Disclaimer

• The information in this presentation represents a snapshot of work in progress within the DMTF.

• This information is subject to change without notice. The standard specifications remain the normative reference for all information.

• For additional information, see the DMTF website: www.dmtf.org
Getting involved in Redfish

• Redfish Standards page
  • Schemas, Specs, Mockups, White Papers & more
  • [http://www.dmtf.org/standards/redfish](http://www.dmtf.org/standards/redfish)

• Redfish Developer Portal
  • Redfish Interactive Resource Explorer
  • Educational material, documentation & other links
  • [http://redfish.dmtf.org](http://redfish.dmtf.org)

• Redfish User Forum
  • User forum for questions, suggestions and discussion
  • [http://www.redfishforum.com](http://www.redfishforum.com)

• DMTF Feedback Portal
  • Provide feedback or submit proposals for Redfish standards
  • [https://www.dmtf.org/standards/feedback](https://www.dmtf.org/standards/feedback)

• DMTF Redfish Forum
  • Join the DMTF to get involved in future work
  • [http://www.dmtf.org/standards/spmf](http://www.dmtf.org/standards/spmf)
Goal: Define new Cable resource

- Connectivity information
  - Upstream and downstream attachments
  - Cable types and supported protocols
  - Location within a physical system
  - Generic support for point-to-point, fan-out, and multipoint cables

- Assembly information
  - Part numbers, serial numbers, manufacturer, length, etc.
Assumptions on usage

• Cable Collection will probably be externally managed
  • Not all cables can be actively managed by a computer system: allow users to enter this information into the Redfish model
    • e.g. passive HDMI cable, power cables

• Cables can be proprietary and unique to the user
  • We will define enums for common types, while leaving freeform descriptions available for client-specific information not captured in the general types
Cable resource tree additions
Cable Definition: SATA Cable Example

```json
{
  "@odata.type": "#Cable.v1_0_0.Cable",
  "@odata.id": "/redfish/v1/Cables/sata0",
  "Id": "sata0",
  "Description": "SATA Cable",
  "UpstreamName": "SATA0",
  "DownstreamName": "sda",
  "CableDescription": "SATA",
  "UpstreamConnectorTypes": [ "SATA" ],
  "DownstreamConnectorTypes": [ "SATA" ],
  "LengthMeters": 0.05,
  "Links": {
    "UpstreamPorts": [
      { "@odata.id": "/redfish/v1/Systems/system/Storage/1/Controllers/1/Ports/1" }
    ],
    "DownstreamResources": [
      { "@odata.id": "/redfish/v1/Chassis/chassis/Drives/1" }
    ]
  },
  "PartNumber": "EZS2JM",
  "Manufacturer": "Satasaurus Co.",
  "SerialNumber": "123456",
  "Vendor": "Cablestore",
  "ReportableStatus": {
    "Health": "OK"
  }
}
```

- **Upstream/DownstreamName** are freeform string descriptions for a cable’s endpoints.
- **Upstream/DownstreamConnectorTypes** is an array to support multipoint-to-multipoint cables.
- Upstream/Downstream Links can be to Ports, Chassis, or Resources.
- **ReportableStatus** would allow user to update the status manually if needed.
Cable Definition: Power Cable Example

```json
{
    "@odata.type": "#Cable.v1_0_0.Cable",
    "@odata.id": "/redfish/v1/Cables/power0",
    "Id": "power0",
    "Name": "Main Power Cable",
    "Description": "Power Cable",
    "UpstreamName": "POWER",
    "DownstreamName": "Outlet",
    "ReportableStatus": "Present",
    "CableDescription": "Power",
    "LengthMeters": 0.5,
    "UpstreamConnectorTypes": [ "DCPower" ],
    "DownstreamConnectorTypes": [ "DCPower" ],
    "Links": {
        "UpstreamChassis": [ {
            "@odata.id": "/redfish/v1/Chassis/chassis"
        } ]
    },
    "PartNumber": "60320",
    "Manufacturer": "Farm To Cable Inc.",
    "SerialNumber": "123456",
    "Vendor": "Cablestore",
    "ReportableStatus": {
        "Health": "OK"
    }
}
```

Upstream/Downstream Links may not have an endpoint if the cable is connected to something not modelled in the Redfish implementation
Cable Definition: Optical Cable Example

```json
{
   "odata.type": "#Cable.v1_0_0.Cable",
   "odata.id": "/redfish/v1/Cables/splitter0",
   "Id": "splitter0",
   "Description": "Passive Optical Cable Splitter",
   "UpstreamName": "op_in",
   "DownstreamName": "op_out",
   "CableDescription": "Splitter",
   "UpstreamConnectorTypes": [ "QSFP" ],
   "DownstreamConnectorTypes": [ "QSFP", "QSFP", "QSFP", "QSFP" ],
   "Links": {
      "UpstreamPorts": [
         { "@odata.id": "/redfish/v1/Chassis/main/NetworkAdapters/0/Ports/1" }
      ],
      "DownstreamResources": [
         { "@odata.id": "/redfish/v1/Chassis/main/NetworkAdapters/0/Ports/1" }
      ]
   },
   "PartNumber": "EZS2JM",
   "Manufacturer": "Cellulose Networks",
   "SerialNumber": "123456",
   "Vendor": "Cablestore",
   "ReportableStatus": {
      "Health": "OK"
   }
}
```

- **CableDescription** is a freeform string and not an enum
- **Upstream/DownstreamConnectorTypes** is an array of enums to support multipoint-to-multipoint cables
- Upstream/Downstream Links are arrays to support multiple endpoints

© 2021 DMTF
Proposed Connector Type Enums

- Common connector types
  - AC Power
  - DC Power
  - DisplayPort
  - Ethernet
  - HDMI
  - ICI
  - IPASS
  - PCIe
  - Proprietary
  - RJ45
  - SATA
  - SCSI
  - SlimSAS
  - USB
  - USBC
  - QSFP

Prefer to use the name “Proprietary” over “OEM”
• **Upstream and Downstream links**
  - v1 will include links to one of Resource, Chassis, or Ports
  - Future revisions can include one of the v1 Links as a “superset” link to ensure backwards compatibility with v1 clients
  - Clients will know that they can always look at v1 resources to discover all relevant links

```json
"Links": {
    "DownstreamHyperdriveEngines": [
        {"@odata.id": "/redfish/v1/Chassis/Falcon/Engines/1"}
    ],
    "DownstreamResources": [
        {"@odata.id": "/redfish/v1/Chassis/Falcon/Engines/1"}
    ]
}
```

Hypothetical future Resource with a link to Cable. The link will include the future specific resource alongside a v1 resource for backwards compatibility
Areas for Feedback

- Are the basic fields sufficient in capturing common use cases?
- Are the enums sufficient in capturing most common types?
- Do we anticipate any issues with future extensibility?
Appendix: Proposed properties for Cable schema

- CableDescription: string
- LengthMeters: decimal
- DownstreamName: string
- UpstreamName: string
- Model: string
- Manufacturer: string
- Vendor: string
- SKU: string
- SerialNumber: string
- PartNumber: string
- AssetTag: string
- Up|DownstreamConnectorTypes: Collection(Cable.ConnectorType)
- ReportableStatus: string
- Location: Resource.Location
- Assembly: Assembly.Assembly
- Links:
  - Up|DownstreamChassis: Collection(Chassis.Chassis)
  - Up|DownstreamPort: Collection(Port.Port)
  - Up|DownstreamResource: Collection(Resource.Resource)