DRAFT International Standard

ISO/IEC DIS 30115-1

Information technology — Redfish scalable platforms management API specification —

Part 1: Redfish Specification v1.13.0

ICS: 35.020

ISO/IEC JTC **1**

Secretariat: **ANSI**

Voting begins on:
2025-05-23

Voting terminates on:
2025-08-15

THIS DOCUMENT IS A DRAFT CIRCULATED FOR COMMENTS AND APPROVAL. IT IS THEREFORE SUBJECT TO CHANGE AND MAY NOT BE REFERRED TO AS AN INTERNATIONAL STANDARD UNTIL PUBLISHED AS SUCH.

IN ADDITION TO THEIR EVALUATION AS BEING ACCEPTABLE FOR INDUSTRIAL, TECHNOLOGICAL, COMMERCIAL AND USER PURPOSES, DRAFT INTERNATIONAL STANDARDS MAY ON OCCASION HAVE TO BE CONSIDERED IN THE LIGHT OF THEIR POTENTIAL TO BECOME STANDARDS TO WHICH REFERENCE MAY BE MADE IN NATIONAL REGULATIONS.

RECIPIENTS OF THIS DRAFT ARE INVITED TO SUBMIT, WITH THEIR COMMENTS, NOTIFICATION OF ANY RELEVANT PATENT RIGHTS OF WHICH THEY ARE AWARE AND TO PROVIDE SUPPORTING DOCUMENTATION.

This document is circulated as received from the committee secretariat.

Reference number
ISO/IEC DIS 30115-1:2025(en)

© ISO/IEC 2025



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2025

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

CONTENTS

Foreword . . . 11
Acknowledgments . . . 11
Introduction . . . 13
1 Scope . . . 14
2 Normative references . . . 15
3 Terms, definitions, symbols, and abbreviated terms . . . 17
3.1 Hardware terms . . . 17
3.1.1 baseboard management controller (BMC) . . . 17
3.1.2 IPMI . . . 17
3.1.3 KVM-IP . . . 17
3.1.4 NIC . . . 18
3.1.5 PCI . . . 18
3.1.6 PCIe . . . 18
3.2 Web development terms . . . 18
3.2.1 CORS . . . 18
3.2.2 CRUD . . . 18
3.2.3 CSRF . . . 18
3.2.4 event . . . 18
3.2.5 excerpt . . . 18
3.2.6 HTTP . . . 18
3.2.7 HTTPS . . . 18
3.2.8 hypermedia API . . . 19
3.2.9 IP . . . 19
3.2.10 JSON . . . 19
3.2.11 member . . . 19
3.2.12 message . . . 19
3.2.13 OData . . . 19
3.2.14 OData service document . . . 19
3.2.15 operation . . . 19
3.2.16 parent resource . . . 19
3.2.17 property . . . 20
3.2.18 request . . . 20
3.2.19 response . . . 20
3.2.20 subscription . . . 20
3.2.21 task . . . 20
3.2.22 task monitor . . . 20
3.2.23 TCP . . . 20
3.2.24 TLS . . . 20
3.2.25 XSS . . . 20
3.3 Redfish terms . . . 20
3.3.1 collection . . . 20
3.3.2 Redfish client . . . 21
3.3.3 Redfish protocol . . . 21
3.3.4 Redfish schema . . . 21

© ISO/IEC 2025 – All rights reserved

3.3.5 Redfish service . . . 21
3.3.6 resource . . . 21
3.3.7 resource collection . . . 21
3.3.8 resource tree . . . 21
3.3.9 resource type . . . 21
3.3.10 service root . . . 21
3.3.11 subordinate resource . . . 22
4 Typographical conventions . . . 23
5 Overview . . . 24
5.1 Goals . . . 24
5.2 Design tenets . . . 24
5.3 Limitations . . . 25
5.4 Additional design background and rationale . . . 25
5.4.1 REST-based interface . . . 25
5.4.2 Data-oriented . . . 26
5.4.3 Separation of protocol from data model . . . 26
5.4.4 Hypermedia API service root . . . 26
5.4.5 OpenAPI v3.0 support . . . 26
5.4.6 OData conventions . . . 26
5.5 Service elements . . . 26
5.5.1 Synchronous and asynchronous operation support . . . 26
5.5.2 Eventing mechanism . . . 27
5.5.3 Actions . . . 27
5.5.4 Service discovery . . . 27
5.5.5 Remote access support . . . 27
5.6 Security . . . 28
6 Protocol details . . . 29
6.1 Universal Resource Identifiers . . . 29
6.2 HTTP methods . . . 31
6.3 HTTP redirect . . . 31
6.4 Media types . . . 31
6.5 ETags . . . 32
6.6 Protocol version . . . 33
6.7 Redfish-defined URIs and relative reference rules . . . 34
7 Service requests . . . 36
7.1 Request headers . . . 36
7.2 GET (read requests) . . . 39
7.2.1 GET (read requests) overview . . . 39
7.2.2 Resource collection requests . . . 39
7.2.3 Service root request . . . 40
7.2.4 OData service and metadata document requests . . . 40
7.3 Query parameters . . . 40
7.3.1 Query parameter overview . . . 40
7.3.2 The $expand query parameter . . . 43
7.3.3 The $select query parameter . . . 45
7.3.4 The $filter query parameter . . . 46
7.4 HEAD . . . 47

