Redfish Conformance and Test Tools White Paper

Supersedes: None
Document Class: Informational
Document Status: Published
Document Language: en-US
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.

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patent owner or claimant, and shall have no liability or responsibility for costs or losses incurred if a standard is withdrawn or modified after publication, and shall be indemnified and held harmless by any party implementing the standard from any and all claims of infringement by a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.

For information about patents held by third-parties which have notified 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.

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

Foreword ................................................................. 4
Acknowledgments ....................................................... 4
1 Introduction .......................................................... 5
2 Installing DMTF's conformance tools .............................. 6
3 Redfish Protocol Validator ........................................... 7
4 Redfish Service Validator ............................................ 10
5 Redfish Interop Validator ............................................. 13
6 Appendix A: References ............................................. 16
7 Appendix B: Change log .............................................. 17
Foreword

The Redfish Conformance and Test Tools White Paper was prepared by DMTF’s Redfish Forum.

DMTF is a not-for-profit association of industry members dedicated to promoting enterprise and systems management and interoperability. For information about DMTF, see http://www.dmtf.org.

Acknowledgments

DMTF acknowledges the following individuals for their contributions to this document:

• Michael Raineri — Dell Technologies
1 Introduction

DMTF's Redfish Forum produces specifications that contain normative requirements expected by all Redfish services. Redfish service designers are expected to implement their services according to these requirements. Redfish clients rely on the support of these requirements to ensure that they can communicate properly with Redfish services and consume their data.

DMTF's Redfish Forum maintains several open-source Redfish conformance tools to verify Redfish services are conformant with DMTF specifications and third-party prescriptions. Service developers should use the tools in this white paper to verify their Redfish implementation.
2 Installing DMTF's conformance tools

All Redfish conformance tools maintained by DMTF are written for Python3. These tools are also maintained in GitHub, but also have releases pushed to the Python Package Index (PyPI). This allows users to use `pip` or other built-in Python functions to install these tools without downloading source code manually.

The following commands can be used to install the tools documented in this white paper:

```
pip install redfish_protocol_validator -U
pip install redfish_service_validator -U
pip install redfish_interop_validator -U
```

New versions of the tools in this white paper are released periodically to address bugs reported by users or add new features. The previous commands can also be used to download and apply updates to the tools.
3 Redfish Protocol Validator

The Redfish Protocol Validator tests a service for the protocol requirements defined in the Redfish Specification. This includes tests that verify:

- HTTP request and response headers
- HTTP status codes
- HTTP methods
- Request and response body encoding
- Security requirements

The following command can be used to run the Redfish Protocol Validator:

```
rf_protocol_validator -u <USERNAME> -p <PASSWORD> -r https://<IP> --no-cert-check
```

where

- `<USERNAME>` is the username of the Redfish account to use during testing
- `<PASSWORD>` is the password of the Redfish account to use during testing
- `<IP>` is the IP address of the Redfish service to test

When the test complete, an HTML report will be produced and saved in the `results` subdirectory from where the tool was invoked. The following figures show example portions of the HTML report. This first figure shows the report heading, which contains information such as the tool version, the system under test, and a summary of the results. The second figure shows an example portion of the report with results for specific tests.
Figure 1
<table>
<thead>
<tr>
<th>Result</th>
<th>Method</th>
<th>Status code</th>
<th>URI</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>PASS</td>
<td>GET</td>
<td>200</td>
<td>/redfish</td>
<td>Test passed</td>
</tr>
<tr>
<td>PASS</td>
<td>GET</td>
<td>200</td>
<td>/redfish/v1/odata</td>
<td>Test passed</td>
</tr>
<tr>
<td>PASS</td>
<td>GET</td>
<td>200</td>
<td>/redfish/v1/$metadata</td>
<td>Test passed</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Result</th>
<th>Method</th>
<th>Status code</th>
<th>URI</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>PASS</td>
<td>GET</td>
<td>200</td>
<td>/redfish/v1/</td>
<td>Test passed</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Result</th>
<th>Method</th>
<th>Status code</th>
<th>URI</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>PASS</td>
<td>GET</td>
<td>200</td>
<td>/redfish/v1/</td>
<td>Test passed</td>
</tr>
</tbody>
</table>

PROTO STD URI SERVICE_ROOT: "The root URI for this version of the Redfish protocol shall be /redfish/v1/.

