



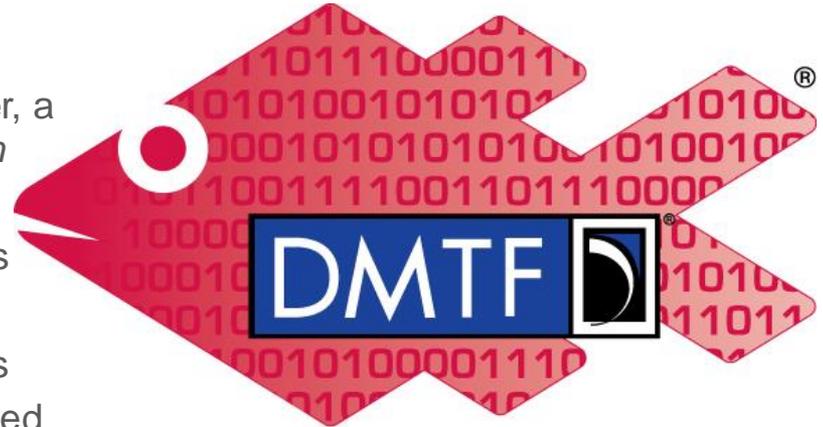
# Redfish Release 2025.1

DMTF Redfish Forum  
March 2025



# Redfish Release 2025.1

- Redfish Specification 1.22.0
  - Added *?includeoriginofcondition* query parameter, a specialized *\$expand* query for *OriginOfCondition*
- Redfish Schema Bundle 2025.1
  - DSP8010 contains all released Redfish schemas
- Redfish Message Registry Bundle 2025.1
  - DSP8011 contains all released Redfish registries
  - Base v1.21.0 adds messages for password-related errors (recent re-use, lack of complexity or length)
- 14 Updated schemas (highlights)
  - See release notes in DSP8010 for errata details
  - Added password complexity guidance to **AccountService**
  - Added *ConfiguredSpeedGbps* and *ConfiguredWidth* to **Port**
  - Added *SecurityMode* in **Manager** to report standardized security operating modes, and *UpdateSecurityMode* action to configure those modes
- Download all published material at: <http://www.dmtf.org/standards/redfish>



# Redfish



## **NEW** ?*includeoriginofcondition* query parameter

- Desire to support the *IncludeOriginOfCondition* option for **EventDestination** instances using server-sent eventing (SSE)
  - This functionality is useful for event receivers to obtain context of an event, without requiring subsequent GET operations
    - The resource linked by *OriginOfCondition* is included in the payload
  - For SSE, unlike other *SubscriptionTypes*, this must be provided on the URI
- New ?*includeoriginofcondition* query parameter
  - Allows this functionality on any resource with an *OriginOfCondition*
    - For example, in *Conditions[]* within *Status*
    - Does not apply to *OriginOfCondition* inside of *Links* (e.g. **LogEntry**)
- Functionality can be discovered in **EventService** and **ServiceRoot**
- SSE example:
  - <https://webserver42.contoso.com/redfish-sse?includeoriginofcondition>



## NEW Password complexity guidance

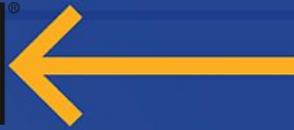
- A myriad of “required password complexity” rules exist, which are difficult to explain in a standard fashion
  - Expect that any attempt to describe these rules would result in a highly complex, and likely unusable, set of properties
  - But still a desire to make these rules discoverable by humans
- Two new properties added to **AccountService** to report these rules
- *PasswordGuidanceMessage* – Vendor-provided string describing the required password complexity rules for the service
- *PasswordGuidanceMessageId* – Vendor-provided *MessageId* to reference the message registry source for this message
  - Allows for localization (translations provided by vendor) of the password guidance message string

## NEW Password complexity guidance example

```
{
  "@odata.type": "#AccountService.v1_18_0.AccountService",
  "Id": "AccountService",
  "Name": "Account Service",
  "ServiceEnabled": true,
  "AuthFailureLoggingThreshold": 3,
  "MinPasswordLength": 8,
  "EnforcePasswordHistoryCount": 5,
  "PasswordGuidanceMessageId": "ContosoService.1.2.PasswordComplexity",
  "PasswordGuidanceMessage": "Password must contain at least three of the following: an uppercase character, a lowercase character, a numeric digit, or a punctuation character.",
  "AccountLockoutThreshold": 5,
  "AccountLockoutDuration": 30,
  "AccountLockoutCounterResetAfter": 30,
  "AccountLockoutCounterResetEnabled": true,
  "Accounts": {
    "@odata.id": "/redfish/v1/AccountService/Accounts"
  },
  "Roles": {
    "@odata.id": "/redfish/v1/AccountService/Roles"
  },
  "LocalAccountAuth": "Enabled",
  "RequireChangePasswordAction": false,
  "@odata.id": "/redfish/v1/AccountService"
}
```

*MessageId* from an OEM-defined message registry (in any vendor-supported language) that provides the password guidance message text

Password complexity requirements and any other guidance text (in default language for the service), which can be displayed to the user.



