Deep Sea Fishing: A Swordfish Deep-Dive

Richelle Ahlvers
Principal Storage Management Architect Broadcom Limited

SNIA Scalable Storage Management (SSM) Technical Work Group Chair
Abstract

- Building on the concepts presented in the Introduction to Redfish and Swordfish sessions, Deep-Sea Fishing goes into more detail on the SNIA Swordfish™ specification concepts, including Class of Service.

- This deep-dive also provides a look at both the CSDL and json schema formats, including OData syntax to support a limited set of OData client capability.
Disclaimer

- The information in this presentation represents a snapshot of work in progress within SNIA
- This information is subject to change without notice.
- For additional information, see the SNIA website: [www.snia.org/swordfish](http://www.snia.org/swordfish)
A Recap: The Resource Map
Starting with Redfish: An Overview

Redfish Resource Map
Adding Storage to Redfish…

- **/redfish/v1/Root Resource**
  - Links to all content
  - Sessions
  - Accounts
  - Schemas
  - Events

- **/redfish/v1/Managers**
  - Collection of Managers
    - BMC functionality
    - 1..n

- **/redfish/v1/Managers/<id>**
  - BMC
    - System Manager operations
    - Power
    - Thermal
    - Logs
    - Services
    - 1..n

- **/redfish/v1/Systems**
  - Collection of Systems
    - "Logical" view of general purpose systems
    - 1..n

- **/redfish/v1/Systems/<id>**
  - Server Information
    - Model #, Serial #, Boot Order, NIC, MAC, status, etc.
    - Processor
    - Disks
    - NICs
    - Chassis
      - Chassis global physical asset info
      - Power
      - Thermal
      - Logs
      - Services
      - Sessions
      - Accounts
      - Schemas
      - Events

- **/redfish/v1/Chassis**
  - Collection of Chassis
    - "Physical" view of the system
    - 1..n

- **/redfish/v1/Chassis/<id>**
  - Chassis
    - Chassis global physical asset info
    - Processor
    - Disks
    - NICs
Adding Storage to Redfish…

- **/redfish/v1/Chassis**
  - Collection of Chassis
  - "Physical" view of the system

- **/redfish/v1/Managers**
  - Collection of Managers
  - BMC functionality

- **/redfish/v1/Systems**
  - Collection of Systems
  - "Logical" view of general purpose systems

- **/redfish/v1/StorageSystems**
  - Collection of StorageSystems
  - "Logical" view of dedicated systems

- **/redfish/v1/StorageServices**
  - Collection of StorageServices
  - Storage functionality: block, file, object

- **/redfish/v1/StorageSystems/<id>**
  - StorageSystem Information
  - Model #, Serial #, UIDs, status, etc.

- **/redfish/v1/Managers/<id>**
  - BMC
  - Chassis global physical asset info

- **/redfish/v1/Systems/<id>**
  - Server Information
  - Model #, Serial #, Boot Order, NIC MAC, status, etc.

- **/redfish/v1/Chassis/<id>**
  - Chassis
  - Power
  - Thermal

- **/redfish/v1/StorageSystems/<id>**
  - StorageService Information
  - Class of Service, Pools, Groups, Endpoints, Volumes/Files, Drives

- **/redfish/v1/Systems**
  - Collection of Systems
  - "Logical" view of general purpose systems

- **/redfish/v1/Managers**
  - Collection of Managers
  - BMC functionality

- **/redfish/v1/StorageServices**
  - Collection of StorageServices
  - Storage functionality: block, file, object

- **/redfish/v1/StorageSystems**
  - Collection of StorageSystems
  - "Logical" view of dedicated systems

- **/redfish/v1/StorageServices/<id>**
  - StorageService Information
  - Class of Service, Pools, Groups, Endpoints, Volumes/Files, Drives

- **/redfish/v1/Systems**
  - Collection of Systems
  - "Logical" view of general purpose systems

- **/redfish/v1/Managers**
  - Collection of Managers
  - BMC functionality
Swordfish Concepts
Storage Systems

- Contains a list of Computer System instances that are used for storage applications
- Each instance in the collection also belongs to the overall Computer System collection
  - The collection will be a subset of what’s in the Computer System collection

```
/_redfish/v1/Systems
Collection of Systems
“Logical” view of the system

/Redfish/v1/StorageSystems
Collection of Systems
Systems used for storage

.../Systems/<id>
System Information
System type, part number, serial number, model, SKU, asset tag, LED, power state, boot options, reset actions, etc
```
Storage Services

- Contains a list of Storage Service instances that manage the different storage related functionality
- Focus of StorageService is on the logical aspects of the storage (such as classes of service, volumes, file systems / file shares, storage pools)
Defining Classes of Service

