

SDC 18

September 24-27, 2018
Santa Clara, CA

www.storagedeveloper.org

Fishing with Open Source Tools for SNIA Swordfish



Swordfish™

Scalable Storage Management

Don Deel
NetApp, Inc.

Agenda

- ❑ SNIA Swordfish™ Open Source Projects
 - ❑ Swordfish API Emulator
 - ❑ Swordfish Basic Web Client
 - ❑ Swordfish Sample Dashboard Datadog Integration
 - ❑ Swordfish Sample Dashboard Power BI Integration
- ❑ More About the Swordfish API Emulator
- ❑ Swordfish and Redfish Information

SNIA Swordfish™ Open Source Projects

- ❑ Swordfish was developed by the SNIA Scalable Storage Management Technical Work Group (SSM TWG)
 - ❑ Public information is available at www.snia.org/swordfish
- ❑ The SSM TWG maintains four open source projects
 - ❑ Swordfish API Emulator
 - ❑ Swordfish Basic Web Client
 - ❑ Swordfish Datadog Sample Dashboard Integration
 - ❑ Swordfish Power BI Sample Dashboard Integration
- ❑ Initial work funded by the Storage Management Initiative

Swordfish API Emulator

- ❑ Emulates a Swordfish system with storage services
- ❑ Responds to create, read, update, and delete operations
 - ❑ POST, GET, PATCH, DELETE
- ❑ Extends the DMTF [Redfish Interface Emulator](#)
 - ❑ Adds code for Swordfish resources
- ❑ Link: <https://github.com/SNIA/Swordfish-API-Emulator>
- ❑ Includes installation, user, and developer documentation

Swordfish API Emulator Console (Default Config)

```
(SAE180828) C:\Users\ddon\Documents\SAE180828>python emulator.py
['Redfish']
* Redfish endpoint at localhost:5000
{'rb': '/redfish/v1/', 'sys_id': 'System-1'}
{'rb': '/redfish/v1/', 'sys_id': 'System-2'}
{'rb': '/redfish/v1/', 'sys_id': 'System-3'}
{'rb': '/redfish/v1/', 'sys_id': 'System-4'}
{'rb': '/redfish/v1/', 'sys_id': 'System-5'}
{'rb': '/redfish/v1/', 'sys_id': 'System-6'}
{'rb': '/redfish/v1/', 'sys_id': 'System-7'}
* Running in Redfish mode
* Serving Flask app "g" (lazy loading)
* Environment: production
  WARNING: Do not use the development server in a production environment.
  Use a production WSGI server instead.
* Debug mode: off
```

Swordfish API Emulator Browser Output (Default)

```
localhost:5000/redfish/v1/

{
  @odata.context: "/redfish/v1/$metadata#ServiceRoot",
  @odata.type: "#ServiceRoot.1.0.0.ServiceRoot",
  @odata.id: "/redfish/v1/",
  Id: "RootService",
  Name: "Root Service",
  ServiceVersion: "1.0.0",
  UUID: "79ea8662-0349-4390-883c-b917c8f65e6b",
  - Links: {
    - Chassis: {
      @odata.id: "/redfish/v1/Chassis"
    },
    - Managers: {
      @odata.id: "/redfish/v1/Managers"
    },
    - TaskService: {
      @odata.id: "/redfish/v1/TaskService"
    },
    - SessionService: {
      @odata.id: "/redfish/v1/SessionService"
    },
  },
}
```


```
- StorageServices: {
  @odata.id: "/redfish/v1/StorageServices"
},
- StorageSystems: {
  @odata.id: "/redfish/v1/StorageSystems"
},
- AccountService: {
  @odata.id: "/redfish/v1/AccountService"
},
- EventService: {
  @odata.id: "/redfish/v1/EventService"
},
- Registries: {
  @odata.id: "/redfish/v1/Registries"
},
- Systems: {
  @odata.id: "/redfish/v1/Systems"
},
- CompositionService: {
  @odata.id: "/redfish/v1/CompositionService"
}
}
```

Swordfish Basic Web Client

- ❑ Web client that can connect to multiple Redfish and/or Swordfish services simultaneously
- ❑ Presents the entire Redfish and Swordfish hierarchy in a browser web frame
- ❑ Provides basic capabilities for viewing resources and updating properties that are writeable
- ❑ Link: <https://github.com/SNIA/Swordfish-basic-web-client>
- ❑ Includes installation, user, and developer documentation