© ISO/IEC 2025 – All rights reserved

7.5 Data modification requests . . . 48
7.5.1 Data modification requests overview . . . 48
7.5.2 Modification success responses . . . 48
7.5.3 Modification error responses . . . 48
7.6 PATCH (update) . . . 49
7.7 PATCH on array properties . . . 49
7.8 PUT (replace) . . . 50
7.9 POST (create) . . . 51
7.10 DELETE (delete) . . . 51
7.11 POST (action) . . . 51
7.12 Operation apply time . . . 53
7.13 Deep operations . . . 56
8 Service responses . . . 61
8.1 Response headers . . . 61
8.2 Link header . . . 63
8.3 Status codes . . . 63
8.4 OData metadata responses . . . 66
8.4.1 OData metadata responses overview . . . 66
8.4.2 OData $metadata . . . 66
8.4.2.1 Referencing other schemas . . . 66
8.4.2.2 Referencing OEM extensions . . . 67
8.4.3 OData service document . . . 67
8.5 Resource responses . . . 68
8.6 Error responses . . . 68
9 Data model . . . 70
9.1 Resources . . . 70
9.2 Resource types . . . 70
9.3 Resource collections . . . 71
9.4 OEM resources . . . 71
9.5 Common data types . . . 71
9.5.1 Primitive types . . . 71
9.5.2 Enumerations . . . 72
9.5.3 Empty string values . . . 72
9.5.4 GUID and UUID values . . . 73
9.5.5 Date-Time values . . . 73
9.5.6 Duration values . . . 73
9.5.7 Reference properties . . . 74
9.5.8 Non-resource reference properties . . . 74
9.5.9 Array properties . . . 74
9.5.10 Structured properties . . . 75
9.5.11 Message object . . . 76
9.5.11.1 Overview . . . 76
9.5.11.2 MessageId format . . . 76
9.6 Properties . . . 77
9.6.1 Properties overview . . . 77
9.6.2 Resource identifier (@odata.id) property . . . 78
9.6.3 Resource type (@odata.type) property . . . 78

© ISO/IEC 2025 – All rights reserved

9.6.4 Resource ETag (@odata.etag) property . . . 78
9.6.5 Resource context (@odata.context) property . . . 78
9.6.6 Id . . . 79
9.6.7 Name . . . 79
9.6.8 Description . . . 79
9.6.9 MemberId . . . 79
9.6.10 Count (Members@odata.count) property . . . 79
9.6.11 Members . . . 80
9.6.12 Next link (Members@odata.nextLink) property . . . 80
9.6.13 Links . . . 80
9.6.13.1 Reference to a related resource . . . 80
9.6.13.2 References to multiple related resources . . . 81
9.6.14 Actions property . . . 81
9.6.14.1 Action representation . . . 81
9.6.14.2 Action responses . . . 82
9.6.15 Oem . . . 82
9.6.16 Status . . . 83
9.7 Naming conventions . . . 83
9.7.1 Naming rules . . . 83
9.7.2 URI naming rules . . . 84
9.8 Extending standard resources . . . 85
9.8.1 Extending standard resources overview . . . 85
9.8.2 OEM property format and content . . . 85
9.8.3 OEM-specified object naming . . . 85
9.8.4 OEM resource types . . . 86
9.8.5 OEM registries . . . 86
9.8.6 OEM URIs . . . 87
9.8.7 OEM property examples . . . 87
9.8.8 OEM actions . . . 88
9.9 Payload annotations . . . 89
9.9.1 Payload annotations overview . . . 89
9.9.2 Allowable values for strings . . . 89
9.9.3 Allowable patterns for string values . . . 90
9.9.4 Allowable values for numbers and durations . . . 90
9.9.5 Extended information . . . 91
9.9.5.1 Extended object information . . . 91
9.9.5.2 Extended property information . . . 92
9.9.6 Action info annotation . . . 93
9.9.7 Settings and settings apply time annotations . . . 93
9.9.8 Operation apply time and operation apply time support annotations . . . 94
9.9.9 Maintenance window annotation . . . 94
9.9.10 Collection capabilities annotation . . . 94
9.9.11 Requested count and allow over-provisioning annotations . . . 96
9.9.12 Zone affinity annotation . . . 96
9.9.13 Supported certificates annotation . . . 97
9.9.14 Deprecated annotation . . . 97
9.9.15 Writable properties annotation . . . 97

