

STORAGE DEVELOPER CONFERENCE



BY Developers FOR Developers

Virtual Conference
September 28-29, 2021

Swordfish for NVMe Requirements and Implementation

Highlights and tips for NVMe Swordfish implementations

Curtis Ballard, HPE Storage Distinguished Technologist

Hewlett Packard Enterprise

SNIA SSM TWG, Swordfish for NVMe Taskforce



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



Abstract



- Developed by the Storage Networking Industry Association (SNIA)
- SNIA Swordfish™ is an extension of the DMTF Redfish specification
- A unified approach for the management of storage equipment and services

This presentation will provide an overview of the most recent work adding detailed implementation requirements for specific configurations; ensuring NVMe and NVMe-oF environments can be represented entirely in Swordfish and Redfish environments.



Swordfish in 2021

Filling out the NVMe Story

In 2021 - NVMe and more NVMe

- SDC 2020 introduced NVMe mappings for Swordfish
- Most common NVMe and NVMe-oF models now fully mapped and documented
- New profiles for:
 - NVMe Drives with standard features
(SwordfishNVMeDrive.json)
 - NVMe Drives with advanced features
(SwordfishNVMeDriveAdvancedFeatures.json)
 - Native Ethernet Attached NVMe Drives (SwordfishNVMeEthernetAttach.json)
 - Arrays with NVMe-oF Host Interfaces
(SwordfishNVMeFrontEnd.json)
 - Opaque Array example in the mapping guide uses an NVMe-oF Host Interface and this profile

Future profiles

- NVMe-oF JBOF
- Native Ethernet NVMe-oF JBOF
- NVMe-oF logical subsystems composed from underlying resources (aligns with TP6011 in development at [NVM Express](#))
- All could be implemented today with Swordfish specification and mapping guidance in the SNIA Swordfish NVMe Model Overview and Mapping Guide

What else is new?

For more details on new Swordfish features watch the SDC presentation from Richelle Ahlvers

[What's New in SNIA Swordfish in 2021](https://storagedeveloper.org/events/sdc-2021/abstracts#resource-Ahlvers2)

(<https://storagedeveloper.org/events/sdc-2021/abstracts#resource-Ahlvers2>)



Swordfish Requirements

What are Swordfish Profiles?

Profiles – Your place for requirements

- What are all these files in my Swordfish bundle?

Name	Status	Date modified	Type	Size
 Swordfish_v1.2.3_Profiles.zip	✓	8/29/2021 8:32 PM	Compressed (zipp...	22 KB
 SwordfishBlockCapacityManagement.json	✓	8/29/2021 8:32 PM	JSON File	2 KB
 SwordfishBlockCoSLocalReplication.json	✓	8/29/2021 8:32 PM	JSON File	3 KB
 SwordfishBlockCoSRemoteReplication.js...	✓	8/29/2021 8:32 PM	JSON File	3 KB
 SwordfishBlockLocalReplication.json	✓	8/29/2021 8:32 PM	JSON File	3 KB
 SwordfishBlockMappingMasking.json	✓	8/29/2021 8:32 PM	JSON File	3 KB
 SwordfishBlockProvisioning.json	✓	8/29/2021 8:32 PM	JSON File	2 KB
 SwordfishBlockRemoteReplication.json	✓	8/29/2021 8:32 PM	JSON File	3 KB
 SwordfishCoSDiscovery.json	✓	8/29/2021 8:32 PM	JSON File	6 KB
 SwordfishDiscovery.json	✓	8/29/2021 8:32 PM	JSON File	3 KB
 SwordfishEnergyStar.json	✓	8/29/2021 8:32 PM	JSON File	3 KB
 SwordfishEventNotification.json	✓	8/29/2021 8:32 PM	JSON File	9 KB
 SwordfishFileCapacityManagement.json	✓	8/29/2021 8:32 PM	JSON File	2 KB
 SwordfishFileProvisioning.json	✓	8/29/2021 8:32 PM	JSON File	3 KB
 SwordfishIOPerformance.json	✓	8/29/2021 8:32 PM	JSON File	3 KB
 SwordfishNVMeDrive.json	✓	8/29/2021 8:32 PM	JSON File	19 KB
 SwordfishNVMeDriveAdvancedFeatures.j...	✓	8/29/2021 8:32 PM	JSON File	14 KB
 SwordfishNVMeEthernetAttach.json	✓	8/29/2021 8:32 PM	JSON File	8 KB
 SwordfishNVMeFrontEnd.json	✓	8/29/2021 8:32 PM	JSON File	15 KB

What are profiles?