Swordfish Basic Web Client Screen (Service Login)

← → ↻ ⓘ Not secure | 192.168.1.146:3000/#/home



Swordfish Service + Add - Remove

Add Swordfish Service ✕

No Services are available

IP Address:port


Domain Name

User Name

Password

Swordfish Basic Web Client Screen (Service Root)

← → ↻ ⓘ Not secure | 192.168.1.146:3000/#/home

 SAE180828

Swordfish Service **+** Add **-** Remove

SAE180828 >

Explore The Resources ✕

- Chassis >
- Managers >
- TaskService >
- SessionService >
- StorageServices >
- StorageSystems >
- AccountService >
- EventService >
- Registries >
- Systems >
- CompositionService >

Swordfish Basic Web Client Screen (StorageServices)

← → ↻ ⓘ Not secure | 192.168.1.146:3000/#/home

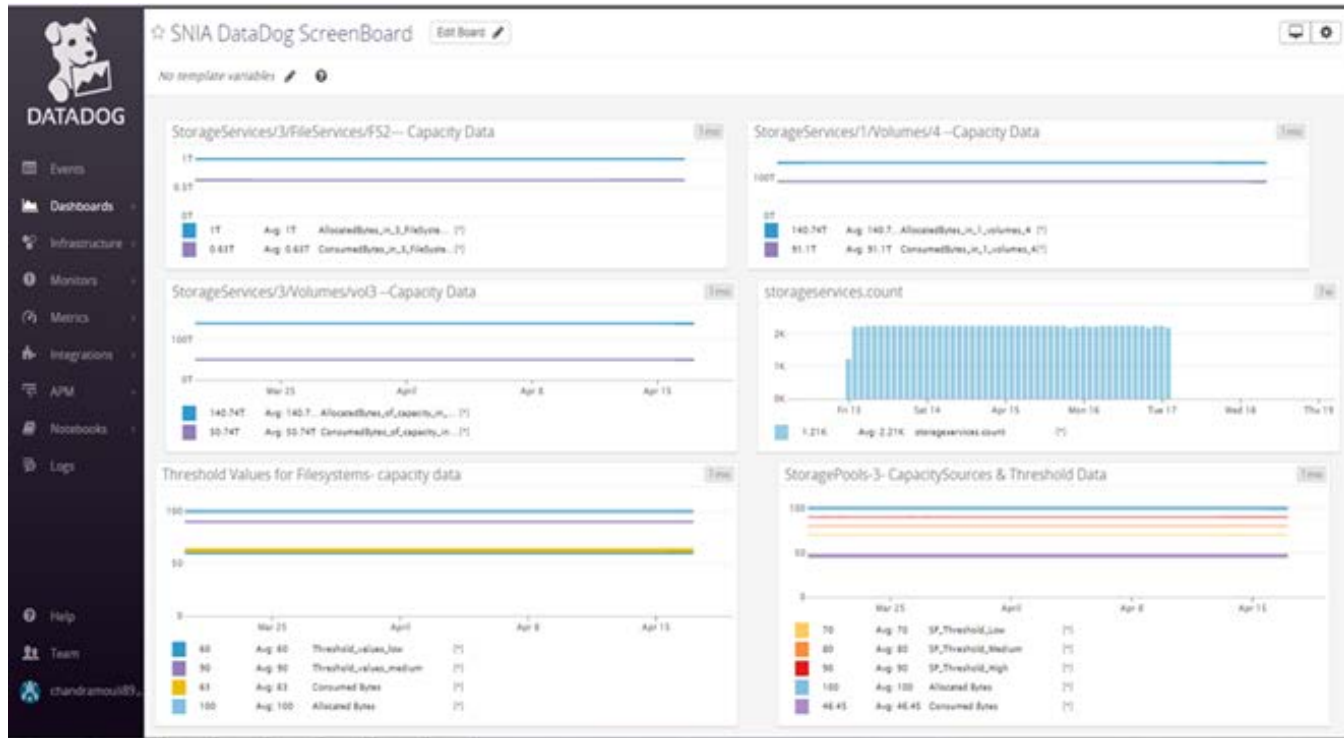
SAE180828 > StorageServices

Swordfish Service + Add - Remove	Explore The Resources ×	StorageServices + Add - Remove ×
SAE180828 ▶	Chassis ▶	1 ▶
	Managers ▶	2 ▶
	TaskService ▶	AFF-1 ▶
	SessionService ▶	
	StorageServices ▶	▼ Properties ✎ ↻
	StorageSystems ▶	Name : Storage Service Collection
	AccountService ▶	▶ ODATA
	EventService ▶	▶ LINKS
	Registries ▶	
	Systems ▶	
	CompositionService ▶	