© ISO/IEC 2025 – All rights reserved

9.10 Settings resource . . . 98
9.11 Special resource situations . . . 100
9.11.1 Overview . . . 100
9.11.2 Absent resources . . . 100
9.12 Registries . . . 101
9.13 Schema annotations . . . 101
9.13.1 Schema annotations overview . . . 101
9.13.2 Description annotation . . . 102
9.13.3 Long description annotation . . . 102
9.13.4 Resource capabilities annotation . . . 102
9.13.5 Resource URI patterns annotation . . . 103
9.13.6 Additional properties annotation . . . 103
9.13.7 Permissions annotation . . . 103
9.13.8 Required annotation . . . 104
9.13.9 Required on create annotation . . . 104
9.13.10 Units of measure annotation . . . 104
9.13.11 Expanded resource annotation . . . 104
9.13.12 Owning entity annotation . . . 104
9.13.13 Deprecated annotation . . . 105
9.13.14 URI segment annotation . . . 105
9.13.15 URI annotation . . . 105
9.14 Versioning . . . 105
9.15 Localization . . . 106
10 File naming and publication . . . 107
10.1 Registry file naming . . . 107
10.2 Profile file naming . . . 107
10.3 Dictionary file naming . . . 107
10.4 Localized file naming . . . 107
10.5 DMTF Redfish file repository . . . 108
11 Schema definition languages . . . 109
11.1 OData Common Schema Definition Language . . . 109
11.1.1 OData Common Schema Definition Language overview . . . 109
11.1.2 File naming conventions for CSDL . . . 109
11.1.3 Core CSDL files . . . 109
11.1.4 CSDL format . . . 109
11.1.4.1 Referencing other CSDL files . . . 110
11.1.4.2 CSDL data services . . . 110
11.1.5 Elements of CSDL namespaces . . . 111
11.1.5.1 Qualified names . . . 111
11.1.5.2 Entity type and complex type elements . . . 111
11.1.5.3 Action element . . . 112
11.1.5.4 Action element for OEM actions . . . 113
11.1.5.5 Action with a response body . . . 113
11.1.5.6 Property element . . . 114
11.1.5.7 Navigation property element . . . 115
11.1.5.8 Enum type element . . . 115
11.1.5.9 Annotation element . . . 116

© ISO/IEC 2025 – All rights reserved

11.2 JSON Schema . . . 119
11.2.1 JSON Schema overview . . . 119
11.2.2 File naming conventions for JSON Schema . . . 119
11.2.3 Core JSON Schema files . . . 119
11.2.4 JSON Schema format . . . 120
11.2.5 JSON Schema definitions body . . . 120
11.2.5.1 Resource definitions in JSON Schema . . . 120
11.2.5.2 Enumerations in JSON Schema . . . 121
11.2.5.3 Actions in JSON Schema . . . 122
11.2.5.4 OEM actions in JSON Schema . . . 123
11.2.5.5 Action with a response body . . . 124
11.2.6 JSON Schema terms . . . 124
11.3 OpenAPI . . . 125
11.3.1 OpenAPI overview . . . 125
11.3.2 File naming conventions for OpenAPI schema . . . 125
11.3.3 Core OpenAPI schema files . . . 126
11.3.4 openapi.yaml . . . 126
11.3.5 OpenAPI file format . . . 128
11.3.6 OpenAPI components body . . . 128
11.3.6.1 Resource definitions in OpenAPI . . . 128
11.3.6.2 Enumerations in OpenAPI . . . 129
11.3.6.3 Actions in OpenAPI . . . 129
11.3.6.4 OEM actions in OpenAPI . . . 130
11.3.7 OpenAPI terms used by Redfish . . . 131
11.4 Schema modification rules . . . 131
12 Service details . . . 133
12.1 Eventing . . . 133
12.1.1 Eventing overview . . . 133
12.1.2 POST to subscription collection . . . 133
12.1.3 Open an SSE connection . . . 134
12.1.4 EventType-based eventing . . . 134
12.1.5 Subscribing to events . . . 135
12.1.6 Event formats . . . 136
12.1.7 OEM extensions . . . 137
12.2 Asynchronous operations . . . 137
12.3 Resource tree stability . . . 139
12.4 Discovery . . . 139
12.4.1 Discovery overview . . . 139
12.4.2 UPnP compatibility . . . 139
12.4.3 USN format . . . 140
12.4.4 M-SEARCH response . . . 140
12.4.5 Notify, alive, and shutdown messages . . . 140
12.5 Server-sent events . . . 141
12.5.1 General . . . 141
12.5.2 Event service . . . 141
12.5.2.1 Event message SSE stream . . . 143
12.5.2.2 Metric report SSE stream . . . 144

