Document Identifier: DSP2068

Date: 2024-01-25

Version: 1.0.0

Redfish Conformance and Test Tools White Paper

Supersedes: None

Document Class: Informational

Document Status: Published

Document Language: en-US

CONTENTS

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.

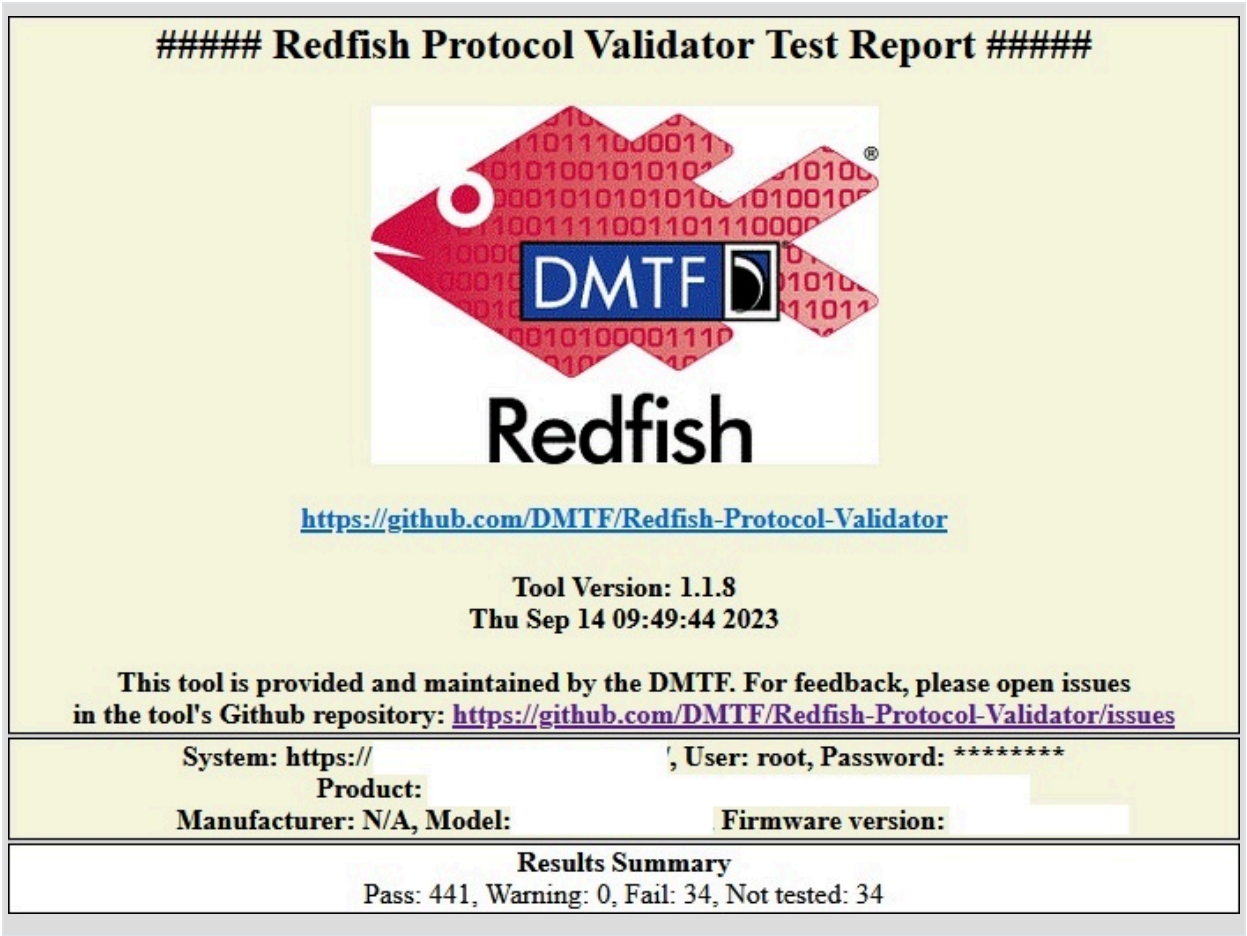

Redfish Protocol Validator Test Report
DMTF
Redfish
https://github.com/DMTF/Redfish-Protocol-Validator
Tool Version: 1.1.8
Thu Sep 14 09:49:44 2023
This tool is provided and maintained by the DMTF. For feedback, please open issues
in the tool's Github repository: https://github.com/DMTF/Redfish-Protocol-Validator/issues
System: https://
', User: root, Password: ********
Product:
Manufacturer: N/A, Model:
Firmware version:
Results Summary
Pass: 441, Warning: 0, Fail: 34, Not tested: 34

Figure 1

PROTO_STD_URIS_SUPPORTED: "A Redfish Service shall support these Redfish-defined URIs: /redfish, /redfish/v1/, /redfish/v1/odata, /redfish/v1/$metadata"				
Result	**Method**	**Status code**	**URI**	**Message**
PASS	GET	200	/redfish	Test passed
PASS	GET	200	/redfish/v1/odata	Test passed
PASS	GET	200	/redfish /v1/$metadata	Test passed
PASS	GET	200	/redfish/v1/	Test passed
PROTO_STD_URI_SERVICE_ROOT: "The root URI for this version of the Redfish protocol shall be /redfish/v1/."				
Result	**Method**	**Status code**	**URI**	**Message**
PASS	GET	200	/redfish/v1/	Test passed
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/)."				
Result	**Method**	**Status code**	**URI**	**Message**
PASS	GET	200	/redfish/v1	Test passed
PROTO_STD_URI_VERSION: "A GET operation on the /redfish resource shall return this response body: {"v1": "/redfish/v1/"}"				
Result	**Method**	**Status code**	**URI**	**Message**
PASS	GET	200	/redfish	Test passed

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.

