Document Identifier: DSP2043

Date: 2021-01-18

Version: 2020.4

Redfish Mockup Readme

Supersedes: None

Document Class: Informational

Document Status: Published

Document Language: en-US

16 DMTF is a not-for-profit association of industry members dedicated to promoting enterprise and systems management and interoperability. Members and non-members may reproduce DMTF specifications and documents, provided that correct attribution is given. As DMTF specifications may be revised from time to time, the particular version and release date should always be noted.

17 Implementation of certain elements of this standard or proposed standard may be subject to third party patent rights, including provisional patent rights (herein "patent rights"). DMTF makes no representations to users of the standard as to the existence of such rights, and is not responsible to recognize, disclose, or identify any or all such third party patent right, owners or claimants, nor for any incomplete or inaccurate identification or disclosure of such rights, owners or claimants. DMTF shall have no liability to any party, in any manner or circumstance, under any legal theory whatsoever, for failure to recognize, disclose, or identify any such third party patent rights, or for such party's reliance on the standard or incorporation thereof in its product, protocols or testing procedures. DMTF shall have no liability to any party implementing such standard, whether such implementation is foreseeable or not, nor to any patent owner or claimant, and shall have no liability or responsibility for costs or losses incurred if a standard is withdrawn or modified after publication, and shall be indemnified and held harmless by any party implementing the standard from any and all claims of infringement by a patent owner for such implementations.

18 For information about patents held by third-parties which have notified the DMTF that, in their opinion, such patent may relate to or impact implementations of DMTF standards, visit http://www.dmtf.org/about/policies/disclosures.php.

19 This document's normative language is English. Translation into other languages is permitted.

CONTENTS

21 1 Foreword

22 The following files are part of the Redfish development effort:

- DSP0266 - Redfish Specification - This file is the main Redfish Specification.
- DSP0270 - Redfish Host Interface Specification - This document specifies the "in-band" or "OS-based" Redfish Host Interface.
- DSP0272 - Redfish Interoperability Profile Specification - Specifies the structure and JSON document used to define and publish an interoperability profile used to check an implementation's conformance to a defined minimum set of functionality.
- DSP2044 - Redfish White Paper - This is intended to be a non-normative document helping those new to Redfish understand how to interact with the Redfish Service and understand common functions and tasks.
- DSP2043 - Redfish Mockup - This is a mockup that can be used as sample of output from GETs from a Redfish Service. Informative in nature, it was used to develop the schema. A person can set up an NGINX or similar server and configure it to output JSON format and then use this directory for demonstration purposes.
- DSP2046 - Redfish Resource and Schema Guide - This contains informative documentation regarding common Redfish Resource properties, as well as a listing of properties that can be found in each of the Redfish Resources.
- DSP8010 - Redfish Schema - This contains the Redfish Schema definitions. These files are normative in nature and are normatively referenced by the Redfish Specification. There are two Schema formats - CSDL (OData Common Schema Definition Language format, which is in XML) and JSON Schema. These Schema definitions should be functionally equivalent, thus specifying the schema in two different languages.
- DSP8011 - Redfish Standard Registries - This contains the Redfish Registry definitions. This bundle of Redfish registries includes Message registries used for Redfish-defined messages (including events) and Privilege maps.
- DSP8013 - Redfish Interoperability Profiles - A bundle of published Redfish Interoperability Profile documents as well as supporting schema and sample documents used for creating profiles.

23 2 Redfish Mockups

24 This archive contains a number of mockups of various Redfish service implementations. They are intended to be a guide for learning about the Redfish Specification by showing typical examples of implementations. These mockups are not prototypes and do not reflect any actual product or Redfish implementation.

25 Many of these mockups are also used to populate the Redfish Resource Explorer, part of the Redfish Developer Hub located at: http://redfish.dmtf.org

26 2.1 Simple Rack-mounted Server (public-rackmount1)

27 This illustration of a Redfish service implementation shows a typical rack-mount server, as commonly used in scale-out data centers. It depicts the types of information that can be expected, but does not represent an actual implementation.

28 2.2 Bladed System (public-bladed)

29 This example represents an enclosure of "blade servers" that share infrastructure components, such as power supplies and fans. Depicting an enclosure containing four blade servers (a total of five "Chassis"), this mockup demonstrates the modeling of multiple chassis and systems managed from a single Redfish service.

30 2.3 Simple Rack-mounted Server with Local Storage (public-localstorage)

31 This example shows a server with an implementation of the Redfish storage resources, showing an integrated RAID controller with four attached drives.

32 2.4 SAS Fabric (public-sasfabric)

33 This example shows a more complex storage implementation using a pair of SAS switches, storage enclosures and multiple storage devices.

34 2.5 Proposed OCP Redfish Profile (public-catfish)

35 This draft example, for ongoing development, represents a proposed minimal Redfish data model "profile" that meets the needs of the Open Compute Project's Hardware Management requirements. This draft profile is intended to help define a list of required properties so that essential management-related tasks, as defined by OCP, can be performed on any Redfish implementation.

36 2.6 Composable System via Specific Compositions (public-composability)

37 This example shows a service with various sets of disaggregated hardware as resources. It provides an example two composed systems utilizing some of the disaggregated hardware. It also shows how Resource Zones can provide information about binding restrictions. It also shows how to express composition requests using the specific composition format.

38 2.7 Bladed Partitions (public-bladed-partitions)

39 This example shows how Redfish Composability can be used to create composed Computer System instances from smaller sets of Computer Systems. A top level enclosure called "Enclosure" contains a set of blades, which are used to create the composed Computer Systems.

40 2.8 Composable System via Constrained Compositions (public-constrained-composition)

41 This example shows a service with various sets of disaggregated hardware as resources. It provides an example two composed systems utilizing some of the disaggregated hardware. It also shows how Resource Zones can provide information about binding restrictions. It also shows how to express composition requests using the constrained composition format.

42 2.9 Expansion Enclosure with Resource Blocks (public-expansion-box)

43 This example shows a service with various sets of disaggregated hardware as resources. The service itself provides information about the types of hardware available in the enclosure, but provides no composability functionality. In these circumstances, an external Redfish service might be used to orchestrate how the equipment is provisioned for composability.

44 2.10 Telemetry Service (public-telemetry)

45 This example shows a service that supports reporting telemetry data through the Telemetry Service. It has sample metric definitions and metric reports based upon data found in other portions of the data model.

46 2.11 Infrastructure Sample (public-infrastructure)

47 Sample representation of DCIM equipment.

48 2.12 Sample Service with OEM Extensions (public-oem-examples)

49 A lightweight Redfish service with OEM examples. The Service Root resource has been extended to have an OEM section. The service container also has an extension. The Account Service resource contains and OEM action. These OEM extensions are defined in the Contoso.com folder of the mockup.

50 2.13 Mockup for PMEM technologies (public-pmem-configuration)

51 A mockup of initial configurations for reporting PMEM Devices and then the configurations after a sequence of configuration requests.

52 2.14 NVMe-oF JBOF (public-nvmeof-jbof)

53 This mockup contains a sample Redfish service for an NVMe-oF JBOF. The storage resources off of service root contain the provisionable storage for external hosts. The fabric portion of the data model is used to express host connectivity to the different NVMe namespaces. It also contains a single SoC, represented as a computer system, that is used as the front-end for receiving NVMe-oF traffic before the drives are accessed.