

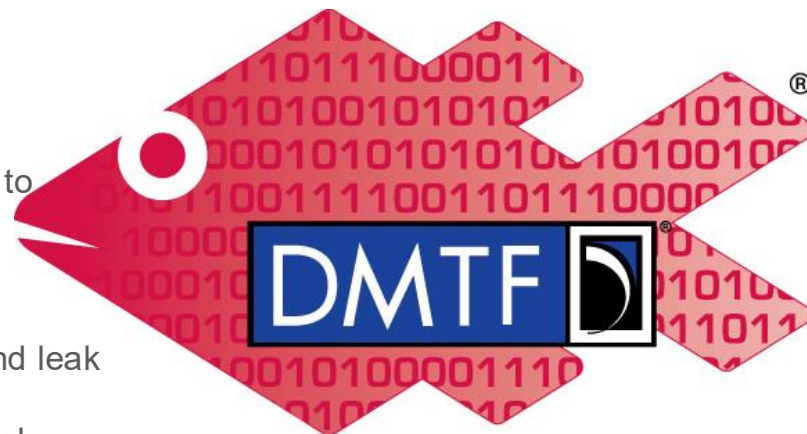


Redfish Release 2025.3

DMTF Redfish Forum
October 2025

Redfish Release 2025.3

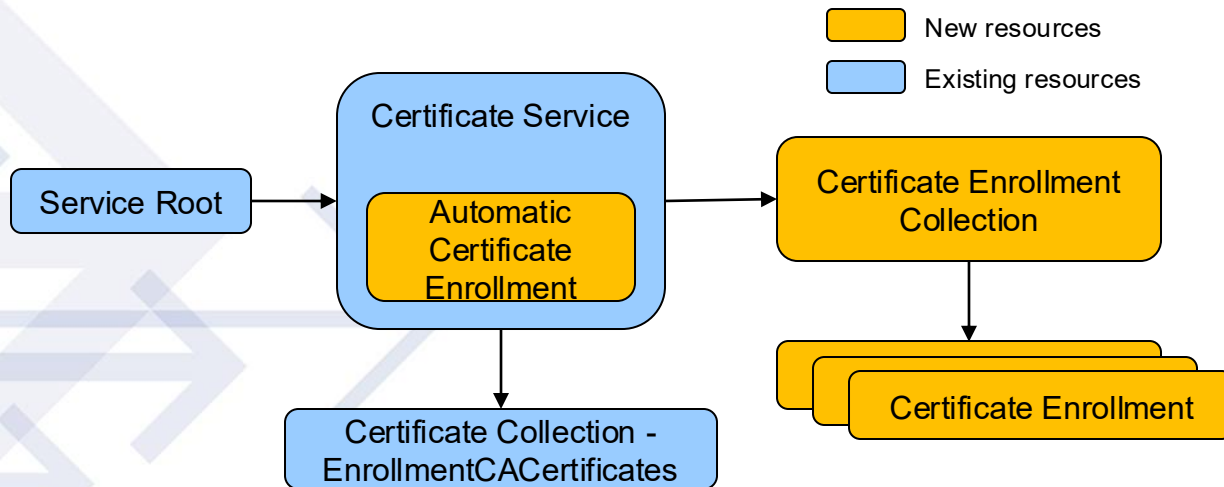
- Redfish Specification 1.23.0
 - Added a clause to describe using HTTP multipart operations to import a configuration file
 - Added a clause to define how a clients can connect to WebSockets provided by a service
- Redfish Message Registry Bundle 2025.3
 - DSP8011 contains all released Redfish registries
 - **Environmental** v1.3.0 adds messages for valves and leak detectors
 - **Power** v1.1.0 adds messages for chassis power level
 - **Privilege Registry** v1.8.0 adds messages for certificate enrollment
- Redfish Schema Bundle 2025.3
 - DSP8010 contains all released Redfish schemas, errata
- **NEW CertificateEnrollment** schema to support automatic certificate enrollment
- 41 Updated schemas (highlights)
 - Local image repository
 - Export and import of configurations in **ComputerSystem**, **CoolingUnit**, and **PowerDistribution**
 - Extended cooling modeling support in **CoolantConnector**, **CoolingLoop**, and **CoolingUnit**
 - Added port splitting support to **NetworkAdapter**, **Port**, and **PortMetrics**
 - Added support for the use of multiple physical functions to **Storage**, **StorageController**
- Download all published material at: <http://www.dmtf.org/standards/redfish>