<table>
<thead>
<tr>
<th>Result</th>
<th>Method</th>
<th>Status code</th>
<th>URI</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>PASS</td>
<td>GET</td>
<td>200</td>
<td>/redfish/v1/</td>
<td>Test passed</td>
</tr>
</tbody>
</table>

PROTO STD URI SERVICE_ROOT REDIRECT: "The service shall process the [/redfish/v1] URI without a trailing slash in one of these ways: Redirect it to the associated Redfish-defined URI, or treat it as the equivalent URI to the associated Redfish-defined URI (/redfish/v1/)."

<table>
<thead>
<tr>
<th>Result</th>
<th>Method</th>
<th>Status code</th>
<th>URI</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>PASS</td>
<td>GET</td>
<td>200</td>
<td>/redfish/v1/</td>
<td>Test passed</td>
</tr>
</tbody>
</table>

PROTO STD URI VERSION: "A GET operation on the /redfish resource shall return this response body: {'v1': '/redfish/v1/'}"

<table>
<thead>
<tr>
<th>Result</th>
<th>Method</th>
<th>Status code</th>
<th>URI</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>PASS</td>
<td>GET</td>
<td>200</td>
<td>/redfish</td>
<td>Test passed</td>
</tr>
</tbody>
</table>

Figure 2
4 Redfish Service Validator

The Redfish Service Validator tests the response payloads from a service conform to the schema definitions in the Redfish Schema Bundle. This includes tests that verify:

- Mandatory properties are present
- The data type of each property is correct
- There are no undefined properties
- URIs for each resource match expected URI patterns

The following command can be used to run the Redfish Service Validator:

```
rf_service_validator -u <USERNAME> -p <PASSWORD> -r https://<IP>
```

where

- `<USERNAME>` is the username of the Redfish account to use during testing
- `<PASSWORD>` is the password of the Redfish account to use during testing
- `<IP>` is the IP address of the Redfish service to test

When the test complete, an HTML report will be produced and saved in the `logs` subdirectory from where the tool was invoked. The following figures show example portions of the HTML report. This first figure shows the report heading, which contains information such as the tool version, the system under test, and a summary of the results. The second figure shows an example portion of the report with results for specific tests.
### Redfish Conformance Test Report

[Redfish Logo]

[GitHub Link]

**Tool Version:** 2.3.7  
**Fri Nov 3 09:30:25 2023**  
**(Run time: 0:00:09)**

This tool is provided and maintained by the DMTF. For feedback, please open issues in the tool's Github repository: [GitHub Issues Link]

<table>
<thead>
<tr>
<th>Description</th>
<th>System: http://</th>
</tr>
</thead>
<tbody>
<tr>
<td>badNamespaceInclude: 1</td>
<td>err.Edm.DateTimeOffset: 3</td>
</tr>
<tr>
<td>err.Edm.String: 1</td>
<td>err.EventDestination.v1_3_0.SubscriptionType: 3</td>
</tr>
<tr>
<td>err.Outlet.Outlet: 1</td>
<td>err.PCIeDevice.PCIeDevice: 1</td>
</tr>
<tr>
<td>errorMissingOdataId: 2</td>
<td>errorMissingRefOdata: 1</td>
</tr>
<tr>
<td>failAdditional.complex: 7</td>
<td>failGet: 1</td>
</tr>
<tr>
<td>failMandatoryExist: 15</td>
<td>failProp: 5</td>
</tr>
</tbody>
</table>

---

Figure 3
### Figure 4

![Certificate Details](image)

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Value</th>
<th>Type</th>
<th>Exists</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>@odata.id</td>
<td><code>/redfish/v1/Systems/437XR1138R2/Certificates/contoso-subca</code></td>
<td>odata</td>
<td>Exists</td>
<td>FAIL</td>
</tr>
<tr>
<td>@odata.type</td>
<td>Certificate.v1_6_0.Certificate</td>
<td>odata</td>
<td>Exists</td>
<td>PASS</td>
</tr>
<tr>
<td>SPDM</td>
<td>-</td>
<td>Certificate.v1_5_0.SPD</td>
<td>No</td>
<td>Optional</td>
</tr>
<tr>
<td>Links</td>
<td>[JSON Object]</td>
<td>Certificate.v1_4_0.Links</td>
<td>Yes</td>
<td>complex</td>
</tr>
<tr>
<td>Links.Issuer</td>
<td>Link: <code>/redfish/v1/Systems/437XR1138R2/Certificates/contoso-root</code></td>
<td>link to: Certificate</td>
<td>Yes</td>
<td>PASS</td>
</tr>
<tr>
<td>Links.Subjects</td>
<td>Array (size: 3)</td>
<td>links: Certificate</td>
<td>Yes</td>
<td>...</td>
</tr>
<tr>
<td>Links.Subjects[0]</td>
<td>Link: <code>/redfish/v1/Chassis/1U/TrustedComponents/AC-ROTo/Certificates/DeviceCert</code></td>
<td>link: Certificate</td>
<td>Yes</td>
<td>PASS</td>
</tr>
</tbody>
</table>
5 Redfish Interop Validator

The Redfish Interop Validator tests a service to verify conformance to a Redfish interoperability profile. Redfish interoperability profiles are created by organizations, customers, or other third parties to ensure a minimum set of the Redfish data model is supported.