Swordfish Datadog Sample Dashboard Integration

- ❑ Basic dashboard for the Datadog monitoring service
- ❑ Connects to a Swordfish service and provides an integration to the Datadog User Interface
- ❑ Displays storage system capacity information and the available storage capacity thresholds
- ❑ Can be a starting point for a customized Datadog plugin
- ❑ Link: <https://github.com/SNIA/Swordfish-datadog-sample-dashboard-integration>
- ❑ Includes installation, user, and developer documentation

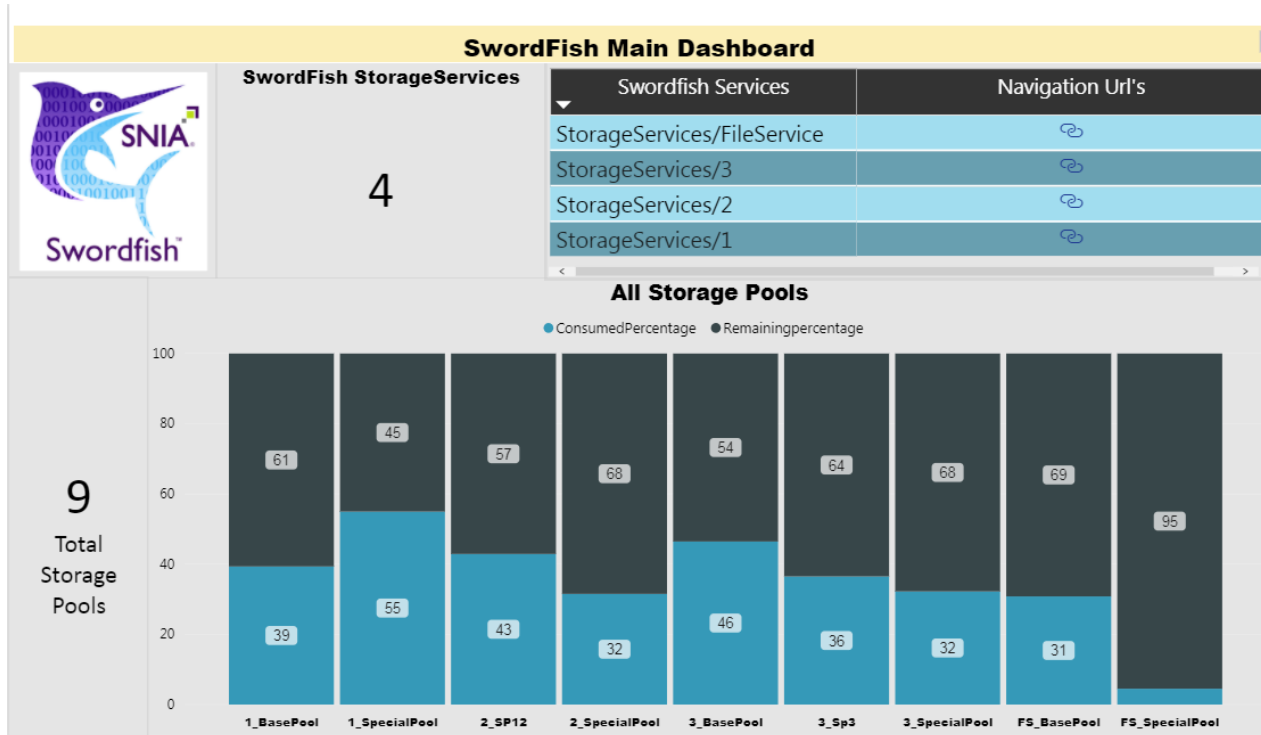
Swordfish Datadog Sample Dashboard Output