Redfish

Automatic Certificate Enrollment

- **NEW CertificateEnrollment** resource to model automatic certificate enrollment using standard protocols such as ACME or SCEP
 - Each resource represents an automatic enrollment configuration
 - Contains enrollment parameters, such as CSR contents, ACME/SCEP settings, and the target certificate to renew
 - The service renews the certificate as it approaches its expiration



Other Updates for Automatic Certificate Enrollment

- **NEW** *AutomaticCertificateEnrollment* property added to **CertificateService**
 - Shows supported enrollment capabilities of the service
- **NEW** *CertificateEnrollment* property added to **CertificateService**
 - Resource collection to manage certificate enrollments
- **NEW** *EnrollmentCACertificates* property added to **CertificateService**
 - Resource collection to manage the CAs allowed to perform enrollment

Local Image Repository

- **NEW** *LocalImageStore* property added to **UpdateService**
 - Collection of software images stored by the service that a client can reference later, such as for mounting virtual media
- Updated the *SimpleUpdate* action and multipart HTTP push update parameters to add the *LocalImage* parameter
 - Clients set this flag to signify the update image is for the local image store
- Example usage with **VirtualMedia**

```
POST /redfish/v1/Systems/1/VirtualMedia/1/Actions/VirtualMedia.InsertMedia

{
  "Image": "/redfish/v1/UpdateService/LocalImageStore/ContosoLinux.iso"
}
```

Export and Import Configurations

- **NEW** *ExportConfiguration* action in **ComputerSystem**, **CoolingUnit**, and **PowerDistribution**
 - Allows a client to export the configuration of the equipment as a vendor-defined configuration file
 - The configuration file contains settings such as user account information, BIOS settings, network settings, and storage settings
 - Can be configured to support different use cases, such as migrating to a replacement device or applying common settings across a set of similar equipment
- **NEW** *MultipartImportConfigurationPushURI* property in **ComputerSystem**, **CoolingUnit**, and **PowerDistribution**
 - Allows a client to send a vendor-defined configuration file to the equipment
 - Applies configuration settings collected from the *ExportConfiguration* action
 - Transport follows the multipart HTTP push update design



Export Configuration Example

POST /redfish/v1/Systems/1/Actions/ComputerSystem.ExportConfiguration

```
{  
  "ExportType": "Replacement",  
  "Components": [  
    "All"  
  ],  
  "EncryptionPassphrase": "FluffyBunny23",  
  "Security": "IncludeSensitiveData"  
}
```

HTTP/1.1 204 No Content

Location: /redfish/v1/Systems/1/config.bin

Import Configuration Example

```
POST /redfish/v1/Systems/1/ImportConfiguration HTTP/1.1
```

```
Content-Type: multipart/form-data; boundary=-----d74496d66958873e
```

```
-----d74496d66958873e
```

```
Content-Disposition: form-data; name="ImportParameters"
```

```
Content-Type: application/json
```

```
{  
  "EncryptionPassphrase": "FluffyBunny23"  
}
```

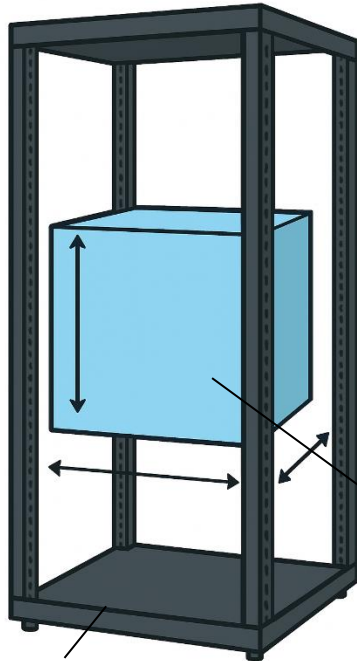
```
-----d74496d66958873e
```

```
Content-Disposition: form-data; name="ImportFile"; filename="config.bin"
```

```
Content-Type: application/octet-stream
```

```
<configuration file>
```


