```

```

```

**HTTP Status Code Returned:** 400

**Headers:** No additional headers required.

**Body:**

```
{
 "error": {
 "code": "Base.1.21.PropertyNotUpdated",
 "Message": "The property IsCompressed was not updated due to an internal
service error. The service is still operational.",
 }
}
```

**AdditionalContext:** None.

## 5.17 Error Case: Property Not Writable

**Summary:** When a client sends a request to a Swordfish implementation to change a **single** property, but the requested property is read-only, the `PropertyNotWritable` message shall be returned.

**Common Cause:** The request to change a single property, references a property that is read-only.

Example: User tries to set the Name property on a Volume.

### Basic Course of Events:

1. The user attempts to PATCH the Volume name.

**Request:** PATCH /redfish/v1/StorageServices/ISC/Volumes/1

**Headers:** No additional headers required.

### Body:

```
{
 "@Redfish.Copyright": "Copyright 2015-2019 SNIA. All rights reserved.",
 "@odata.context": "/redfish/v1/$metadata#Volume.Volume",
 "@odata.id": "/redfish/v1/StorageServices/ISC/Volumes/1",
 "@odata.type": "#Volume.v1_2_1.Volume",
 "Id": "1",
 "Name": "Danny's Volume"
}
```

**HTTP Status Code Returned:** 400

**Headers:** No additional headers required.

### Body:

```
{
 "error": {
 "code": "Base.1.21.PropertyValueNotWritable",
 "message": "The property Name is a read only property and cannot be assigned a
value."
 }
}
```

**AdditionalContext:** None.

## 5.18 Error Case: Property Unknown

**Summary:** When a client sends a request to a Swordfish implementation and includes an invalid property in the request body, the `PropertyUnknown` message shall be returned.

**Common Cause:** An incorrect property name is used in the request body.

**Error Group:** `PropertyErrors`

Example: User tries to create a Volume, but provides a `size` parameter, rather than `CapacityBytes`.

**Basic Course of Events:**

1. Post the definition of the new volume to the `AllocatedVolumes` resource collection.

**Request:**

POST /redfish/v1/Storage/1/StoragePools/PrimaryPool/AllocatedVolumes

- **Headers:** Content-type : application/json
- **Body:**

```
{
 "Name" : "MyVolume",
 "SizeBytes" : 1099511627776,
 "RAIDType" : "RAID1"
}
```

**HTTP Status Code Returned:** 400

**Headers:** No additional headers required.

**Body:**

```
{
 "error": {
 "code": "Base.1.21.PropertyUnknown",
 "Message": "The property SizeBytes is not in the list of valid properties for
the resource."
 }
}
```

**AdditionalContext:** : None.

## 5.19 *Error Case: Property Value Conflict*

**Summary:** When a client sends a request to a Swordfish implementation with an invalid combination of (valid) properties included in the request body, the `PropertyValueConflict` message shall be returned.

**Common Cause:** The request uses an invalid combination of (valid) properties.

**Example:** User attempts to set both FC and iSCSI parameters simultaneously.

**Inputs:**

- URL for Endpoint: `/redfish/v1/Storage/MidrangeStorageSystem/Endpoints/TargetEndpoint1`
- Set of properties that logically do not apply to a given endpoint (both FC and iSCSI properties)

**Basic Course of Events:**

1. The user attempts to PATCH an invalid combination of properties, both FC and iSCSI, to an Endpoint.

**Request:** PATCH `/redfish/v1/Storage/MidrangeStorageSystem/Endpoints/TargetEndpoint1`

**Headers:** No additional headers required.

**Body:**

```
{
 "EndpointProtocol": "FC",
 "IPTransportDetails": [{
 "IPv4Address": {
 "Address": "192.168.1.63"
 }
 }],
}
```

```
}
```

**HTTP Status Code Returned:** 400 Bad Request

**Headers:** None

**Body:**