© ISO/IEC 2025 – All rights reserved

12.6 Update service . . . 145
12.6.1 Overview . . . 145
12.6.2 Software update types . . . 145
12.6.2.1 Simple updates . . . 145
12.6.2.2 Multipart HTTP push updates . . . 145
12.7 Outbound connections . . . 147
12.7.1 Overview . . . 147
12.7.2 Establishing an outbound connection . . . 147
12.7.3 MTLS for an outbound connection . . . 148
12.7.4 Handling Redfish requests over an outbound connection . . . 148
12.7.5 Closing an outbound connection . . . 149
13 Security details . . . 150
13.1 Transport Layer Security (TLS) protocol . . . 150
13.1.1 Transport Layer Security (TLS) protocol overview . . . 150
13.1.2 Cipher suites . . . 150
13.1.3 Certificates . . . 151
13.2 Sensitive data . . . 151
13.3 Authentication . . . 151
13.3.1 Authentication overview . . . 151
13.3.2 Authentication requirements . . . 151
13.3.2.1 Resource and operation authentication requirements . . . 151
13.3.2.2 HTTP header authentication requirements . . . 152
13.3.2.3 Authentication failure requirements . . . 152
13.3.3 HTTP Basic authentication . . . 152
13.3.4 Redfish session login authentication . . . 152
13.3.4.1 Redfish login sessions . . . 153
13.3.4.2 Session login . . . 153
13.3.4.3 Session lifetime . . . 154
13.3.4.4 Session termination or logout . . . 155
13.3.5 Client certificate authentication . . . 155
13.4 Authorization . . . 155
13.4.1 Authorization overview . . . 155
13.4.2 Privilege model . . . 156
13.4.2.1 Roles . . . 156
13.4.2.2 Restricted roles and restricted privileges . . . 157
13.4.2.3 OEM privileges . . . 158
13.4.3 Redfish service operation-to-privilege mapping . . . 158
13.4.3.1 Why specify operation-to-privilege mapping? . . . 158
13.4.3.2 Representing operation-to-privilege mappings . . . 158
13.4.3.3 Operation map syntax . . . 159
13.4.3.4 Mapping overrides syntax . . . 159
13.4.3.5 Property override example . . . 160
13.4.3.6 Subordinate override . . . 161
13.4.3.7 Resource URI override . . . 162
13.4.3.8 Privilege AND and OR syntax . . . 163
13.4.4 Delegated authorization with OAuth 2.0 . . . 163
13.4.4.1 OAuth 2.0 overview . . . 163

© ISO/IEC 2025 – All rights reserved

13.4.4.2 OAuth 2.0 data model requirements . . . 164
13.4.4.3 OAuth 2.0 access tokens . . . 164
13.4.4.4 Redfish OAuth2.0 scope usage . . . 166
13.5 Account service . . . 166
13.5.1 Account service overview . . . 166
13.5.2 Password management . . . 166
13.5.3 Atomic password changes . . . 166
13.5.4 Password change required handling . . . 167
13.6 Asynchronous tasks . . . 167
13.7 Event subscriptions . . . 167
14 Redfish Host Interface . . . 169
15 Redfish composability . . . 170
15.1 Composition requests . . . 170
15.1.1 Composition requests overview . . . 170
15.1.2 Specific composition . . . 171
15.1.3 Constrained composition . . . 171
15.1.4 Expandable resources . . . 172
15.2 Updating a composed resource . . . 173
16 Aggregation . . . 174
16.1 Classes of aggregators . . . 174
16.1.1 Implicit and complex aggregators . . . 174
16.1.2 Use cases . . . 174
16.2 Aggregation service . . . 175
16.2.1 Aggregation service overview . . . 175
16.2.2 Aggregator requirements . . . 175
16.2.3 Aggregates . . . 176
16.2.4 Aggregation sources and connection methods . . . 176
17 ANNEX A (informative) Change log . . . 177
18 Bibliography . . . 195

© ISO/IEC 2025 – All rights reserved

Foreword

DMTF's Redfish Forum develops the Redfish standard.

DMTF is a not-for-profit association of industry members that promotes enterprise and systems management and interoperability. For information about DMTF, see DMTF.