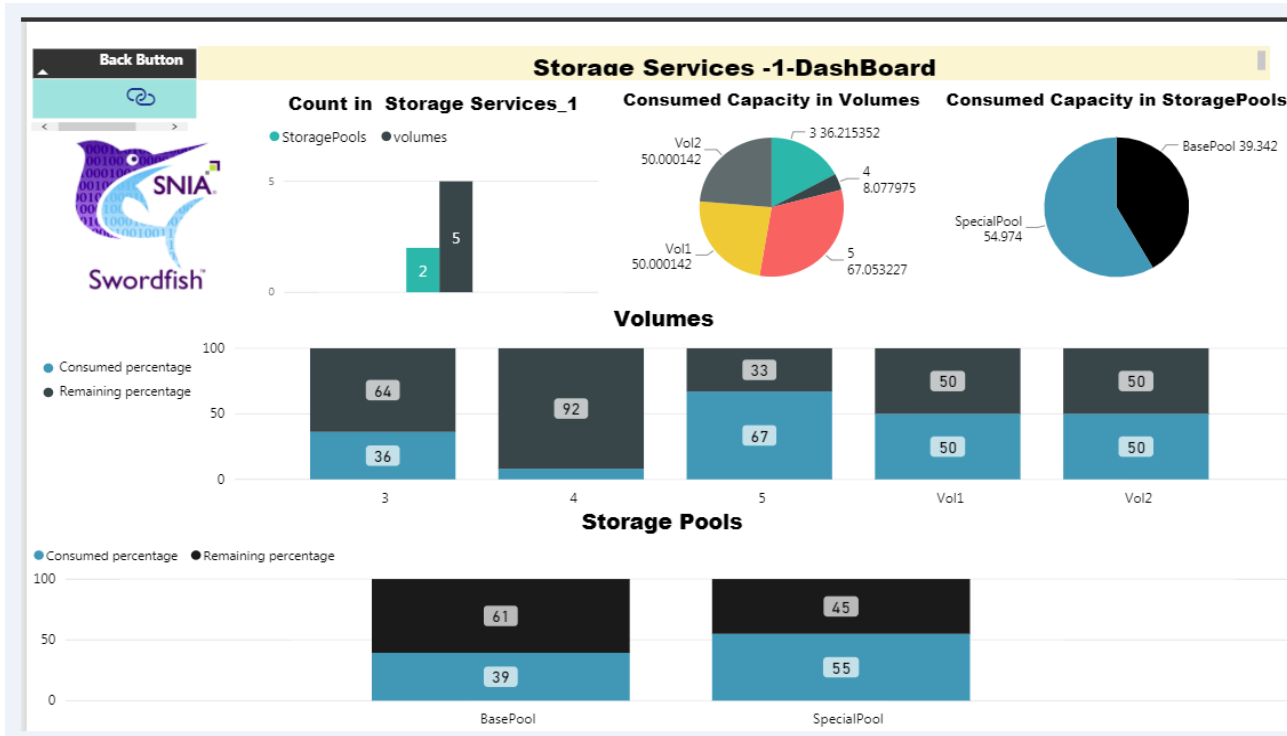
Swordfish Power BI Sample Dashboard Integration

- ❑ Basic dashboard for the Power BI monitoring system
- ❑ Connects to a Swordfish service and provides an integration to the Power BI User Interface
- ❑ Displays storage system capacity information and the available storage capacity thresholds
- ❑ Can be a starting point for a customized Power BI plugin
- ❑ Link: <https://github.com/SNIA/Swordfish-powerBI-sample-dashboard-integration>
- ❑ Includes installation, user, and developer documentation

Swordfish Power BI Sample Dashboard (Main)



Swordfish Power BI Sample Dashboard (child)



More About the Swordfish API Emulator

- ❑ Emulator Python Environment
- ❑ Installing the Emulator
- ❑ Notes About the Emulator
- ❑ How the Emulator Works
- ❑ Adding New Dynamic Resources
- ❑ More Notes About the Emulator

Emulator Python Environment

- ❑ Python 3.6 or above
- ❑ virtualenv recommended but not required
- ❑ Python packages
 - ❑ flask flask_restful flask_httpauth
 - ❑ requests aniso8601 markupsafe pytz
 - ❑ itsdangerous StringGenerator urllib3