```
{
 "error": {
 "code": "Base.1.21.PropertyValueConflict",
 "message": "The property 'IPTransportDetails' could not be written because its
 value would conflict with the value of the 'EndpointProtocol' property."
 }
}
```

**AdditionalContext:** : None.

## 5.20 *Error Case: Property Value Not In List*

**Summary:** When a client sends a request to a Swordfish implementation using a correct value type for a given property, but the implementation does not support the selected value, the `PropertyValueNotInList` message shall be returned.

**Common Cause:** The request uses a valid value type for a given property according to the Swordfish specification, but the implementation does not support that value.

Example: User requests an unsupported replication type on `AssignReplicaTarget`

**Inputs:**

- URL for target volume: `/redfish/v1/Storage/1/Volumes/650973452245`
- Requested replica type: `TokenizedClone`
- `ReplicaUpdateMode`: `Synchronous`

**Basic Course of Events:**

1. Post (as an Action) the request on the source Volume.

This instructs the service to use the identified Volume as the source Volume for the specified replication relationship. For any additional details required, the service will rely on default values.

**Request:** POST /redfish/v1/Storage/1/Volumes/1/Volume.AssignReplicaTarget

**Headers:** No additional headers required.

**Body:**

```
{
 "ReplicaUpdateMode": "Synchronous",
 "TargetVolume": "/redfish/v1/Storage/1/Volumes/650973452245",
 "ReplicaType": "`TokenizedClone`"
}
```

**HTTP Status Code Returned:** 501 Not Implemented

**Headers:** None

**Body:**

```
{
 "error": {
 "code": "Base.1.21.PropertyValueNotInList",
 "message": "The value TokenizedClone for the property ReplicaType is not in the
 list of acceptable values."
 }
}
```

**AdditionalContext:** None.

## 5.21 *Error Case: Property Value Out of Range*

**Summary:** When a client sends a request to a Swordfish implementation, but a properties value falls outside the supported range, the ValueOutOfRange message shall be returned.

**Common Cause:** An integer property value is too small or too large

**Error Group:** PropertyErrors

Example: User tries to create an Endpoint, but supplies a value for Port cannot be held in 16 bits.

**Basic Course of Events:**

1. Post the definition of the new volume to the Volumes resource collection with no Body.

**Request:** POST /redfish/v1/Fabric/Endpoints

**Headers:** No additional headers required.

**Body:**

```
{
 "Name": "InitiatorEndpoint1"
 "EntityRole": "Initiator"
 "EndpointProtocol": "iSCSI"
 "IPTransportDetails":
 {
 "IPv4Address": {
 "Address": "192.168.21.137"
 },
 "Port": 99999
 }
}

...

HTTP Status Code Returned: 400

Headers: No additional headers required.

Body:

```json
{
  "error": {
    "code": "Base.1.21.PropertyValueOutOfRange",
    "Message": "The value '9999' for the property Port is not in the supported range
               of acceptable values.",
  }
}
```

AdditionalContext: None.

5.22 Error Case: Property Value Resource Conflict

Summary: When a client sends a request to a Swordfish implementation, which attempts to update a property value, and the request cannot be completed due to the current state or configuration of another resource, the `PropertyValueResourceConflict` message shall be returned.

Common Cause: Secondary or associated resources impacted by a query cannot be updated.

Error Group: PropertyErrors

Example: User tries to bring a Volume online, but the associated Controller is offline.

Preconditions:

Assume the following partial response to a GET /redfish/v1/Systems/Sys-1/Storage/NVMeSSD-EG/Controllers/1 request:

```
{
  "@odata.type": "#StorageController.v1_9_1.StorageController",
  "Id": "1",
  "Links": {
    "Properties": {
      "AttachedVolumes": [
        "/redfish/v1/Storage/1/Volumes/1"
      ]
    }
  },
  "Status": {
    "State": "Quiesced"
  }
}
```

Basic Course of Events:

1. Attempt to enable compression on the chosen Volume.

Request: PATCH /redfish/v1/Storage/1/Volumes/1

Headers: No additional headers required.