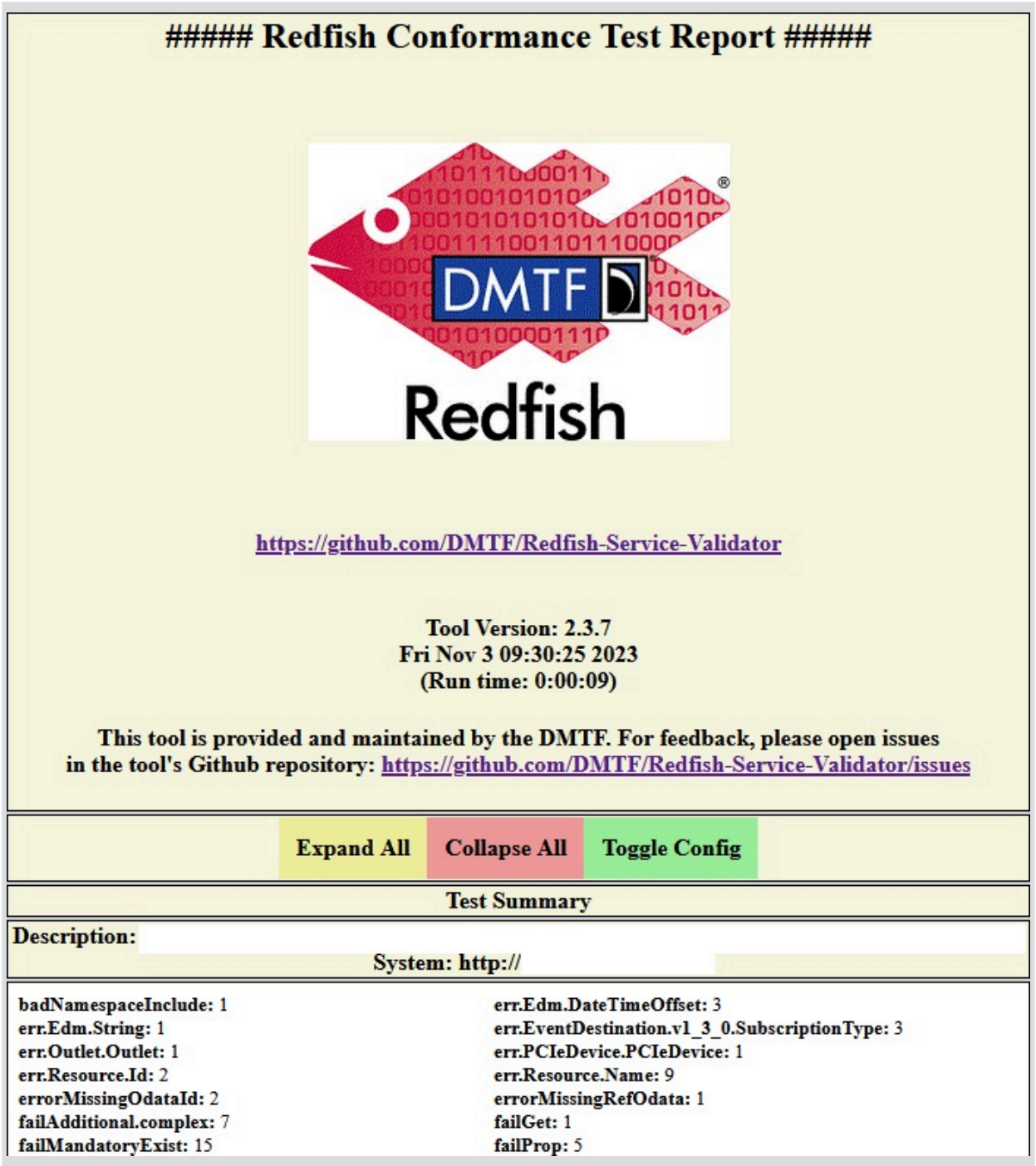