This version supersedes version 1.19.1. For a list of the changes, see ANNEX A (informative) Change log.

Acknowledgments

DMTF acknowledges the following individuals for their contributions to the Redfish standard, including this document and Redfish schemas, interoperability profiles, and message registries:

- Rafiq Ahamed — Hewlett Packard Enterprise
- Richelle Ahlvers — Intel Corporation, Broadcom Inc.
- Jeff Autor — Hewlett Packard Enterprise
- David Black — Dell Technologies
- Jeff Bobzin — Insyde Software Corp.
- Patrick Boyd — Dell Technologies
- David Brockhaus — Vertiv
- Richard Brunner — VMware Inc.
- Sean Byland — Hewlett Packard Enterprise
- Lee Calcote — Seagate Technology
- Keith Campbell — Lenovo
- Derek Chan — Google LLC
- P Chandrasekhar — Dell Technologies
- Barbara Craig — Hewlett Packard Enterprise
- Chris Davenport — Hewlett Packard Enterprise
- Gamma Dean — Vertiv
- Michael Du — Huawei Technologies Co., Ltd.
- Daniel Dufresne — Dell Technologies
- Samer El-Haj-Mahmoud — Arm Limited, Lenovo, Hewlett Packard Enterprise
- George Ericson — Dell Technologies
- Wassim Fayed — Microsoft Corporation
- Kevin Ferguson — Vertiv
- Mike Garrett — Hewlett Packard Enterprise
- Steve Geffin — Vertiv
- Martin Halstead — Hewlett Packard Enterprise
- Joe Handzik — Hewlett Packard Enterprise
- Jon Hass — Dell Technologies
- Jeff Hilland — Hewlett Packard Enterprise
- Blake Hilliard — Hewlett Packard Enterprise

© ISO/IEC 2025 – All rights reserved

- Chris Hoffman — Vertiv
- Cactus Jiang — Vertiv
- Barry Kittner — Intel Corporation
- Steven Krig — Intel Corporation
- Jennifer Lee — Intel Corporation
- John Leung — Intel Corporation
- Magnus Lundmark — Ericsson AB
- Steve Lyle — Hewlett Packard Enterprise
- Gunnar Mills — IBM
- Jagan Molleti — Dell Technologies
- Milena Natanov — Microsoft Corporation
- Balaji Natrajan — Microchip Technology Inc., Hewlett Packard Enterprise
- Scott Phuong — Cisco Systems, Inc.
- Michael Pizzo — Microsoft Corporation
- Chris Poblete — Dell Technologies
- Slawek Putyrski — Intel Corporation
- Michael Raineri — Dell Technologies
- Joseph Reynolds — IBM
- Irina Salvan — Microsoft Corporation
- Joseph-Jonathan Salzano — HP, Inc
- Bill Scherer — Hewlett Packard Enterprise
- Geoff Schunicht — Hewlett Packard Enterprise
- Abhirup Seal — Dell Technologies
- Hemal Shah — Broadcom Inc.
- Jim Shelton — Vertiv
- Tom Slaight — Intel Corporation
- Josiah Smith — Eaton
- Donnie Sturgeon — Vertiv
- Pawel Szymanski — Intel Corporation
- Ed Tanous — Google LLC
- Willy Tu — Google LLC
- Paul Vancil — Dell Technologies
- Ganesh Viswanathan — Dell Technologies
- Claire Weinan — Google LLC
- Joseph White — Dell Technologies
- Linda Wu — NVIDIA Corporation, Super Micro Computer, Inc.
- Justin York — Hewlett Packard Enterprise

© ISO/IEC 2025 – All rights reserved

Introduction

Redfish is a standard that uses RESTful interface semantics to access a schema-based data model to conduct management operations. It is suitable for a wide range of devices, from stand-alone servers to composable infrastructures and large-scale cloud environments.

The initial Redfish scope targeted servers. DMTF and its alliance partners expanded that scope to cover most data center IT equipment and other solutions, and both in- and out-of-band access methods.

Additionally, DMTF and other organizations that use Redfish as part of their industry standard or solution have added educational material.

This document defines the RESTful interface protocol and the various concepts and services necessary to implement a Redfish interface. The definition of the schema based data model and standard messages for the Redfish interface are covered separately in the following documents:

- DMTF DSP8010, *Redfish Schema Bundle*, https://www.dmtf.org/dsp/DSP8010 contains the individual schema definition files in multiple schema description languages.
- DMTF DSP0268, *Redfish Data Model Specification*, https://www.dmtf.org/dsp/DSP0268 contains the normative descriptions and example payloads for all standard Redfish schema in a single reference guide.
- DMTF DSP8011, *Redfish Standard Registries Bundle*, https://www.dmtf.org/dsp/DSP8011 contains the message registries used for error reporting and event messages.