Body: json { "Compressed": True }

HTTP Status Code Returned: 400

Headers: No additional headers required.

Body:

```
{
  "error": {
    "code": "Base.1.21.PropertyValueResourceConflict",
    "Message": "The property 'Compressed' with the requested value of 'True' could not be written because the value conflicts with the state or configuration of the resource at '/redfish/v1/Systems/Sys-1/Storage/NVMeSSD-EG/Controllers/1'."
  }
}
```

AdditionalContext: None.

5.23 Error Case: Property Value Type Error

Summary: When a client sends a request to a Swordfish implementation, but includes but the value provided for required property is of the wrong type, the `PropertyValueTypeError` message shall be returned.

Common Cause: A string value is provided when a numeric value is required.

Error Group: PropertyErrors

Example: User tries to create a Volume, but provides a string value for CapacityBytes.

Basic Course of Events:

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

Request:

POST /redfish/v1/Storage/1/StoragePools/PrimaryPool/AllocatedVolumes

- **Headers:** Content-type : application/json
- **Body:**

```
{
  "Name" : "MyVolume",
  "CapacityBytes" : "1TiB",
  "RAIDType" : "RAID1"
}
```

HTTP Status Code Returned: 400

Headers: No additional headers required.

Body:

```
{
  "error": {
    "code": "Base.1.21.PropertyValueTypeError",
    "Message": "The value '1TiB' for the property CapacityBytes is not a type that the property can accept.",
  }
}
```

AdditionalContext: : None.

5.24 *Error Case: Resource Already Exists*

Summary: When a client requests a create operation on a resource, but the implementation will not accept the request because the resource already exists and returns `ResourceAlreadyExists` error.

Common Cause: A CREATE the implementation cannot be accepted, because the resource already exists.

Example: User tries to add a volume with the same ID as an existing volume.

Basic Course of Events:

1. POST the volume to the Volumes collection.

Request: POST /redfish/v1/Systems/1/Storage/1/Volumes

Headers: No additional headers required.

Body:

```
{
  "ID": "1",
  "CapacityBytes": "9284327497"
}
```

HTTP Status Code Returned: 409

Headers: No additional headers required.

Body:

```
{
  "error": {
    "code": "Base.1.21.ResourceAlreadyExists",
    "Message": "The requested resource of type Volume with the property ID with the value 1 already exists."
  }
}
```

Additional Context: This message also technically covers a PATCH case (“change or”) in the Redfish Base Message Registry (v1.6.1). At this time, we do not have any specific examples to cover this case.

5.25 Error Case: Resource At URI In Unknown Format

Summary: When a client sends a request that includes the URI of a required resource to a Swordfish implementation, but that URI cannot be parsed by the requesting service, the `ResourceAtURIInUnknownFormat` message shall be returned.

Common Cause: A malformed or incorrect URI

Error Group: ResourceErrors

Example: User tries to update the firmware of a Resource, but provides an invalid URI.

Basic Course of Events:

1. Call the FirmwareUpdate service

Assumptions:

A prior query of the update service returned the following valid firmware URIs:

```
{
  "@odata.context": "/redfish/v1/$metadata#SoftwareInventoryCollection.SoftwareInventoryCollection",
  "@odata.id": "/redfish/v1/UpdateService/FirmwareInventory",
  "@odata.type": "#SoftwareInventoryCollection.SoftwareInventoryCollection",
  "Description": "Collection of Firmware Inventory",
  "Members": [
    {
      "@odata.id": "/redfish/v1/UpdateService/FirmwareInventory/Installed-0-20.06.05.11__SSD.1-1-1"
    },
    {
      "@odata.id": "/redfish/v1/UpdateService/FirmwareInventory/Installed-0-20.06.05.11__SSD.1-2-1"
    }
  ],
  "Members@odata.count": 18,
  "Name": "Firmware Inventory Collection"
}
```

Request:

```
POST /redfish/v1/Systems/Sys-1/Storage/SimplestNVMeSSD/Controllers/
NVMeIOController/Actions/UpdateService.SimpleUpdate
```

- **Headers:** Content-type : application/json
- **Body:** json { "ImageURI": "/redfish/v1/\$metadata#SoftwareInventoryCollection.SoftwareInventoryCollection", "@Redfish.OperationApplyTime": "OnTargetReset" } **Response:**

HTTP Status Code Returned: 400

Headers: No additional headers required.