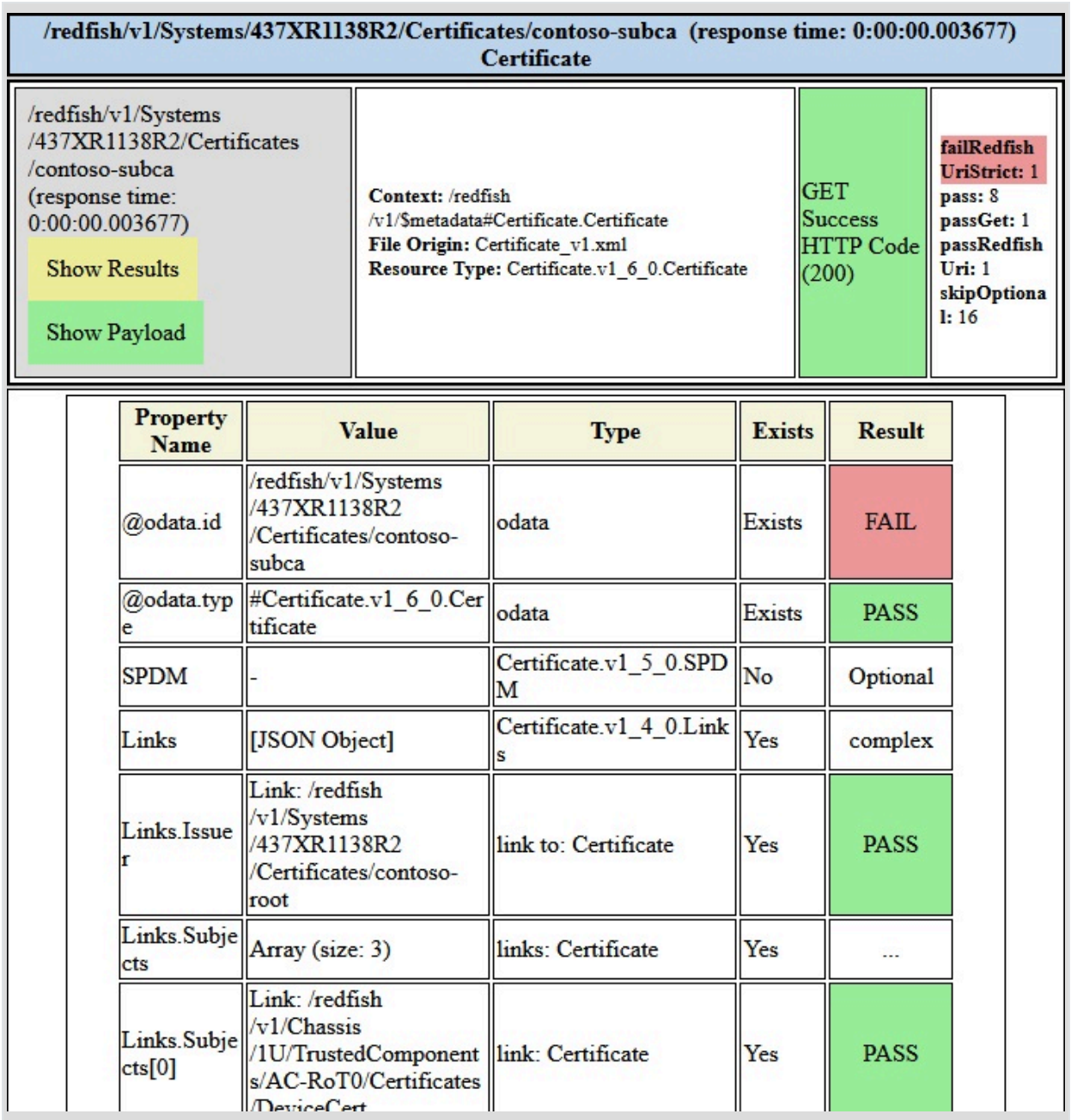