- Classes of Service are composed of Lines of Service
  - Types: Data Protection, Data Security, Data Storage, Connectivity, Performance
    - Properties of each are defined in …LoSCapabilities
  - Can be constructed with one to all LinesOfService and properties (simple to complex)

```
../StorageServices/<id>/ClassesOfService
Collection of ClassOfService
Classes of Service used by the service

../StorageServices/<id>/LinesOfService
Collections of LineOfService
LoS Options available to construct Classes of Server

../LoSCapabilities
LineOfService Capability Definitions
Supported capabilities of the specific system
```
Simple Example:

- Use a single capability property to create two possible classes of service
  - IOConnectivityLoSCapabilities. SupportedAccessProtocols (e.g., drive type)
- Create LinesOfService that includes this property
- Create two classes of service: One for NVMe drives, one for SAS / SATA drives
Example: Two Classes Of Service

```json
"@odata.id": "/redfish/v1/StorageServices/1/ClassesOfService/HighCapacity",

"@odata.type": 
"#ClassOfService_1_0_0.ClassOfService",

"Name": "HighCapacity",

"Description": "HighCapacity Class Of Service (SATA)",

"Id": "HighCapacity",

"ClassOfServiceVersion": "1.01.00",

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

"LinesOfService": {
  "IOLoConnectivityLineOfService": {
    "Name": "Serial Attached ATA",
    "AccessProtocol": "SATA"
  }
}

"@odata.context": "/redfish/v1/$metadata#StorageServices/1/ClassesOfService/HighPerformance",

"@odata.id": "/redfish/v1/StorageServices/1/ClassesOfService/HighPerformance",

"@odata.type": 
"#ClassOfService_1_0_0.ClassOfService",

"Name": "HighPerformance",

"Description": "HighPerformance Class Of Service (NVMe SSDs)",

"ClassOfServiceVersion": "1.01.00",

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

"LinesOfService": {
  "IOLoConnectivityLineOfService": {
    "Name": "NVMe",
    "AccessProtocol": "NVMe"
  }
}
```
Using Classes Of Service

- Once ClassesOfService exist (either pre-existing from vendor or constructed from capabilities -> LinesOfService -> ClassesOfService):
  - Create StoragePools using specified Classes of Service
  - When allocating volumes, use the StoragePools’ Class of Service attributes to determine which to use to request capacity
Example: Create Volume

`POST /redfish/v1/StorageServices(1)/Volumes/`
{
   "Name": "Volume56",
   "CapacityBytes": 1099511627776,
   "Links": {
      "ClassOfService": {"odata.id":
          "/redfish/v1/StorageServices(1)/ClassesofService(HighPerformance)"
      }
   }
}

Schema Primer
Schema Files

- Each resource is described by a single schema file (in CSDL)
  - Each schema file may pull in external definitions
- Each response contains an "@odata.type" property to provide top level decoding information
  - "@odata.type" is broken down as "#Namespace.Entity"
  - Example: "#Volume.v1_2_0.Volume" means it’s a “Volume” entity in the “Volume.v1_2_0” namespace
- Generic clients may dynamically parse schema files as it’s interacting with a service in order to automatically build data models
- Purpose built clients may not necessarily read schema files directly
  - Developers may reference schema files when writing these clients to understand what resources and properties are available
- All standard schema files are (re-)published on the DMTF website
  - DMTF schema published directly
  - SNIA schema “republished”
  - OEM schema may also be (and are recommended by SSM to be) “republished” on the DMTF site
- Clients and services may have local copies
Schema File Types