Body:

```
{
  "error": {
    "code": "Base.1.21.0.ResourceAtURIInUnknownFormat",
    "Message": "The resource at the URI '/redfish/v1/$metadata#SoftwareInventoryCollection.SoftwareInventoryCollection' is in a format not recognized by the service."
  }
}
```

AdditionalContext: None.

5.26 Error Case: Resource At URI Unauthorized – WIP

JMS: Do we need a privilege registry example for this message? Is there more here than HTTP status of 403?

Summary: When a client sends a request to a Swordfish implementation, but fails to include any properties required to process the request when one or more properties are expected, the “EmptyJSON” message shall be returned.

Common Cause: Add common cause of error

Error Group: One of GeneralErrors ActionErrors JSONErrors PropertyErrors ResourceErrors

Example: User tries to create a Volume, but omits any properties.

Basic Course of Events:

1. Post the definition of the new volume to the Volumes resource collection with no Body.

Request: POST /redfish/v1/Storage/1/Volumes

Headers: No additional headers required.

Body: ``json {
}

...

HTTP Status Code Returned: 400

Headers: No additional headers required.

Body:

```
{  
  "error": {  
    "code": "Base.1.21.EmptyJSON",  
    "Message": "The request body submitted contained an empty JSON object and the  
               service is unable to process it."  
  }  
}
```

AdditionalContext: Questions: What is the relative relationship of this case vs PropertyMissing? What guidance should we provide for required odata properties vs required Redfish.Required vs profile required? Is this a meaningful distinction between these cases?

5.27 *Error Case: Resource Cannot Be Deleted*

Summary: When a client requests a delete operation on a resource that cannot be deleted, the implementation will return a ResourceCannotBeDeleted error.

Common Cause: The named resource cannot be deleted.

Example: User tries to remove a storage controller.

Basic Course of Events:

1. Delete the storage controller from the Storage object.

Request: DELETE /redfish/v1/Systems/1/Storage/1#/StorageController/0

Headers: No additional headers required.

Body: None.

HTTP Status Code Returned: 405

Headers: No additional headers required.

Body:

```
{
  "error": {
    "code": "Base.1.21.ResourceCannotBeDeleted",
    "Message": "The delete request failed because the resource requested cannot be
    deleted."
  }
}
```

Additional Context: This use case covers deletions that are prevented by schema notation (i.e., `deletable = false`). It also covers deletions that are allowed by the schema but are prohibited by the implementation.

5.28 Error Case: Resource Creation Conflict

Summary: When a client requests the creation of a new `Resource`, but completing the request would conflict with existing resources, the `ResourceCreationConflict` message shall be returned.

Common Cause: Recreation of an existing `Resource`

Error Group: `ResourceErrors`

Example: User tries to create a `Volume`, but a `Volume` of that name already exists in the chosen `StoragePool`.

Basic Course of Events:

Assumptions: A `Volume` named “MyVolume” has already been created from `StoragePool` “PrimaryPool”.

1. Post the definition of the new volume to the `AllocatedVolumes` resource collection.

This instructs the service to use the identified `StoragePool` to allocate a new volume of the requested size that meets the requirements of the specified protection level. Since additional details are not provided, the service will rely on default values as required.

Request:

POST /redfish/v1/Storage/1/StoragePools/PrimaryPool/AllocatedVolumes

- **Headers:** Content-type : application/json
- **Body:**

```
{
  "Name" : "MyVolume",
  "CapacityBytes" : 1099511627776,
  "RAIDType" : "RAID1"
}
```

HTTP Status Code Returned: 400

Headers: No additional headers required.