© ISO/IEC 2025 – All rights reserved

1 Scope

This specification defines the required protocols, data model, behaviors, and other architectural components for an interoperable, multivendor, remote, and out-of-band capable interface. This interface meets the cloud-based and web-based IT professionals' expectations for scalable platform management. While large and hyperscale environments are the primary focus, clients can use the specification for individual system management.

The specification defines the required elements for all Redfish implementations, and the optional elements that system vendors and manufacturers can choose. This specification also defines at which points an implementation can provide OEM-specific extensions.

The specification sets normative requirements for *Redfish services* and associated materials, such as Redfish schema files. In general, the specification does not set requirements for Redfish clients but indicates how a client can successfully and effectively access and use a Redfish service.

The specification does not require that implementations of the Redfish interfaces and functions require particular hardware or firmware.

© ISO/IEC 2025 – All rights reserved

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

- DMTF DSP0270, *Redfish Host Interface Specification*, https://www.dmtf.org/sites/default/files/standards/documents/DSP0270_1.0.0.pdf
- Redfish Schema: RedfishExtensions v1.0.0, https://redfish.dmtf.org/schemas/v1/RedfishExtensions_v1.xml
- *Transport Layer Security (TLS) Parameters*, https://www.iana.org/assignments/tls-parameters/tls-parameters.xhtml
- *JSON Schema: A Media Type for Describing JSON Documents draft-handrews-json-schema-01*, https://tools.ietf.org/html/draft-handrews-json-schema-01
- *JSON Schema Validation: A Vocabulary for Structural Validation of JSON draft-handrews-json-schema-validation-01*, https://tools.ietf.org/html/draft-handrews-json-schema-validation-01
- IETF RFC1738, T. Berners-Lee et al., *Uniform Resource Locators (URL)*, https://tools.ietf.org/html/rfc1738
- IETF RFC3986, T. Berners-Lee et al., *Uniform Resource Identifier (URI): Generic Syntax*, https://tools.ietf.org/html/rfc3986
- IETF RFC4122, P. Leach et al., *A Universally Unique IDentifier (UUID) URN Namespace*, https://tools.ietf.org/html/rfc4122
- IETF RFC5280, D. Cooper et al., *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*,
- IETF RFC6455, I. Fette et al., *The WebSocket Protocol*, https://tools.ietf.org/html/rfc6585
- IETF RFC6585, M. Nottingham et al., *Additional HTTP Status Codes*, https://tools.ietf.org/html/rfc6585
- IETF RFC6749, D. Hardt, Ed., *The OAuth 2.0 Authorization Framework*, https://tools.ietf.org/html/rfc6749
- IETF RFC6901, P. Bryan, Ed. et al., *JavaScript Object Notation (JSON) Pointer*, https://tools.ietf.org/html/rfc6901
- IETF RFC7230, R. Fielding et al., *Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing*, https://tools.ietf.org/html/rfc7230
- IETF RFC7231, R. Fielding et al., *Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content*, https://tools.ietf.org/html/rfc7231
- IETF RFC7232, R. Fielding et al., *Hypertext Transfer Protocol (HTTP/1.1): Conditional Requests*, https://tools.ietf.org/html/rfc7232
- IETF RFC7234, R. Fielding et al., *Hypertext Transfer Protocol (HTTP/1.1): Caching*, https://tools.ietf.org/html/rfc7234
- IETF RFC7540, M. Belshe et al., *Hypertext Transfer Protocol Version 2 (HTTP/2)*, https://tools.ietf.org/html/rfc7540
- IETF RFC7519, M. Jones et al., *JSON Web Token (JWT)*, https://tools.ietf.org/html/rfc7519
- IETF RFC7525, Y. Sheffer et al., *Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)*, https://tools.ietf.org/html/rfc7525
- IETF RFC7578, L. Masinter et al., *Returning Values from Forms: multipart/form-data*, https://tools.ietf.org/html/rfc7578
- IETF RFC7617, J. Reschke et al., *The 'Basic' HTTP Authentication Scheme*, https://tools.ietf.org/html/rfc7617