Chassis additions to describe rack mounting



- **NEW** describe the rack-mounting area of a “Rack” **Chassis**
 - These describe the capacity to hold rack-mountable equipment
 - Different (smaller) dimensions than the rack **Chassis** itself
- *RackMountCapacityUnits* : “Rack Unit” space provided
- *RackMountWidth* : Spec-based width and hole pattern
- *RackMountDepthMm* : Max chassis depth supported
- **NEW** For any **Chassis**, describe height in “rack units”
- *RackUnits* : EIA 1.75” or OpenU 48mm rack units
- *HeightRackUnits* : “Unit” based chassis height

“RackMountCapacityUnits”: 42,
“RackMountWidth”: “EIA_310”,
“RackMountDepthMm”: 855,
“RackUnits”: “EIA_310”,
“HeightMm”: 1988,
“WidthMm”: 560,
“DepthMm”: 960,

“RackUnits”: “EIA_310”,
“HeightRackUnits”: 8,
“HeightMm”: 354,
“WidthMm”: 482,
“DepthMm”: 533,

CoolantConnector and LeakDetection enhancements

- **NEW** *ValveControl* action in **CoolantConnector**
 - Allows control of valve state independent of set point
 - *ValveState* : “Open” or “Closed”
 - *ValveStateReason*: Provide (optional) reason for the action taken
 - “Normal” – normal operation
 - “NotInUse” – Closed with no connection (an “Absent” state)
 - “LeakDetected” – Closed due to a detected or reported leak
- Additional states added to *LeakDetectorState* in **LeakDetector**
 - Clarifies the reporting of “non-normal” states from a leak detector device
 - “OK”, “Warning”, “Critical” – existing states for normal or leak detected
 - “Unavailable” – new state for a faulty leak detector
 - “Absent” – no leak detector installed (but capability exists)

NEW Port Splitting for Network Adapters

- Adds support for dividing the lanes of physical network ports into one or more subports
- **NetworkAdapter.PortSplitting** adds properties for port splitting:
 - **MaximumSubports** – maximum subports supported by a network adapter
 - **MaximumSubportsPerPort** – maximum split factor for a port
 - **SupportedConfigurations** – a list of port splitting configurations the network adapter supports
 - **CurrentConfiguration** – a selection from the list of supported configurations
- **Port** schema adds properties to show splitting:
 - **IsSplit** – indicates whether a port is a physical port or a subport
 - **Width** (existing) – reflects the number of lanes dedicated to a port
 - **FirstLane** – indicates the first lane dedicated to a port or subport

Port Splitting Configurations

- Configurations consist of one or more subconfigurations, each of which details the splitting for one or more physical ports.
 - Collectively, the subconfigurations cover all physical ports
 - Ports do not all have to be split the same way
- **Subconfiguration** properties:
 - **StartingPhysicalPort** – the first port that a subconfiguration applies to
 - **EndingPhysicalPort** – the last port that a subconfiguration applies to
 - **FirstSubportID** – starting point for logical port numbers assigned to subports
 - IDs are consecutive within a subconfiguration
 - IDs may be skipped going from one subconfiguration to the next
 - **SubportsPerPort** – split factor for physical ports
 - **Lanes** (array) – number of lanes dedicated to each subport
 - **LinkSpeedGbps** (array) – bandwidth for each subport