Body:

```
{
  "error": {
    "code": "Base.1.21.ResourceCreationConflict",
    "Message": "The resource could not be created. The service has a resource at
URI '/redfish/v1/Storage/1/StoragePools/PrimaryPool/AllocatedVolumes' that
conflicts with the creation request."
  }
}
```

AdditionalContext: None.

5.29 Error Case: Resource Exhaustion

Summary: When a client sends a request to a Swordfish implementation that cannot be satisfied due to a shortage of resources, the `ResourceExhaustion` message shall be returned.

Common Cause: An attempt to expand a `Volume` beyond the underlying storage capacity.

Error Group: ResourceErrors

Example: User tries to expand a `Volume` beyond the underlying storage capacity.

Basic Course of Events:

Assumptions:

The underlying `StoragePool` for the target `Volume` returns the following information:

```

{
...
"@odata.id": "/redfish/v1/StorageServices/1/StoragePools/BasePool",
"Capacity": {
  "Data": {
    "ConsumedBytes": 824633720832,
    "AllocatedBytes": 1099511627776
  },
  "Metadata": null,
  "Snapshot": null
},
...
}

```

1. Use expand action to increase the Volume capacity to 5 TB .

Note that there are two different properties that can be used to report available capacity, depending on the implementation - either CapacityBytes or Capacity->Data->AllocatedBytes. This use case shows how to do this function using the latter property, but would work equally if the implementation supported this function using the CapacityBytes property instead.

Request:

```
PATCH /redfish/v1/StorageServices/1/Volumes/61001234876545676100123487654567
```

- **Headers:** Content-type : application/json
- **Body:**

```

{
  "Capacity": {
    "Data": {
      "AllocatedBytes": 5000000000000
    }
  }
}

```

HTTP Status Code Returned: 400

Headers: No additional headers required.

Body:

```

{
  "error": {
    "code": "Base.1.21.0.ResourceExhaustion",
    "Message": "The resource '/redfish/v1/StorageServices/1/StoragePools/BasePool'
was unable to satisfy the request due to unavailability of resources."
  }
}

```

AdditionalContext: None.

5.30 *Error Case: Resource In Standby*

Summary: When a client attempts to employ a Resource that has been quiesced, the `ResourceInStandby` message shall be returned.

Common Cause: Attempting to use a Resource that has been quiesced, or is not available for some other reason.

Error Group: ResourceErrors

Example: User tries to add a Drive to an existing StoragePool, but `Drive.State` is set to `StandbyOffline`.

Basic Course of Events:

1. Post the Drive to the StoragePool resource collection with no Body.

Assumptions:

Assume a Drive (`/redfish/v1/Chassis/1/Drives/1`) that returns the following information:

```

```json
{
 "@odata.type": "#Drive.v1_15_0.Drive",
 "Id": "0TH9AWMU",
 "Name": "Drive 4",
 "Status": {
 "State": "StandbyOffline",
 "Health": "OK"
 },
 "StatusIndicator": "OK",
 "CapacityBytes": 300067890136,
 "FailurePredicted": false,
 "Protocol": "SAS",
 "MediaType": "HDD",

```

```

"Manufacturer": "HDD-Company",
"SerialNumber": "0TH9AWMU",
"PartNumber": "HUC156030CSS200",
"Identifiers": [
 {
 "DurableNameFormat": "NAA",
 "DurableName": "300062B202B21841"
 }
],
"RotationSpeedRPM": 15000,
"BlockSizeBytes": 512,
"NegotiatedSpeedGbs": 12,
"Links": {
 "StoragePools": [
 {
 "@odata.id": "/redfish/v1/Storage/MidrangeStorageSystem/StoragePools/StoragePool2"
 }
]
},
"@odata.id": "/redfish/v1/Chassis/StorageEnclosure1/Drives/0TH9AWMU",
"@Redfish.Copyright": "Copyright 2015-2026 SNIA. All rights reserved."

```

```
} ``
```

**Request:** POST /redfish/v1/Storage/1/StoragePools/PrimaryPool.AddDrives

- **Headers:** Content-type : application/json
- **Body:**