© ISO/IEC 2025 – All rights reserved

- IETF RFC8259, T. Bray, Ed., *The JavaScript Object Notation (JSON) Data Interchange Format*, https://tools.ietf.org/html/rfc7617
- IETF RFC8288, M. Nottingham, *Web Linking*, https://tools.ietf.org/html/rfc8288
- ISO 639-1:2002, *Codes for the representation of names of languages - Part 1: Alpha-2 code*, https://www.iso.org/standard/22109.html
- 24 February 2014, *OData Version 4.0 Part 1: Protocol*, https://docs.oasis-open.org/odata/odata/v4.0/os/part1-protocol/odata-v4.0-os-part1-protocol.html
- 24 February 2014, *OData Version 4.0 Part 3: Common Schema Definition Language (CSDL)*, https://docs.oasis-open.org/odata/odata/v4.0/os/part3-csdl/odata-v4.0-os-part3-csdl.html
- 10 March 2016, *OData Version 4.0 Plus Errata 03 OASIS Standard incorporating Draft 01 of Errata 03*, https://docs.oasis-open.org/odata/odata/v4.0/errata03/csd01/complete/vocabularies/Org.OData.Measures.V1.xml
- 20 November 2014, SNIA *TLS Specification for Storage Systems*, https://www.snia.org/tech_activities/standards/curr_standards/tls
- *The OpenAPI Specification*, https://swagger.io/specification/
- *The Unified Code for Units of Measure*, https://ucum.org/ucum.html
- 9 September 2021, *Fetch Living Standard*, https://fetch.spec.whatwg.org/
- 17 September 2021, *9.2 Server-sent events* in the *HTML Living Standard*, https://html.spec.whatwg.org/multipage/server-sent-events.html

© ISO/IEC 2025 – All rights reserved

3 Terms, definitions, symbols, and abbreviated terms

Some terms and phrases in this document have specific meanings beyond their typical English meanings. This clause defines those terms and phrases.

The terms “shall” (“required”), “shall not”, “should” (“recommended”), “should not” (“not recommended”), “may”, “need not” (“not required”), “can” and “cannot” in this document are to be interpreted as described in ISO/IEC Directives, Part 2, Clause 7. The terms in parenthesis are alternatives for the preceding term, for use in exceptional cases when the preceding term cannot be used for linguistic reasons. Note that ISO/IEC Directives, Part 2, Clause 7 specifies additional alternatives. Occurrences of such additional alternatives shall be interpreted in their normal English meaning.

The terms “clause”, “subclause”, “paragraph”, and “annex” in this document are to be interpreted as described in ISO/IEC Directives, Part 2, Clause 6.

The terms “normative” and “informative” in this document are to be interpreted as described in ISO/IEC Directives, Part 2, Clause 3. In this document, clauses, subclauses, or annexes labeled “(informative)” do not contain normative content. Notes and examples are always informative elements.

The term “deprecated” in this document is to be interpreted as material that is not recommended for use in new development efforts. Existing and new implementations may use this material, but they should move to the favored approach. Deprecated material may be implemented in order to achieve backwards compatibility. Deprecated material should contain references to the last published version that included the deprecated material as normative material and to a description of the favored approach. Deprecated material may be removed from the next major version of the specification.

This document defines these additional terms:

3.1 Hardware terms

3.1.1 baseboard management controller (BMC)

embedded device or service

Note 1 to entry: Typically, an independent microprocessor or system-on-chip with associated firmware in a computer system that completes out-of-band systems monitoring and management-related tasks.

3.1.2 IPMI

Intelligent Platform Management Interface

3.1.3 KVM-IP

keyboard, video, mouse redirection over IP

© ISO/IEC 2025 – All rights reserved

3.1.4 NIC

network interface controller

3.1.5 PCI

Peripheral Component Interconnect

3.1.6 PCIe

Peripheral Component Interconnect Express

3.2 Web development terms

3.2.1 CORS

cross-origin resource sharing

3.2.2 CRUD

basic **C**reate, **R**ead, **U**pdate, and **D**elete operations that any interface can support

3.2.3 CSRF

cross-site request forgery

3.2.4 event

data structure that corresponds to one or more alerts

3.2.5 excerpt

subset of data that is copied from one resource and presented in another resource

Note 1 to entry: An excerpt provides data in convenient locations without duplication of entire resources.

3.2.6 HTTP

Hypertext Transfer Protocol

3.2.7 HTTPS

Hypertext Transfer Protocol Secure

© ISO/IEC 2025 – All rights reserved

Note 1 to entry: TLS secures HTTP.

3.2.8 hypermedia API

API that enables you to navigate through URIs that a service returns

3.2.9 IP

Internet Protocol

3.2.10 JSON

JavaScript Object Notation

3.2.11 member

single resource instance in a resource collection

3.2.12 message

complete HTTP-formatted or HTTPS-formatted request or response

Note 1 to entry: In the REST-based Redfish protocol, every request results in a response.

