Redfish Release 2023.2

- **Redfish Specification v1.19.0**
  - Added Atomic Password Changes clause
  - Added Outbound Connections clause
  - Updated Session Login clause to define behavior when a token is required but not provided

- **Redfish Specification v1.18.1 (errata release)**
  - Recommend weak Etag compares for PUT and PATCH
  - Updated Deep Operations clause to show deep POST may include different types of resources
  - Recommend $expand responses include entire resource

- **Redfish Schema Bundle 2023.2**
  - DSP8010 contains all released Redfish schemas

- **Redfish Message Registry Bundle 2023.2**
  - DSP8011 contains all released Redfish registries
  - Added OneTimePasscodeSent to Base message registry

- **6 new schemas**
  - Application, Container, ContainerImage, OperatingSystem for container, operating system, and application management
  - MemoryRegion to support CXL dynamic capacity devices (DCD)
  - OutboundConnection for enabling connections to cloud-based services

- **31 updated schemas (highlights)**
  - Added PhysicalAddress to Location
  - Added Decommission action to ComputerSystem
  - Added RemoteServerSSHKeys to UpdateService
  - See release notes in DSP8010 for errata details

- **Download all published material at**: [http://www.dmtf.org/standards/redfish](http://www.dmtf.org/standards/redfish)

Copyright 2023 © DMTF
Schema and Registry Guide documents

- Redfish Data Model Specification (DSP0268)
  - Document was 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](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](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](https://github.com/DMTF/Redfish-Interop-Validator)
NEW 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](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
NEW Operating System, Application, Container models

- New resources model operating systems and application software
  - Developed with input from the Open Process Automation™ Forum
    - https://www.opengroup.org/forum/open-process-automation-forum
- OperatingSystem - Contains information about the operating system running on the system, including virtualization engines
- Application - Represents a software instance running on a system
- ContainerImage - Represents an available container image
- Container - Represents a container instance running on a system
  - Also contains its own EthernetInterfaceCollection to represent network connectivity of the container
- Existing ComputerSystem model represents virtual machines
  - A virtual machine is identified with SystemType as “Virtual”
  - New links added to show hosting ComputerSystem for the virtual machine
NEW Operating System, Application, Container Model
NEW Outbound Connections

- Internet-based applications or other “cloud-based” services cannot typically initiate HTTP connections to Redfish-enabled devices
  - Firewalls prevent inbound connections unless specifically configured to allow for specific use cases – and this is cumbersome and problematic
  - Devices will instead initiate an outbound connection, typically using WebSockets, to communicate with Internet-hosted clients
  - This avoids need to configure firewalls or other networking infrastructure
- **OutboundConnection** provides a standardized means to configure and establish a Redfish **Session** using a WebSocket
  - New WebSocket mechanism defined for Redfish HTTP operations
    - IANA-registered “Redfish” WebSocket sub-protocol
  - Initiated by the Redfish service, an outbound connection is made to a specified Internet / cloud-based client (hosted web service)
  - Client uses the WebSocket to perform operations and receive responses
Outbound Connection Usage

- An administrator configures the Redfish service to establish a persistent WebSocket connection to a client outside of the firewall
  - Provides URI, required info for external client connection, certificates, etc.
- WebSocket connection is initiated by the Redfish-enabled device
  - The Manager responsible for the Redfish Service opens the WebSocket
- The external client sends Redfish requests using the WebSocket
Additional password handling mechanisms

- **NEW** One-time Passcode support for multi-factor authentication
  - Allows service to generate and deliver a one-time passcode (OTP)
  - Users attempting to create sessions will generate an OTP
    - Initial attempt will fail with the `OneTimePasscodeSent` message returned
    - OTP is sent via email to `OneTimePasscodeDeliveryAddress` from `ManagerAccount`
    - User retrieves OTP from email, uses OTP value as `Token` to create a session

- **NEW** `ChangePassword` action in `ManagerAccount`
  - Provides means for atomic password change
  - May also apply to initial settings for accounts with `PasswordChangeRequired`
  - Action with required `Password` and `SessionAccountPassword` parameters
    - User must provide password associated with the current session
  - `RequireChangePasswordAction` added to `AccountService`
    - Indicates service requires use of the atomic `ChangePassword` action
    - If `true`, service will not allow a PATCH to `Password`
Enhancements to Location for “Physical” Address

- _PostalAddress_ definition in common _Location_ followed RFC5139
- But modern usage makes the cumbersome structure unnecessary
  - Mapping, search, and location APIs don’t use individual partial address properties
  - Developers don’t know which properties to provide or what they can depend on
- **NEW** _PhysicalAddress_ replaces _PostalAddress_ (now deprecated)
  - Fewer properties needed to render a complete address

```
"Location": {
    "PhysicalAddress": {
        "StreetAddress": "95 W Oak #9740",
        "City": "Houston",
        "StateOrProvince": "Texas",
        "Country": "United States",
        "PostalCode": "77070-3002",
        "ISOCountryCode": "USA",
        "ISOSubdivisionCode": "TX"
    }

    // Program-friendly ISOCountryCode and ISOSubdivisionCode
    // Utilize ISO 3166-1 and 3166-2 codes
    // Avoids issues with string matching due to abbreviations, mixed languages, etc.

    // Expect implementations to support “all or none” for interoperability
```
Deprecation of PCIeSlots resource