- A form of template for a category of device
- Specifies Redfish/Swordfish implementation requirements
 - Underlying architecture is open as long as requirements are met
- Provides a common baseline for clients and targets
 - Many optional additional features could be implemented
- Profiles are published in human readable JSON
- Profile specification in DMTF specification DSP0272
 - [All Published Versions of DSP0272 | DMTF](#)

Profile Quickstart

- Header describes file and where it came from

```
"SchemaDefinition": "./RedfishInteroperabilityProfile.v1_3_1.json",  
"ProfileName": "SwordfishNUMeFrontEnd",  
"ProfileVersion": "1.0.0",  
"OwningEntity": "SNIA.org",  
"Purpose": "Defines requirements for Swordfish implementations with NUMe front-end interfaces, such as storage arrays.",  
"ContactInfo": "SNIA.org",
```

- May include other required profiles

```
"ConditionalRequirements": [{  
  "Purpose": "Either SwordfishDiscovery or SwordfishCoSDiscovery must be supported.",  
  "CompareProperty": "RequiredProfiles",  
  "CompareType": "AnyOf",  
  "CompareValues": [  
    [{  
      "SwordfishDiscovery": {  
        "OwningEntity": "SNIA.org",  
        "Repository": "https://redfish.dmtf.org/profiles/swordfish",  
        "MinVersion": "1.1.1"  
      }  
    }  
  ]  
},  
  [{  
    "CoSDiscovery": {  
      "OwningEntity": "SNIA.org",  
      "Repository": "https://redfish.dmtf.org/profiles/swordfish",  
      "MinVersion": "1.1.1"  
    }  
  }  
}]
```

Key concepts

- Empty properties by default are **mandatory**

```
"Identifiers": {  
  "PropertyRequirements": {  
    "DurableName": {},  
    "DurableNameFormat": {}  
  }  
},
```

- Requirements may have **property requirements**

```
"NUMeVersion": {  
  "Purpose": "NUMeVersion is required when the underlying NUMe implementation is revision 1.2 or higher.",  
  "ReadRequirement": "IfImplemented"  
},  
"MaxQueueSize": {}.
```

- **Properties may be included** to restrict or advise against implementation

```
"AssetTag": {  
  "ReadRequirement": "None",  
  "Purpose": "This property is recommended as 'Do not implement' for this device and protocol type.  
This information is available in the Drive resource if needed."  
},  
"CacheSupport": {
```

Profile key concepts

- Requirements can be **If/Then conditional**
 - “Comparison” specifies the test

```
"PropertyRequirements": {  
  "AttachedVolumes": {  
    "ReadRequirement": "None",  
    "ConditionalRequirements": [{  
      "Purpose": "AttachedVolumes must be implemented when the ControllerType is IO.  
Do not implement when ControllerType is Discovery or Admin.",  
      "CompareProperty": "ControllerType",  
      "CompareType": "Equal",  
      "CompareValues": ["IO"],  
      "Comparison": "Present",  
      "ReadRequirement": "Mandatory",  
      "WriteRequirement": "Mandatory"  
    }]  
  },  
}
```

- What about **optional properties**?
 - All properties not listed have no requirements and are optional

More information on profiles

- Helpful DMTF presentation on reading profiles
 - [Redfish Interoperability Profiles \(dmtf.org\)](https://www.dmtf.org/sites/default/files/Redfish_Interoperability_Profiles_v1.0.pdf)
(https://www.dmtf.org/sites/default/files/Redfish_Interoperability_Profiles_v1.0.pdf)
- The DMTF Interoperability Profiles specifications
 - [All Published Versions of DSP0272 | DMTF](https://www.dmtf.org/dsp/DSP0272)
(<https://www.dmtf.org/dsp/DSP0272>)
- A DMTF Presentation on understanding profiles at the OCP Summit
 - [Interpreting Redfish Profiles presentation at OCP](https://www.youtube.com/watch?v=mQpLJSkziUE&t=48s)
(<https://www.youtube.com/watch?v=mQpLJSkziUE&t=48s>)