```

{
 "Drives" : [
 {"@odata.id": "/redfish/v1/Chassis/1/Drives/1"}
]
}

```

**HTTP Status Code Returned:** 400

**Headers:** No additional headers required.

**Body:**

```

{
 "error": {

```

```
 "code": "Base.1.21.ResourceInStandby",
 "Message": "The request could not be performed because the resource is in
standby."
 }
}
```

**AdditionalContext:** None.

## 5.31 *Error Case: Resource In Use*

**Summary:** When a client requests a change to a resource, but the change is rejected by the implementation due to the resource being in use or a transitional state, returns the `ResourceInUse` error.

**Common Cause:** The requested change cannot be completed because the resource is in use or in transition.

Example: User tries to delete a volume marked with a `VolumeUsage` of “InUse”.

**Basic Course of Events:**

1. DELETE the volume.

**Request:** DELETE /redfish/v1/Systems/1/Storage/1/Volumes/3

**Headers:** No additional headers required.

**Body:**

None.

**HTTP Status Code Returned:** 400

**Headers:** No additional headers required.

**Body:**

```
{
 "error": {
 "code": "Base.1.21.ResourceInUse",
 "Message": "The change to the requested resource failed because the resource is
in use or in transition."
 }
}
```

**Additional Context:** None.

## 5.32 Error Case: Resource Missing At URI

**Summary:** When a client sends a request that includes the URI of a required resource to a Swordfish implementation, but the URI doesn't refer to a valid resource, the `ResourceMissingAtURI` message shall be returned.

**Common Cause:** A malformed or incorrect URI

**Error Group:** ResourceErrors

Example: User tries to update the firmware of a Resource, but provides an invalid URI.

**Basic Course of Events:**

JMS: Do we need to preceding steps (e.g., query software inventory, identify target controller)?

1. Call the FirmwareUpdate service

**Assumptions:**

A prior query of the update service returned the following valid firmwre URIs:

```
{
 "@odata.context": "/redfish/
 v1/$metadata#SoftwareInventoryCollection.SoftwareInventoryCollection",
 "@odata.id": "/redfish/v1/UpdateService/FirmwareInventory",
 "@odata.type": "#SoftwareInventoryCollection.SoftwareInventoryCollection",
 "Description": "Collection of Firmware Inventory",
 "Members": [
 {
 "@odata.id": "/redfish/v1/UpdateService/FirmwareInventory/
 Installed-0-20.06.05.11__SSD.1-1-1"
 },
 {
 "@odata.id": "/redfish/v1/UpdateService/FirmwareInventory/
 Installed-0-20.06.05.11__SSD.1-2-1"
 }
],
 "Members@odata.count": 18,
 "Name": "Firmware Inventory Collection"
}
```

**Request:**

```
POST /redfish/v1/Systems/Sys-1/Storage/SimplestNVMeSSD/Controllers/
NVMeIOController/Actions/UpdateService.SimpleUpdate
```

- **Headers:** Content-type : application/json
- **Body:** json { "ImageURI": "https://localhost/redfish/v1/UpdateService/FirmwareInventory/Installed-0-20.06.05.11.SSD.1-2-1",  
"@Redfish.OperationApplyTime": "OnTargetReset" } > Note replacement of '\_\_\_' with '.' in ImageURI.

**Response:**

**HTTP Status Code Returned:** 400

**Headers:** No additional headers required.

**Body:**

```
{
 "error": {
 "code": "Base.1.21.0.ResourceMissingAtURI",
 "Message": "The resource at the URI 'https://localhost/redfish/v1/UpdateService/FirmwareInventory/Installed-0-20.06.05.11.SSD.1-2-1' was not found."
 }
}
```

**AdditionalContext:** None.

## 5.33 Error Case: Resource Not Found

**Summary:** When a client requests a delete operation on a resource that cannot be found, the implementation will accept the request but then return a `ResourceNotFound` error.

**Common Cause:** The DELETE request references a resource that cannot be found.

Example: User tries to remove a volume that has already been deleted by another client or some other task.

**Basic Course of Events:**

1. Delete the volume from the Volumes collection.

**Request:** DELETE /redfish/v1/Systems/1/Storage/1/Volumes/1

**Headers:** No additional headers required.

**Body:** None.

**HTTP Status Code Returned:** 404

**Headers:** No additional headers required.

**Body:**