- PCIeSlots was a mostly static resource that provided information about PCIe slots
- **Slot** object available in **PCIeDevice** provides slot capabilities in context of an installed device
- Chassis includes empty PCIe slots as **PCIeDevice** resources with an “Absent” **State**
  - Shows capabilities of the chassis
  - Provides that data without a separate resource / code path

```json
"odata.type": "#PCIeDevice.v1_12_0.PCIeDevice",
"Id": "Slot2",
"Name": "PCIe Slot #2 (Empty)",
"Status": {
  "State": "Absent"
},
"Slot": {
  "PCIeType": "Gen4",
  "Lanes": 4,
  "SlotType": "FullLength",
  "Location": {
    "PartLocation": {
      "ServiceLabel": "Slot 2",
      "LocationOrdinalValue": 2,
      "LocationType": "Slot",
      "Orientation": "LeftToRight",
      "Reference": "Rear"
    }
  }
}
```
Redfish Schema Minor Revisions (1 of 5)

- **AccountService v1.14.0**
  - Added `RequireChangePasswordAction` and `OutboundConnections`
  - Added `OneTimePasscode` to `MFABypassType`
  - Added `EmailAttribute` to `SearchSettings` in `LDAPService`

- **ActionInfo v1.3.0**
  - Added `AllowableValueDescriptions` to `Parameters`

- **Certificate v1.8.0**
  - Added `IDevID`, `LDevID`, `IAK`, and `LAK` to `CertificateUsageType`

- **Chassis v1.24.0**
  - Added `Doors`
  - Deprecated `PCIeSlots` in favor of `PCIeDevices`

- **ComputerSystem v1.21.0**
  - Added `OperatingSystem`
  - Added `HostingComputerSystem` and `VirtualMachines` to `Links`
  - Added `Decommission` action
  - Deprecated “OS” for `SystemType` in favor of new `OperatingSystem` resource
Redfish Schema Minor Revisions (2 of 5)

- Connection v1.3.0
  - Added MemoryRegionInfo
- CoolingUnit v1.1.0
  - Added CoolantConnectorRedundancy
- CXLLogicalDevice v1.1.0
  - Added MemoryRegions
- DriveMetrics v1.1.0
  - Added NativeCommandQueueDepth
- EthernetInterface v1.11.0
  - Added RoutingScope
- Event v1.9.0
  - Added Resolution and OEMDiagnosticDataType
- ExternalAccountProvider v1.7.0
  - Added EmailAttribute to SearchSettings in LDAPService
- Fan v1.4.0
  - Added SecondarySpeedPercent
Redfish Schema Minor Revisions (3 of 5)

- **Key v1.3.0**
  - Added `RemoteServerHostName` to `SSH`
- **LogService v1.5.0**
  - Added `CXL` to `LogEntryType`
- **ManagerAccount v1.11.0**
  - Added `EmailAddress`, `PhoneNumber`, and `OneTimePasscodeDeliveryAddress`
  - Added `ChangePassword` action
- **Memory v1.18.0**
  - Added `MemoryRegionMediaSources` to `Links`
- **MemoryChunks v1.6.0**
  - Added `MemoryRegions` to `Links`
- **PCIeDevice v1.6.0**
  - Added `LocationIndicatorActive`
  - Added `Processors` to `Links` and `HotPluggable` to `Slot`
  - Added `DynamicCapacity` object to `CXLDevice`
Redfish Schema Minor Revisions (4 of 5)

- **PCleSlots v1.6.0**
  - Deprecated schema in favor of "PCleDevice" schema
- **Port v1.10.0**
  - Deprecated "Enabled" in favor of existing "InterfaceEnabled"
- **PortMetrics v1.5.0**
  - Added "PhyResetProblemCount" to SAS
- **Resource v1.17.0**
  - Added "PhysicalAddress" object to "Location"
  - Deprecated "PostalAddress" in favor of new "PhysicalAddress" property in "Location"
- **SerialInterface v1.2.0**
  - Added "Links"
- **Session v1.7.0**
  - Added "Roles" and "Links"
  - Added "OutboundConnection" to "SessionType"
Redfish Schema Minor Revisions (5 of 5)

- **SoftwareInventory v1.10.0**
  - Added `ReleaseType` and `AssociatedPhysicalContext`
- **ThermalEquipment v1.1.0**
  - Added `CoolingLoopRedundancy`
- **ThermalMetrics v1.3.0**
  - Added `PowerWatts` and `EnergykWh`
- **ThermalSubsystem v1.3.0**
  - Added `Pumps`, `LeakDetection`, and `CoolantConnectorRedundancy`
- **TrustedComponent v1.2.0**
  - Added `Owner` to `Links`
- **UpdateService v1.12.0**
  - Added `RemoteServerSSHKeys` and `VerifyRemoteServerSSHKey`
Redfish Schema Errata Revisions (1 of 2)

- **AccountService v1.x.x and ExternalAccountProvider v1.x.x**
  - Updated descriptions for the `AccountProviderType` values to clarify how each value affects the format of `ServiceAddresses`
  - Expanded description for the LDAP value to show it can contain a scheme and port

- **LogEntry v1.x.x**
  - Clarified that `Resolved` should be ignored if `Severity` contains “OK”

- **Memory v1.x.x**
  - Corrected the usage of volatile and non-volatile in the descriptions for `StagedVolatileSizeMiB` and `StagedNonVolatileSizeMiB`

- **Port v1.x.x**
  - Clarified the usage of `InterfaceEnabled` and `LinkState`

- **Sensor v1.x.x**
  - Added long descriptions to the values for `SensorType`
Redfish Schema Errata Revisions (2 of 2)

- (various files) v1.x.x
  - Clarified the usage of “number” in `ParamTypes` to show that the message argument is converted to a string representation of the number
  - Removed normative language from non-normative descriptions
  - Clarified the range of possible values for properties with percent units to show that it's typically `0` to `100`