Installing the Emulator (Default Configuration)

- ❑ Create a folder/directory for the Emulator
- ❑ Copy in the Redfish Interface Emulator
- ❑ Copy in the Swordfish API Emulator on top of it
- ❑ Install the necessary Python packages
- ❑ Run with “python emulator.py”

Emulator Console Output (Default Configuration)

```
(SAE180828) C:\Users\ddon\Documents\SAE180828>python emulator.py
['Redfish']
* Redfish endpoint at localhost:5000
{'rb': '/redfish/v1/', 'sys_id': 'System-1'}
{'rb': '/redfish/v1/', 'sys_id': 'System-2'}
{'rb': '/redfish/v1/', 'sys_id': 'System-3'}
{'rb': '/redfish/v1/', 'sys_id': 'System-4'}
{'rb': '/redfish/v1/', 'sys_id': 'System-5'}
{'rb': '/redfish/v1/', 'sys_id': 'System-6'}
{'rb': '/redfish/v1/', 'sys_id': 'System-7'}
* Running in Redfish mode
* Serving Flask app "g" (lazy loading)
* Environment: production
WARNING: Do not use the development server in a production environment.
Use a production WSGI server instead.
* Debug mode: off
```

Notes About the Emulator

- ❑ Read the [Redfish Interface Emulator *README.md*](#)
 - ❑ Says how to set up emulator.py flags and emulator-config.json
- ❑ `api_emulator\resource_manager.py` establishes which resources are static and which are dynamic
 - ❑ Static resources are read-only
 - ❑ Dynamic resources support CRUD operations
- ❑ Swordfish resources are all dynamic, but four of the Redfish resources are currently still static
 - ❑ AccountService, Registries, SessionService, TaskService

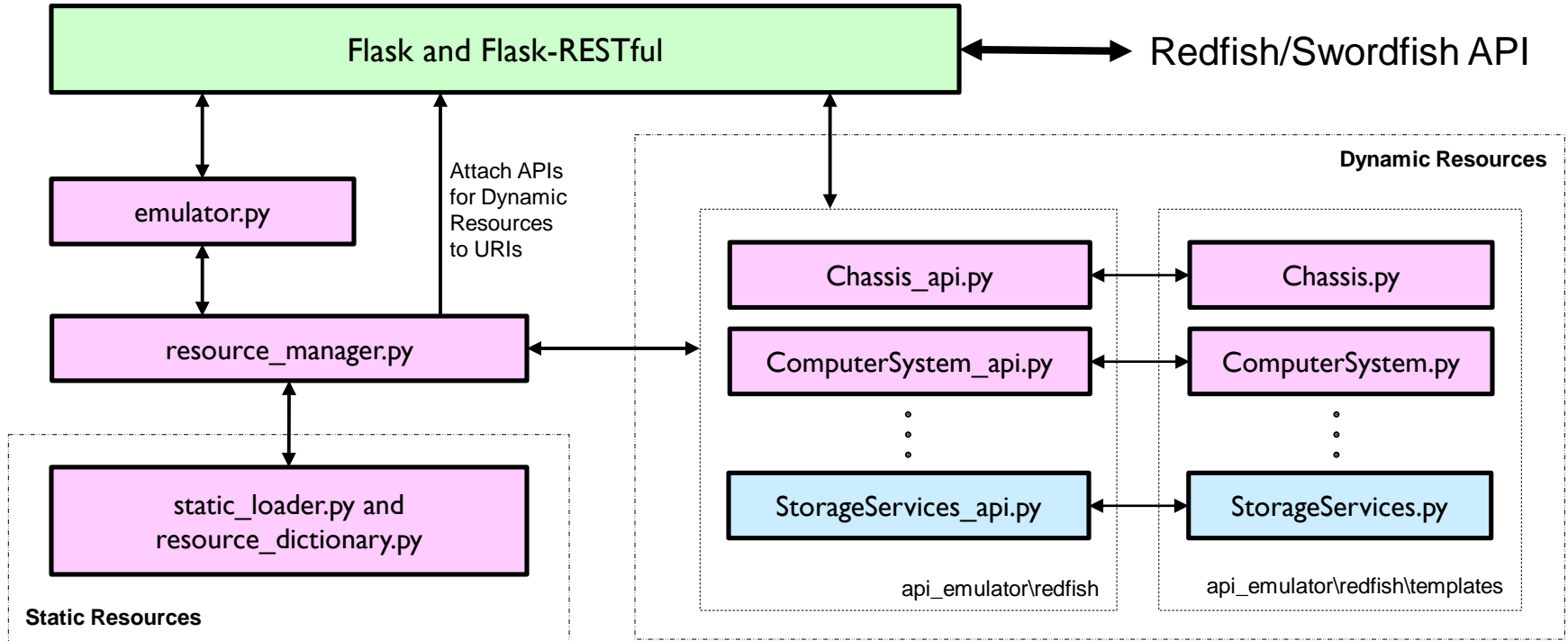
Notes About the Emulator (Continued)