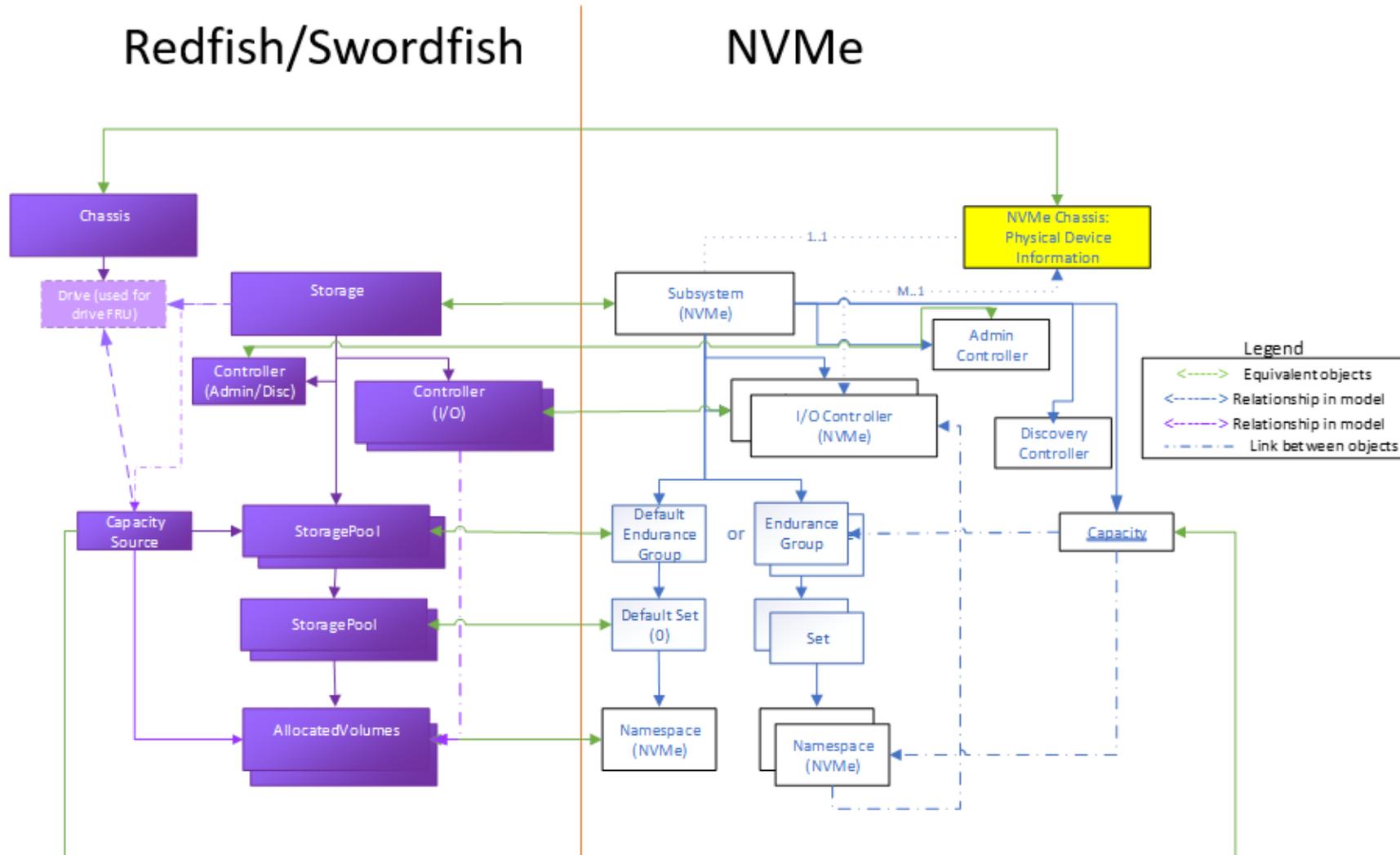
Swordfish NVMe Mapping Guide

Introduction to the Swordfish NVMe Model Overview and Mapping Guide

What is the NVMe mapping guide?