- Currently two types of schema files exist: CSDL and JSON
  - CSDL (one file with all versions)
    - Common Schema Definition Language
    - XML formatted
    - Standardized by OASIS to support the OData standard
  - JSON (one file per version)
    - Follows http://json-schema.org/ format
    - Generated by scripts from the CSDL schemas
- Both sets of schemas are functionally equivalent
- Using one type over the other is a matter of preference on the part of the developer
CSDL Schema Example

```xml
<edmx:Edmx xmlns:edmx="http://docs.oasis-open.org/odata/ns/edmx" Version="4.0">
        <edmx:Include Namespace="Org.OData.Core.V1" Alias="OData"/>
    </edmx:Reference>
        <edmx:Include Namespace="Org.OData.Capabilities.V1" Alias="Capabilities"/>
    </edmx:Reference>
    <edmx:Reference Uri="http://redfish.dmtf.org/schemas/v1/Resource_v1.xml">
        <edmx:Include Namespace="Resource.v1_0_0"/>
    </edmx:Reference>
    <edmx:Reference Uri="http://redfish.dmtf.org/schemas/v1/RedfishExtensions_v1.xml">
        <edmx:Include Namespace="RedfishExtensions.v1_0_0" Alias="Redfish"/>
    </edmx:Reference>

    <edmx:DataServices>

        <Schema xmlns="http://docs.oasis-open.org/odata/ns/edm" Namespace="Session">
            <EntityType Name="Session" BaseType="Resource.v1_0_0.Resource" Abstract="true">
                <Annotation Term="OData.Description" String="..."/>
                <Annotation Term="OData.LongDescription" String="..."/>
            </EntityType>
        </Schema>
    </edmx:DataServices>
</edmx:Edmx>
```
CSDL Schema Example (cont.)

```
<Schema xmlns="http://docs.oasis-open.org/odata/ns/edm" Namespace="Session.v1_0_0">

  <EntityType Name="Session" BaseType="Session.Session">
    <Property Name="UserName" Type="Edm.String">
      <Annotation Term="OData.Permissions" EnumMember="OData.Permission/Read"/>
      <Annotation Term="Redfish.RequiredOnCreate"/>
      <Annotation Term="OData.Description" String="..."/>
      <Annotation Term="OData.LongDescription" String="..."/>
    </Property>
    <Property Name="Password" Type="Edm.String">
      <Annotation Term="OData.Permissions" EnumMember="OData.Permission/Read"/>
      <Annotation Term="Redfish.RequiredOnCreate"/>
      <Annotation Term="OData.Description" String="..."/>
      <Annotation Term="OData.LongDescription" String="..."/>
    </Property>
  </EntityType>

</Schema>
```
CSDL Schema Example (cont.)

```
<Schema xmlns="http://docs.oasis-open.org/odata/ns/edm" Namespace="Session.v1_1_0">
  <EntityType Name="Session" BaseType="Session.v1_0_0.Session">
    <Property Name="Actions" Type="Session.v1_1_0.Actions" Nullable="false">
      <Annotation Term="OData.Description" String="..."/>
      <Annotation Term="OData.LongDescription" String="..."/>
    </Property>
  </EntityType>

  <ComplexType Name="Actions">
    <Annotation Term="OData.AdditionalProperties" Bool="false"/>
    <Annotation Term="OData.Description" String="..."/>
    <Annotation Term="OData.LongDescription" String="..."/>
    <Property Name="Oem" Type="Session.v1_1_0.OemActions" Nullable="false"/>
  </ComplexType>

  <ComplexType Name="OemActions">
    <Annotation Term="OData.AdditionalProperties" Bool="true"/>
    <Annotation Term="OData.Description" String="..."/>
    <Annotation Term="OData.LongDescription" String="..."/>
  </ComplexType>

</Schema>
</edmx:DataServices>
</edmx:Edmx>
```
JSON Schema Example

```json
{
  "$schema": "http://redfish.dmtf.org/schemas/v1/redfish-schema.v1_2_0.json",
  "title": "#Session.v1_1_0.Session",
  "$ref": "#/definitions/Session",
  "definitions": {
    "Session": {
      "type": "object",
      "additionalProperties": false,
      "properties": {
        "@odata.context": { "$ref": "http://.../odata.4.0.0.json#/definitions/context" },
        "@odata.id": { "$ref": "http://.../odata.4.0.0.json#/definitions/id" },
        "@odata.type": { "$ref": "http://.../odata.4.0.0.json#/definitions/type" },
        "Oem": {
          "$ref": "http://.../Resource.json#/definitions/Oem",
          "description": "...",
          "longDescription": "..."
        },
        "Id": {
          "$ref": "http://.../Resource.json#/definitions/Id",
          "readonly": true
        }
      }
    }
  }
}
```
JSON Schema Example (cont.)

"Description": {
    "anyOf": [
        {
            "$ref": "http://.../Resource.json#/definitions/Description",
            "type": "null"
        },
        "readonly": true
    ],
    "readonly": true
},
"Name": {
    "$ref": "http://.../Resource.json#/definitions/Name",
    "readonly": true
},
"UserName": {
    "type": [ "string", "null" ],
    "readonly": true,
    "description": "...",
    "longDescription": "..."
},
"Password": {
    "type": [ "string", "null" ],
    "readonly": true,
    "description": "...",
    "longDescription": "..."
},
JSON Schema Example (cont.)

"Actions": {
    "type": "object",
    "additionalProperties": false,
    "properties": {
        "Oem": {
            "type": "object",
            "additionalProperties": true,
            "properties": {},
            "description": "...",
            "longDescription": "..."
        }
    },
    "description": "...",
    "longDescription": "..."
}

"required": [ "Id", "Name" ],
"requiredOnCreate": [ "UserName", "Password" ],
"description": "...",
"longDescription": "..."
How to Participate: Shaping the Standard

- Find pointers to the latest technical content:
  - [http://snia.org/swordfish](http://snia.org/swordfish)
  - [http://www.snia.org/publicreview#swordfish](http://www.snia.org/publicreview#swordfish)
- Join the SSM TWG
  - By Joining the SNIA and SSM TWG, you can shape the standard: [https://members.snia.org/apps/org/workgroup/ssmtwg](https://members.snia.org/apps/org/workgroup/ssmtwg)
- Through the SNIA feedback portal, providing feedback on “Work In Progress”
  - As the group produces “Works In Progress”, you can provide feedback at [http://www.snia.org/feedback](http://www.snia.org/feedback)