3.2.13 OData

Open Data Protocol (OData), as defined in OData Version 4.0 Part 1: Protocol

3.2.14 OData service document

resource that provides information about the service root for generic OData clients

3.2.15 operation

HTTP `POST`, `GET`, `PUT`, `PATCH`, `HEAD`, and `DELETE` request methods that map to generic CRUD operations

3.2.16 parent resource

parent to another resource if the initial segment of the resource URI is the same as the URI of the other resource, but is at least one level higher

Note 1 to entry: For example, `/redfish/v1/Chassis/A88` is a parent resource of `/redfish/v1/Chassis/A88/Assembly`.

© ISO/IEC 2025 – All rights reserved

3.2.17 property

name-value pair in a Redfish-defined request or response

Note 1 to entry: A property can be any valid JSON data type.

3.2.18 request

message from a client to a service

3.2.19 response

message from a service to a client in response to a request message

3.2.20 subscription

registration of a destination to receive events

3.2.21 task

representation of a long-running operation

3.2.22 task monitor

opaque service-generated URI that the client that initiates the request can use to monitor an asynchronous operation

3.2.23 TCP

Transmission Control Protocol

3.2.24 TLS

Transport Layer Security

3.2.25 XSS

cross-site scripting

3.3 Redfish terms

3.3.1 collection

see resource collection

© ISO/IEC 2025 – All rights reserved

3.3.2 Redfish client

communicates with a Redfish service and accesses one or more of the service's resources or functions

3.3.3 Redfish protocol

discovers, connects to, and inter-communicates with a Redfish service

3.3.4 Redfish schema

a set of human and machine-readable documents that define Redfish resources using one or more of the supported schema definition languages

3.3.5 Redfish service

implementation of the protocols, resources, and functions that deliver the interface that this specification defines and its associated behaviors for one or more managed systems

Note 1 to entry: Also known as the *service*.

3.3.6 resource

URI-addressable Redfish data structure

3.3.7 resource collection

set of similar resources where the number of instances can shrink or grow

3.3.8 resource tree

tree structure of resources accessible through a well-known starting URI

Note 1 to entry: A client can discover the available resources on a Redfish service by following the resource hyperlinks from the base of the tree.

3.3.9 resource type

set of definitions for properties and actions contained within a resource and documented in the Redfish schema files

3.3.10 service root

starting-point resource for locating and accessing the other resources and associated metadata that make up an instance of a Redfish service

© ISO/IEC 2025 – All rights reserved

3.3.11 subordinate resource

is subordinate to another resource if the initial segment of the resource URI is the same as the URI of the other resource, but is at least one level deeper

Note 1 to entry: For example, `/redfish/v1/Chassis/A88/Assembly` is a subordinate resource of the `Chassis` resource named `A88`.

© ISO/IEC 2025 – All rights reserved

4 Typographical conventions

The following typographical convention indicates deprecated material:

DEPRECATED

Deprecated material appears here.

END DEPRECATED

In places where this typographical convention cannot be used, such as tables or figures, the “DEPRECATED” label is used alone.

© ISO/IEC 2025 – All rights reserved

5 Overview

Redfish is a management standard that uses a data model representation with a RESTful interface.

Being RESTful, Redfish is easier to use and implement.

Being model-oriented, it can express the relationships between components and the semantics of the Redfish services and components within them. The model is also easy to extend.

By requiring JSON representation, Redfish enables easy integration with programming environments. It is also easy to interpret by humans.

An interoperable Redfish schema defines this model, which is freely available and published in OpenAPI YAML, OData CSDL, and JSON Schema formats.

5.1 Goals

As an architecture, data model, and set of protocols that enable a client to access Redfish services, Redfish has these goals.

Table 1 describes these goals:

Table 1 — Redfish goals

Goal	Purpose
Scalable	Can scale on stand-alone machines or racks of equipment.
Flexible	Can implement through existing hardware or entirely as a software service.
Extensible	Can easily add new and vendor-specific capabilities to the data model.
Backward-compatible	Can add capabilities while preserving investments in earlier implementations.
Interoperable	Provides consistent functionality across multiple vendor implementations.
Standards-based	Built on ubiquitous and secure protocols. Leverages other standards where applicable.
Simple	Easy-to-use without the need for highly specialized programming skills or systems knowledge.
Lightweight	Designed to reduce complexity and implementation costs. Minimizes the required footprint for implementations.

5.2 Design tenets

To deliver these goals, Redfish:

- Provides a RESTful interface by using a JSON payload and a data model.
- Separates the protocol from the data model, which enables the independent revision and use of each.
- Specifies versioning rules for protocols and schema.
- Leverages strength of ubiquitous standards where it meets architectural requirements, such as JSON, HTTP,

© ISO/IEC 2025 – All rights reserved