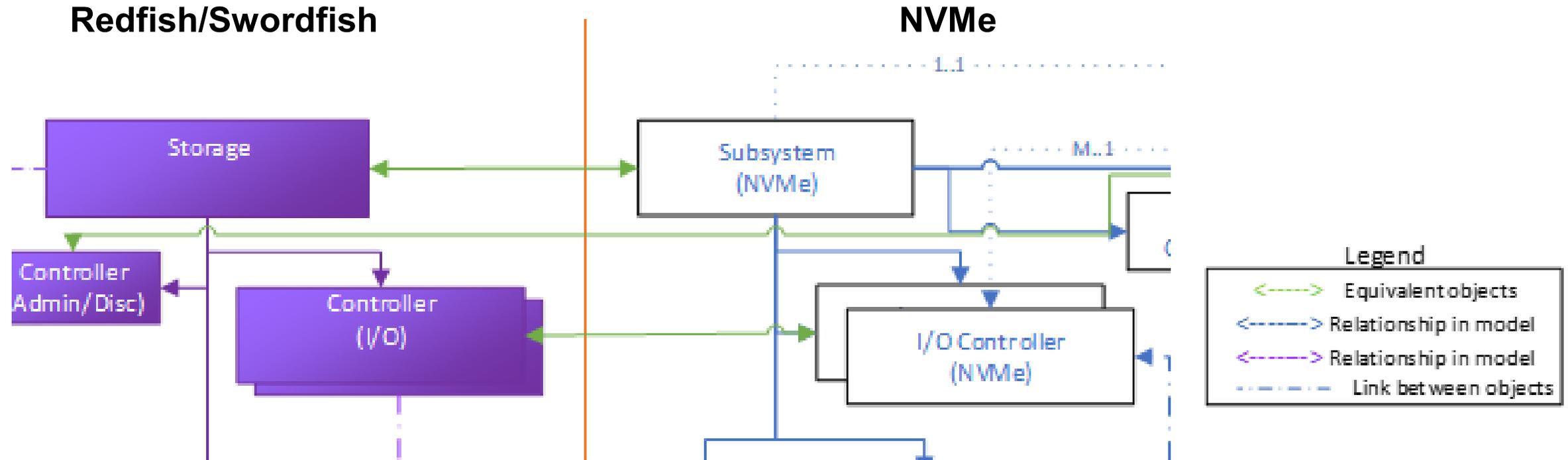
- A guide intended to help explain the mapping between Redfish/Swordfish objects and NVMe concepts
- Basic NVMe mapping model introduced first
- Examples of mapping for several common NVMe subsystem types
- Pages and pages and pages and pages of Redfish/Swordfish to NVMe concept mappings
 - Many, many hours of work represented in the mapping details
 - Intended to make it easy for the NVMe developer or Redfish/Swordfish developer to go from their “thing” to the other domain’s “thing”

Understanding the basic model



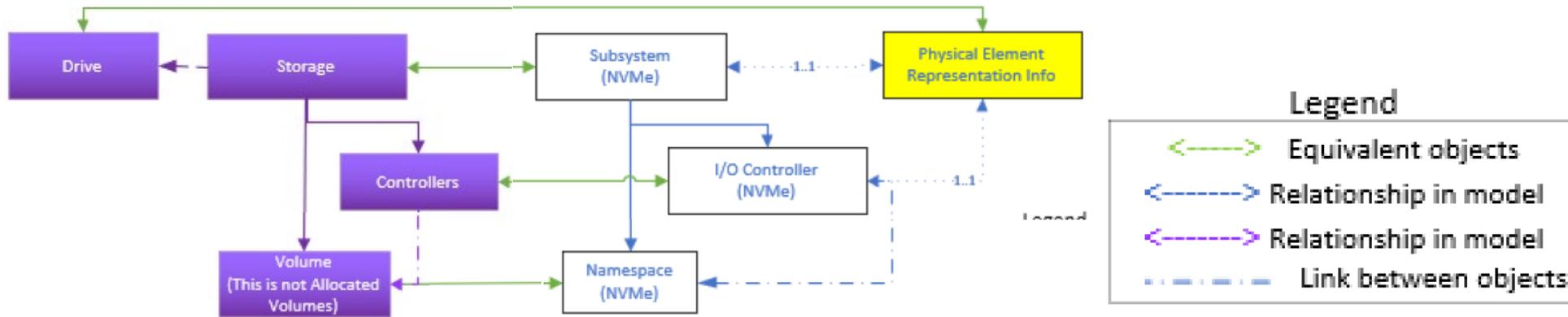
A bit overwhelming at first...
but really helpful once you
are familiar with it

A readable view



- **NVMe developers:** find the NVMe object **on the right**
- **Redfish/Swordfish developers:** find your favorite object **on the left**

Major model objects shown in figures



Example from **Simple SSD model**:

- **NVMe Namespace** == RF/SF Volume is part of an **NVMe subsystem** and is linked (attached) to an **I/O controller**
- **An NVMe I/O Controller** == RF/SF Controller is part of an **NVMe subsystem** and is linked (attached) to a **namespace** and is a **physical element**

Digging deeper

- Lower level properties can't be shown in a graphic
well at least one that you could read without a really big screen
- Mapping guide is written from Redfish/Swordfish perspective
- Find the Redfish/Swordfish object or property that you want to describe
 - Some are new NVMe specific properties!
- Look it up in the mapping guide
- Most objects and properties have a mapping to NVMe with pointers to the specification location
 - Current mapping refers to NVM Express 1.4
- List of tables on page 5 is a good way to get an overview