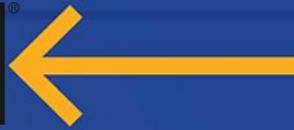
## Port additions and clarifications for autonegotiation

- **NEW** Added *ConfiguredSpeedGbps* and *ConfiguredWidth* to enable forced training to a link speed or width without using autonegotiation
- **NEW** Added *LinkNetworkTechnology* within *LinkConfiguration* to specify technology used with an autonegotiation configuration option
  - Updated description of the root-level *LinkNetworkTechnology* to clarify that it represents the current technology (in use)
- Clarified that an empty list of *ConfiguredNetworkLinks[]* indicates that autonegotiation is unconstrained



## 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>
- **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 3)

- AccountService v1.18.0
  - Added *PasswordGuidanceMessage* and *PasswordGuidanceMessageId*
- ComputerSystem v1.24.0
  - Added “TPM” to the *DecommissionType* parameter of the *Decommission* action
- EnvironmentMetrics v1.4.0
  - Added *AmbientTemperatureCelsius*
- EventService v1.11.0
  - Added *SSEIncludeOriginOfConditionSupported*
- LogEntry v1.18.0
  - Added “Device” to *DiagnosticDataType*
- LogService v1.8.0
  - Added “Device” to *DiagnosticDataType*
  - Added *TargetDevice* to the *CollectDiagnosticData* action

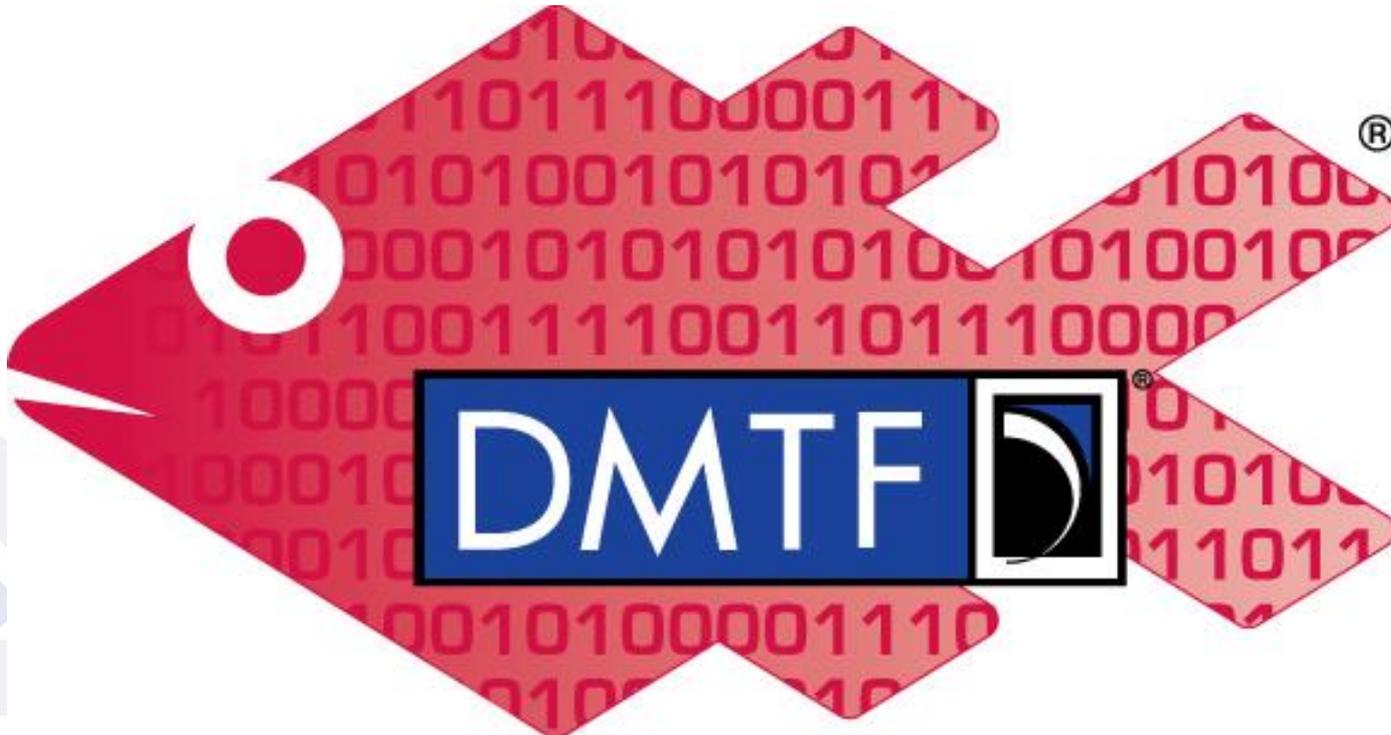
## Redfish Schema Minor Revisions (2 of 3)

- **Manager v1.21.0**
  - Added *SecurityMode* and *OEMSecurityMode*
  - Added *UpdateSecurityMode* action
- **ManagerNetworkProtocol v1.11.0**
  - Added *mDNS*
- **NetworkDeviceFunction v1.10.0**
  - Added *AdditionalProtocols* to *Ethernet* and clarified the usage of various properties
- **PCleDevice v1.18.0**
  - Added *CurrentProtocolVersion* and *CapableProtocolVersions* to *CXLDevice*
- **Port v1.16.0**
  - Added *ConfiguredSpeedGbps* and *ConfiguredWidth*
  - Added *LinkNetworkTechnology* to *LinkConfiguration*



## Redfish Schema Minor Revisions (3 of 3)

- PowerDistributionMetrics v1.4.0
  - Added *AmbientTemperatureCelsius*
- Protocol
  - Added “UEC” as a protocol
- ServiceRoot v1.18.0
  - Added *IncludeOriginOfConditionQuery* to *ProtocolFeaturesSupported*
- SoftwareInventory v1.11.0
  - Added *FactoryConfiguration* and *BootParameters* to *AdditionalVersions*



# Redfish