Metric Support for Subports

- **Transceiver** object is updated
 - **RXPowerMilliwatts**, **TxBiasCurrentMilliAmps**, **TXOutputPowerMilliWatts**, **SupplyVoltage** now all reflect aggregate values across all lanes for the port/subport
- **Transceiver.ByLane** object added to give lane-by-lane breakdown
 - **LaneID** – indicates the lane number
 - **RXPowerMilliwatts**, **TxBiasCurrentMilliAmps**, **TXOutputPowerMilliWatts** give the values for the specific lane



NEW MPF Support for Storage Controllers

- Adds support for storage controllers that use multiple physical functions (MPF) to parallelize data transfer
- Organized by volume, each linked to a specific physical function
- Physical functions are modeled as logical storage controllers
- **Storage.MPF** collects properties specific to these controllers:
 - **MaximumSupportedPhysicalFunctions** – maximum number of PFs a controller can support
 - **ConfiguredPhysicalFunctions** – current number of PFs the controller is configured to use
 - **VolumeAssignmentPolicy** – controls how newly created volumes are assigned to PFs
 - **Unassigned** – no assignment
 - **Supervisor** – assigned to a dedicated supervisor function
 - **WeightedRoundRobin** – spread out evenly

MPF Support for Storage Controllers (ctd.)

- **Storage.Links** adds **UnassignedVolumes** to collect volumes that have not been assigned to a PF
- **StorageController.AttachNamespace** assigns a volume to a PF
- **StorageController.DetachNamespace** removes a volume from a PF (and adds it to the list of **UnassignedVolumes**)
- **StorageController.IsLogical** indicates whether a **StorageController** is a logical entity such as an NVMe controller or a physical function
- **StorageController.MPFProperties** contains MPF-specific properties:
 - **IsSupervisor** – indicates whether the physical function can perform supervisory duties

Message Registry Additions (1 of 3)

- **Base - 1.22.0**
 - Deprecated *AccessForbidden* in favor of the existing *InsufficientPrivilege* message
- **Environmental - 1.3.0**
 - Added *LeakDetectorFailure*, *LeakDetectorConnected*, *LeakDetectorDisconnected*, *LeakDetectorRestored*, *ValveClosed*, *ValveOpened*, *ValvePositionChanged*, *ValveUnableToReachSetPoint*, and *ValveUnableToReachSetPointPercent* messages
 - Corrected the *ClearsMessage* entries for *FanGroupNormal*
- **HeartbeatEvent - 1.1.1 (Errata)**
 - Added missing version tags for the *RedfishServiceStarted* and *RedfishServiceShuttingDown* messages
- **Platform - 1.1.1 (Errata)**
 - Added missing version tags for the *NVRAMClearAsserted* and *SecurityBypassAsserted* messages
- **Platform - 1.3.1 (Errata)**
 - Corrected the data type of the *NumberOfArgs* term for many messages
 - Removed errant space at the end of the *PCieLinkRestored* message identifier

Message Registry Additions (2 of 3)

- **Power - 1.1.0**
 - Added *ChassisPoweredOn* and *ChassisPoweredOff* messages
- **Power - 1.0.3 (Errata)**
 - Corrected the *ClearsMessage* entries for *PowerSupplyGroupNormal*
- **Redfish Privilege Registry - 1.8.0**
 - Added entries for new *CertificateEnrollment* and *CertificateEnrollmentCollection* schemas
- **Redfish Privilege Registry - 1.7.1 (Errata)**
 - Added missing subordinate override statement for *EthernetInterfaceCollection*
- **SensorEvent - 1.1.0**
 - Added *SensorConnected* and *SensorDisconnected* messages
- **StorageDevice - 1.5.1 (Errata)**
 - Clarified the expected user actions needed for *WriteCacheTemporarilyDegraded* and *WriteCacheDegraded* messages
 - Added missing version tags for the *VolumeRebuilding* and *VolumeReconfiguring* messages

Message Registry Additions (3 of 3)