Example – space used in a volume

- From the SwordfishNVMeFrontEnd.json profile
 - The AllocatedBytes and ConsumedBytes values are required for “Volume” (NVMe namespace)

```
"Volume": {
  "PropertyRequirements": {
    "BlockSizeBytes": {},
    "Capacity": {
      "PropertyRequirements": {
        "Data": {
          "PropertyRequirements": {
            "AllocatedBytes": {},
            "ConsumedBytes": {}
          }
        }
      }
    }
  },
  "DisplayName": {},
  "Description": {
    "Comparison": "Equal",
    "Values": ["A Namespace is a quantity of non-volatile memory that may be formatted in
to logical blocks. When formatted, a namespace of size n is a collection of logical blocks with
logical block addresses from 0 to (n-1). NUMe systems can support multiple namespaces."],
    "ReadRequirement": "Mandatory"
  }
},
```

Example – getting implementation guidance

- How to I map the number of bytes used?

- RF/SF Property – Capacity.Data.ConsumedBytes
- From Table 5*

126	BlockSizeBytes mapping	210
127	Capacity.Data.ConsumedBytes mapping	211
128	Capacity.Data.ProvisionedBytes mapping	212

- From Table 127 – this is in section 6.5.2.2, part of Namespace
 - Description – The number of bytes consumed in this data store for this data type
 - Mandatory - Yes

* Multiple NVMe objects have a concept of capacity used so this shows up several places

What do I put in ConsumedBytes?

Property

Capacity.Da
ta.ConsumedBytes

**NVM Spec Property /
Field:** Namespace
Utilization (NUSE) **NVM
Spec: Section:Figure**
NVMe 1.4a: Section 5.15.2.1
(Identify Namespace),
Figure 247

- The mapping guide tries to point you to the exact NVMe construct
- Read the notes too, sometimes they have important information

Notes

Redfish/Swordfish

NVMe / NVMe-oF

Reporting capacity in bytes is the Redfish and Swordfish standard mechanism. Clients expect the capacity information to be reported consistently for these devices, so the calculation here is to convert the NVMe properties (in blocks) to bytes.

Returned in bytes 23:16 of the Identify Namespace Data Structure (NVM Command Set Specific). Reference NVMe Base Specification section n 5.15.2.1 and figure 247).

Combining the profiles and mapping guide

- Start with the baseline requirements from the profile
- Fill in properties using guidance from the mapping guide
- Review mockups from mapping guide and <http://SwordfishMockups.com> for additional recommended properties for some common configurations
- Review properties under each NVMe object in the mapping guide to identify optional properties you want to expose

How does it all come together?

For information and examples using Swordfish for creating and exporting logical NVMe subsystems watch the SDC presentation from Phil Caton

[Managing Exported NVMe-oF Resources and Fabrics in Swordfish and Redfish](https://storagedeveloper.org/events/sdc-2021/abstracts#resource-Cayton)

(<https://storagedeveloper.org/events/sdc-2021/abstracts#resource-Cayton>)



Getting Involved or Getting Started

Where to Find More Info..

SNIA Swordfish™

- **Swordfish Standards**
 - Schemas, Specs, Mockups, User and Practical Guide`s, ...
<https://www.snia.org/swordfish>
- **Swordfish Specification Forum**
 - Ask and answer questions about Swordfish
 - <http://swordfishforum.com/>
- **Scalable Storage Management (SSM) TWG**
 - Technical Work Group that defines Swordfish
 - Influence the next generation of the Swordfish standard
 - Join SNIA & participate: https://www.snia.org/member_com/join-SNIA
- **Join the SNIA Storage Management Initiative**
 - Unifies the storage industry to develop and standardize interoperable storage management technologies
 - <https://www.snia.org/forums/smi/about/join>

DMTF Redfish™

- **Redfish Standards**
 - Specifications, whitepapers, guides,...
<https://www.dmtf.org/standards/redfish>



Open Fabric Management Framework

- **OFMF Working Group (OFMFWG)**
 - Description & Links <https://www.openfabrics.org/working-groups/>
- **OFMFWG mailing list subscription**
 - <https://lists.openfabrics.org/mailman/listinfo/ofmfwg>
- **Join the Open Fabrics Alliance**
 - <https://www.openfabrics.org/membership-how-to-join/>



NVM Express

- **Specifications** <https://nvmexpress.org/developers/>
- **Join:** <https://nvmexpress.org/join-nvme/>





Please take a moment to rate this session.

Your feedback is important to us.