Some example Redfish interoperability profiles include:

- **OCP Hardware Management Profiles**: OCP has several profiles that contain hardware management requirements for various classes of products.
- **OPAF Profiles**: TOG/OPAF has profiles for different tiers of systems for integration into OPAF-defined process automation environments.
- **OpenStack Ironic Profiles**: OpenStack has a profile that contains requirements for a Redfish service to integrate with its bare metal provisioning software, Ironic.

Redfish interoperability profiles are defined by the *Redfish Interoperability Profiles Specification*.

The following command can be used to run the Redfish Interop Validator:

```
rf_interop_validator -u <USERNAME> -p <PASSWORD> -r https://<IP> <PROFILE>
```

where

- `<USERNAME>` is the username of the Redfish account to use during testing
- `<PASSWORD>` is the password of the Redfish account to use during testing
- `<IP>` is the IP address of the Redfish service to test
- `<PROFILE>` is the filepath to the profile to test against

When the test complete, an HTML report will be produced and saved in the `logs` subdirectory from where the tool was invoked. The following figures show example portions of the HTML report. This first figure shows the report heading, which contains information such as the tool version, the system under test, and a summary of the results. The second figure shows an example portion of the report with results for specific tests.
### Redfish Conformance Test Report ###

[Image of Redfish logo]

https://github.com/DMTF/Redfish-Interop-Validator

Tool Version: 2.1.4  
Thu Oct 26 16:02:07 2023  
(Run time: 0:00:00)

This tool is provided and maintained by the DMTF. For feedback, please open issues in the tool’s Github repository: https://github.com/DMTF/Redfish-Interop-Validator/issues

#### Description: ####

Profile: ['OCPPBaselineHardwareManagement.v1_0_1.json']  
Schema: None

<table>
<thead>
<tr>
<th>auth_type: Basic</th>
<th>certificatebundle: None, certificatecheck: False, config: None</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>inaccessibleResource: 2</td>
<td>pass: 657</td>
</tr>
<tr>
<td>passGet: 477</td>
<td>totaltests: 654</td>
</tr>
</tbody>
</table>

---

Figure 5
### GET Success

```
fail.LinkStatus.ReadRequirement: 1
fail.NameServers.ReadRequirement: 1
pass: 38
passGet: 1
totalTests: 40
```

---

**Figure 6**

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Value</th>
<th>Expected</th>
<th>Actual</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>MinVersion</td>
<td>1.1.0</td>
<td>&lt;=</td>
<td>#EthernetInterface.v1_9_0.EthernetInterface</td>
<td>PASS</td>
</tr>
<tr>
<td>MACAddress ReadRequirement</td>
<td>Mandatory</td>
<td>Must Exist</td>
<td>Exists</td>
<td>PASS</td>
</tr>
<tr>
<td>SpeedMbps.ReadRequirement</td>
<td>Mandatory</td>
<td>Must Exist</td>
<td>Exists</td>
<td>PASS</td>
</tr>
<tr>
<td>InterfaceEnabled.ReadRequirement</td>
<td>Mandatory</td>
<td>Must Exist</td>
<td>Exists</td>
<td>PASS</td>
</tr>
<tr>
<td>LinkStatus.ReadRequirement</td>
<td>Mandatory</td>
<td>Must Exist</td>
<td>DNE</td>
<td>FAIL</td>
</tr>
<tr>
<td>Status.ReadRequirement</td>
<td>Mandatory</td>
<td>Must Exist</td>
<td>Exists</td>
<td>PASS</td>
</tr>
</tbody>
</table>
6 Appendix A: References

- DMTF DSP8010, Redfish Schema Bundle, https://www.dmtf.org/dsp/DSP8010
# 7 Appendix B: Change log

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0.0</td>
<td>2024-01-25</td>
<td>Initial release.</td>
</tr>
</tbody>
</table>

DSP2068

Redfish Conformance and Test Tools White Paper

Version 1.0.0

Published