Redfish Conformance Test Report
DMTF
Redfish
https://github.com/DMTF/Redfish-Service-Validator
Tool Version: 2.3.7
Fri Nov 3 09:30:25 2023
(Run time: 0:00:09)
This tool is provided and maintained by the DMTF. For feedback, please open issues in the tool's Github repository: https://github.com/DMTF/Redfish-Service-Validator/issues
Expand All
Collapse All
Toggle Config
Test Summary
Description:
System: http://
badNamespaceInclude: 1
err.Edm.String: 1
err.Outlet.Outlet: 1
err.Resource.Id: 2
errorMissingOdataId: 2
failAdditional.complex: 7
failMandatoryExist: 15
err.Edm.DateTimeOffset: 3
err.EventDestination.v1_3_0.SubscriptionType: 3
err.PCIeDevice.PCIeDevice: 1
err.Resource.Name: 9
errorMissingRefOdata: 1
failGet: 1
failProp: 5

Figure 3



/redfish/v1/Systems/437XR1138R2/Certificates/contoso-subca (response time: 0:00:00.003677)
Certificate

/redfish/v1/Systems /437XR1138R2/Certificates /contoso-subca (response time: 0:00:00.003677) Show Results Show Payload	**Context:** /redfish /v1/$metadata#Certificate.Certificate **File Origin:** Certificate_v1.xml **Resource Type:** Certificate.v1_6_0.Certificate	GET Success HTTP Code (200)	**failRedfishUriStrict:** 1 **pass:** 8 **passGet:** 1 **passRedfishUri:** 1 **skipOptional:** 16

Property Name	Value	Type	Exists	Result
@odata.id	/redfish/v1/Systems /437XR1138R2 /Certificates/contoso-subca	odata	Exists	FAIL
@odata.type	#Certificate.v1_6_0.Certificate	odata	Exists	PASS
SPDM	-	Certificate.v1_5_0.SPDM	No	Optional
Links	[JSON Object]	Certificate.v1_4_0.Links	Yes	complex
Links.Issuer	Link: /redfish /v1/Systems /437XR1138R2 /Certificates/contoso-root	link to: Certificate	Yes	PASS
Links.Subjects	Array (size: 3)	links: Certificate	Yes	...
Links.Subjects[0]	Link: /redfish /v1/Chassis /1U/TrustedComponents/AC-RoT0/Certificates /DeviceCert	link: Certificate	Yes	PASS

Figure 4

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*.