- ❑ Static resources are populated by JSON mockup files in the `api_emulator\redfish\static` directory
 - ❑ Only uses static resources identified in `resource_manager.py`
 - ❑ Dynamic resources are NOT populated or initialized this way
- ❑ Dynamic resources can be populated via the emulator API using CRUD operations
 - ❑ The Redfish Interface Emulator includes a tool called Infragen that can also be used to prepopulate dynamic resources
 - ❑ An external tool for prepopulating from mockups is not there yet

Notes About the Emulator (Continued Again)

- ❑ An emulator-only function can populate dynamic objects
 - ❑ When defined by an api file for a dynamic resource, a POST with an empty body can create a new default singleton instance:
POST <http://localhost:5000/redfish/v1/Chassis/NewThing> {}
 - ❑ The new instance (named “NewThing” here) is defined by a template file for the dynamic resource (“Chassis” in this case)
- ❑ The Swordfish Basic Web Client uses this emulator-only function to create new Redfish and Swordfish singletons
 - ❑ It can then use PATCH operations to alter properties

How the Emulator Works



Adding New Dynamic Resources

- ❑ Dynamic resources are enabled by api/template file pairs
 - ❑ The api file sets REST behaviors for Collections and Singletons
 - ❑ The template file establishes how to create default singletons
- ❑ Example api/template files are in api_emulator\redfish
 - ❑ eg_resource_api.py and template\eg_resource.py
 - ❑ eg_subresource_api.py and template\eg_subresource.py
- ❑ The api example files show where to handle applicable REST commands for Collections and for Singletons
 - ❑ GET, PUT, POST, PATCH, DELETE

Adding New Dynamic Resources (Continued)

- ❑ The template example files show how templates are set up to allow new singleton instances to be created
 - ❑ A template is copied, with some things filled in at runtime
- ❑ When a new api/template pair is created, it is added to the emulator by editing `resource_manager.py`
 - ❑ This will attach the new resource's APIs to URIs

More Notes About the Emulator

- ❑ Areas where the emulator should improve
 - ❑ The remaining static resources should be made dynamic
 - ❑ A couple of Redfish-defined URIs are not supported yet
 - ❑ Need a tool for populating dynamic resources from mockups
- ❑ It matters when URIs do/don't have trailing slashes
 - ❑ The emulator does what the Redfish spec says, but...
- ❑ Feedback is desired
 - ❑ About possibilities as well as issues

Swordfish and Redfish Information

- ❑ Swordfish information on the public SNIA web site
 - ❑ Specs, Schema, Mockups, User's Guide, etc
 - ❑ Link: www.snia.org/swordfish
- ❑ Redfish information on the public DMTF web site
 - ❑ Specs, Schema, Mockups, Educational Material, etc.
 - ❑ Links: www.dmtf.org/redfish and <https://redfish.dmtf.org>
- ❑ Redfish (and Swordfish) Public Discussion Forum
 - ❑ Link: www.redfishforum.com (or www.swordfishforum.com)
 - ❑ Ask questions on this forum and get answers from experts



Contributing to SNIA open source projects

- ❑ SNIA open source projects welcome input!
- ❑ Contributors who are not SNIA members must agree to the terms of the SNIA Contributor License Agreement
 - ❑ Link: <https://www.snia.org/CLA>
- ❑ The SNIA Swordfish open source projects are covered by the terms of the BSD 3-clause License

Thank You!