- **Update - 1.1.1 (Errata)**
 - Added missing version tag for the *NoTargetsDetermined* message
- **Update - 1.2.1 (Errata)**
 - Added missing version tags for the *NoTargetsDetermined*, *UpdateSkipped*, *UpdateSkippedSameVersion*, and *UpdateNotApplicable* messages
- **Update - 1.3.0**
 - Added *UpdateSkippedVersionTooLow* message
- **All Registry Files**
 - Added Release property to every minor message registry release, showing the associated DSP8011 release bundle version

Schema and Registry Guide documents

- **Redfish Data Model Specification (DSP0268)**
 - Document previous titled “Redfish Schema Supplement”
 - Now includes normative statements (“LongDescription”) and informative description details from schema in a single document
 - Intended for both Redfish Service and client-side developers
- **Redfish Resource and Schema Guide (DSP2046)**
 - Presents schema (data model) contents in a more friendly format for end users
 - Includes example payloads for each resource type
- **Redfish Message Registry Guide (DSP2065)**
 - Presents message registry definitions in a more human-readable format
 - Includes summary table and individual message details
- **Redfish Property Guide (DSP2053)**
 - Provides an alphabetical list of all properties defined in Redfish schema
 - Useful for schema writers to locate existing definitions or properties
 - Helps avoid re-defining property names already in use

Available Redfish conformance testing tools

- DMTF Redfish Forum provides open source tools for service developers to validate their conformance with the Redfish protocol, data model, and profiles
- **Redfish Protocol Validator**
 - Tests a live service for conformance to the Redfish HTTP protocol, including response headers and status codes
 - <https://github.com/DMTF/Redfish-Protocol-Validator>
- **Redfish Service Validator**
 - Tests a live service for conformance with Redfish schemas, ensuring the returned JSON payloads validate against the standard data models
 - Recommend that developers run this tool first, as errors in payload are more likely to cause issues for end users and interoperability
 - <https://github.com/DMTF/Redfish-Service-Validator>
- **Redfish Interop Validator**
 - Tests a service against a Redfish interoperability profile
 - <https://github.com/DMTF/Redfish-Interop-Validator>

Redfish-Publications repository

- Public GitHub repository contains an official read-only copy of the Redfish schemas and standard message registries
 - <https://github.com/DMTF/Redfish-Publications>
 - Creates public, durable locations for referencing specific schema or registry items in issue reports, forum postings, or other online references
 - Allows developers to automatically synchronize with new Redfish releases using normal GitHub tools and processes
- Repository will be updated as each Redfish release become public
- Contains materials published as DSP8010 and DSP8011
 - **/csdl** – Redfish schemas in OData CSDL XML format
 - **/json-schema** – Redfish schemas in JSON Schema format
 - **/openapi** – Redfish schemas in OpenAPI YAML format
 - **/dictionaries** – RDE dictionaries
 - **/registries** – Redfish standard message registries

Redfish Schema

MINOR REVISION DETAILS

Redfish Schema Minor Revisions (1 of 7)

- **ActionInfo** - v1.5.0
 - Added DefaultValue and NoDefaultValue properties
- **Assembly** - v1.6.0
 - Added ReadyToRemove property
- **AttributeRegistry** - v1.4.0
 - Added RequiredPrivileges to Attributes
- **Battery** - v1.5.0
 - Added ServicedDate property
- **BatteryMetrics** - v1.1.0
 - Added CRate and ERate properties
- **Certificate** - v1.11.0
 - Added ForceAutomaticRenew action
- **CertificateEnrollment** - v1.0.0
 - Resource that contains automatic certificate enrollment configuration

Redfish Schema Minor Revisions (2 of 7)

- **CertificateService - v1.2.0**
 - Added "AutomaticCertificateEnrollment", "CertificateEnrollments", and "EnrollmentCACertificates" properties
- **Chassis - v1.28.0**
 - Added "ReadyToRemove", "RackUnits", "RackMountCapacityUnits", "RackMountWidth", "RackMountDepthMm", and "HeightRackUnits" properties
- **ComputerSystem - v1.26.0**
 - Added "MultipartImportConfigurationPushURI" property
 - Added "ExportConfiguration" action
 - Added "WebSocket" to "HostConsole"
- **CoolantConnector - v1.3.0**
 - Added "Manufacturer", "Model", "SKU", "SerialNumber", "PartNumber", and "SparePartNumber" properties
 - Added "ValveControl" action
- **CoolingLoop - v1.1.0**
 - Added "CoolingLoopType" property