```
{
 "error": {
 "code": "Base.1.21.ResourceNotFound",
 "Message": "The requested resource of type Volume named 1 was not found."
 }
}
```

**Additional Context:** While the Redfish specification allows implementations to return a status code of 200 for this case, Swordfish recommends a 404 return code, to clarify a successful deletion from the detection of a prior deletion.

## 5.34 Error Case: Resource Type Incompatible

**Summary:** When a client sends a request to a Swordfish implementation, but the `ResourceType` supplied in the request doesn't match that of the target `Resource`, the `ResourceTypeIncompatible` message shall be returned.

**Common Cause:** Attempting to add to a `ResourceCollection` with an instance of the wrong `ResourceType`

**Error Group:** ResourceErrors

Example: User tries to expand a `StoragePool` but the request doesn't contain a `Drive`.

**Basic Course of Events:**

1. Use the "AddDrives" Action on the `PrimaryPool` storage pool, passing the selected drives as input.

**Request:** POST /redfish/v1/Storage/1/StoragePools/PrimaryPool.AddDrives

- **Headers:** Content-type : application/json
- **Body:**

```
{
 "Drives" : [
 {"@odata.id": "/redfish/v1/Storage/MidrangeStorageSystem/StoragePools/
 StoragePool1/CapacitySources/Source1/ProvidingDrives"}
]
}
```

**HTTP Status Code Returned:** 400

**Headers:** No additional headers required.

**Body:**

```
{
 "error": {
 "code": "Base.1.21.EmptyJSON",
 "Message": "The @odata.type of the request body
 Collection(Capacity.CapacitySource) is incompatible with the @odata.type of the
 resource, which is Drive.Drive."
 }
}
```

**AdditionalContext:** None.

## 5.35 *Error Case: Unrecognized Request Body*

**Summary:** When a client sends a request to a Swordfish implementation, but includes a request body that cannot be recognized as JSON, the `UnrecognizedRequestBody` message shall be returned.

**Common Cause:** Invalid request body content, such as a binary file

**Error Group:** JSONErrors

Example: User tries to update the firmware of a Resource, but includes the firmware file directly in the body request, rather than a reference to a valid firmware URI.

**Basic Course of Events:**

1. Call the FirmwareUpdate service

**Request:**

```
POST /redfish/v1/Systems/Sys-1/Storage/SimplestNVMeSSD/Controllers/
```

NVMeIOController/Actions/UpdateService.SimpleUpdate

- **Headers:** Content-type : application/json
- **Body:** 2123 622f 6e69 732f 0a68 230a 4320 706f 7279 6769 7468 2820  
2943 3220 3030 2d31 3032 3230 202c 3032 3730 202c 3032 3031 322d  
3130 2038 7246 6565 5320 666f 7774 7261 2065 6f46 6e75 6164 6974  
6e6f 202c 6e49 2e63 230a 4320 706f 7279 6769 7468 2820 2943 3120  
3939 2c32 3120 3939 2033 654a 6e61 6c2d 756f 2070 6147 6c69 796c  
0a0a 2023 6854 7369 7020 6f72 7267 6d61 6920 2073 7266 6565 7320  
666f 7774 7261

**Response:**

**HTTP Status Code Returned:** 400

**Headers:** No additional headers required.

**Body:**

```
{
 "error": {
 "code": "Base.1.21.0.UnrecognizedRequestBody",
 "Message": "The service detected a malformed request body that it was unable to
interpret."
 }
}
```

**AdditionalContext:** None.

## 6 Event Notifications

This section will provide examples and best practices around event notifications