Redfish Schema Minor Revisions (3 of 7)

- **CoolingUnit - v1.4.0**
 - Added "RatedThermalLossToAirWatts" and "MultipartImportConfigurationPushURI" properties
 - Added "ExportConfiguration" action
- **CXLLogicalDevice - v1.3.0**
 - Added "SetPassphrase", "DisablePassphrase", "PassphraseSecureErase", "FreezeSecurityState", and "Unlock" actions
- **DriveMetrics - v1.3.0**
 - Added "LifetimeStartDateTime" property
- **Event - v1.12.0**
 - Added "OriginOfConditionUnavailable" property
- **Fan - v1.6.0**
 - Added "RatedSpeedRPM" and "RatedSecondarySpeedRPM" properties
- **LeakDetector - v1.5.0**
 - Added "Unavailable" and "Absent" values to "DetectorState"
 - Clarified how "Enabled" will affect "DetectorState"

Redfish Schema Minor Revisions (4 of 7)

- **LogEntry** - v1.20.0
 - Added "OriginOfConditionUnavailable" property
- **Manager** - v1.23.0
 - Added "ReadyToRemove" property
- **Memory** - v1.22.0
 - Added "GDDR7" to "MemoryDeviceType"
- **MemoryMetrics** - v1.8.0
 - Added "LifetimeStartDateTime" property
- **MessageRegistry** - v1.7.0
 - Added "Release" property
- **NetworkAdapter** - v1.13.0
 - Added "PortSplitting" property
- **Port** - v1.17.0
 - Added "IsSplit", "FirstLane", and "AssociatedPhysicalPort" properties
 - Added "PBRComponent" to "ConnectedDeviceType"

Redfish Schema Minor Revisions (5 of 7)

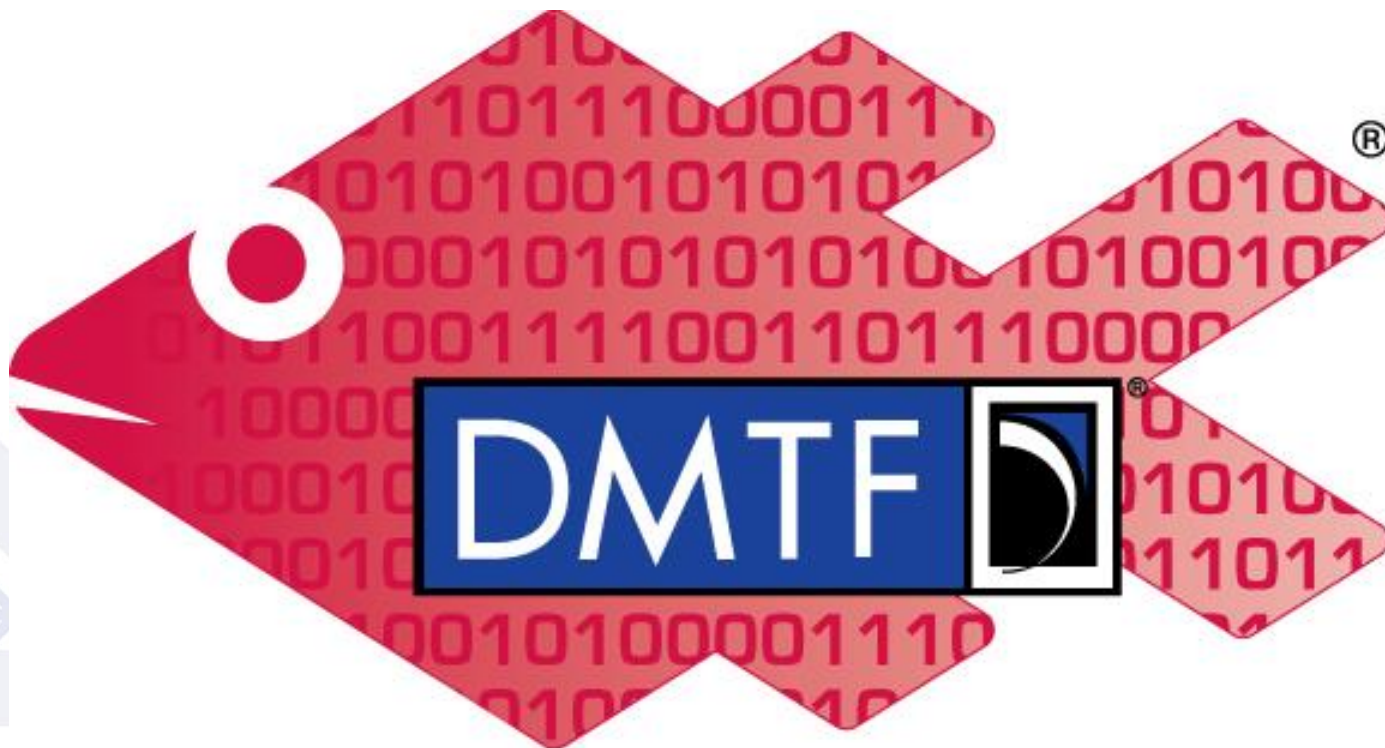
- **PortMetrics - v1.8.0**
 - Added "PCleMetrics" and "ByLane" properties
- **PowerDistribution - v1.5.0**
 - Added "MultipartImportConfigurationPushURI" property
 - Added "ExportConfiguration" action
- **PrivilegeRegistry - v1.2.0**
 - Added "OEMPrivilegeDescriptions" property
- **Processor - v1.21.0**
 - Added "TotalEnabledThreads" property
 - Added "GDDR7" to "ProcessorMemoryType"
- **ProcessorMetrics - v1.7.0**
 - Added "LifetimeStartDateTime" property
- **Redundancy - v1.5.0**
 - Added "ActiveRedundancyGroup" property to "RedundantGroup"
 - Added "ActiveRedundancySet" to "Redundancy"

Redfish Schema Minor Revisions (6 of 7)

- **Resource - v1.23.0**
 - Added "Room" and "FacilityName" to "Placement"
- **SoftwareInventory - v1.13.0**
 - Added "ImageDataURI" and "SizeBytes" properties
- **SoftwareInventoryCollection (not versioned)**
 - Updated the resource capabilities to show resources can be deleted
 - Added new URIs for "SoftwareInventory" resources referenced by "LocalImageStore"
- **Storage - v1.19.0**
 - Added "MPF" object containing "MaximumSupportedPhysicalFunctions", "ConfiguredPhysicalFunctions", and "VolumeAssignmentPolicy"
 - Added "UnassignedVolumes" to "Links"
- **StorageController - v1.10.0**
 - Updated "AttachNamespaces" and "DetachNamespaces" to reference volumes
 - Added "IsLogical" property
 - Added "MPFProperties" object containing "IsSupervisor"

Redfish Schema Minor Revisions (7 of 7)

- **StorageControllerMetrics** - v1.1.0
 - Added "LifetimeStartDateTime" and "MPF" properties
 - Added "UnassignedVolumes" to "Links"
- **SwitchMetrics** - v1.1.0
 - Added "LifetimeStartDateTime", "MPFProperties", and "IsLogical" properties
- **ThermalSubsystem** - v1.5.0
 - Added "FansFullSpeedOverrideEnable" property
- **UpdateService** - v1.17.0
 - Added "LocalImage" and "ExcludeTargets" to "SimpleUpdate" and multipart HTTP push updates
 - Added "LocalImageStoreAvailableCapacityBytes", "LocalImageStoreTotalCapacityBytes", and "LocalImageStore" properties



Redfish