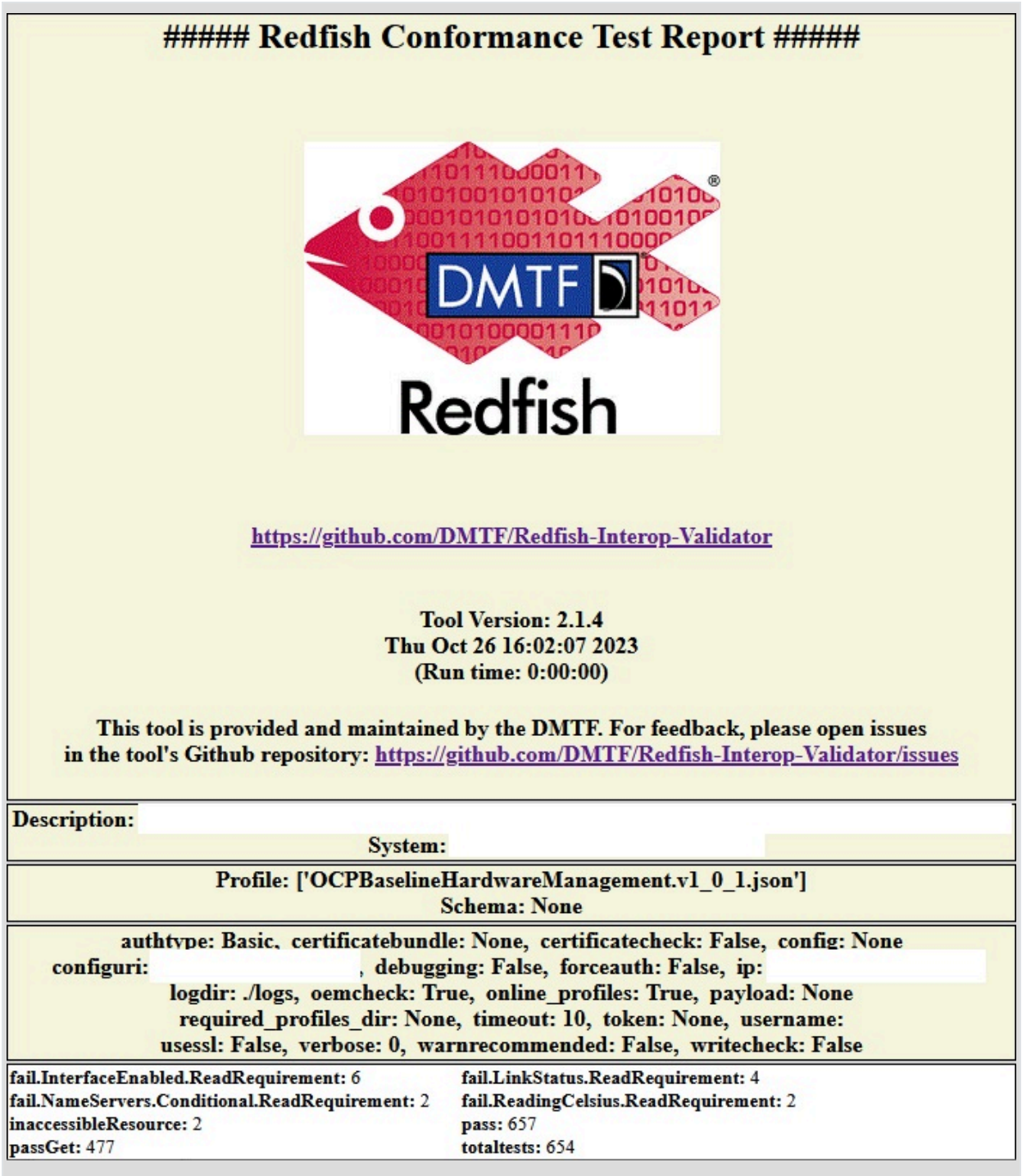
The following command can be used to run the Redfish Interop Validator:

```
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

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:

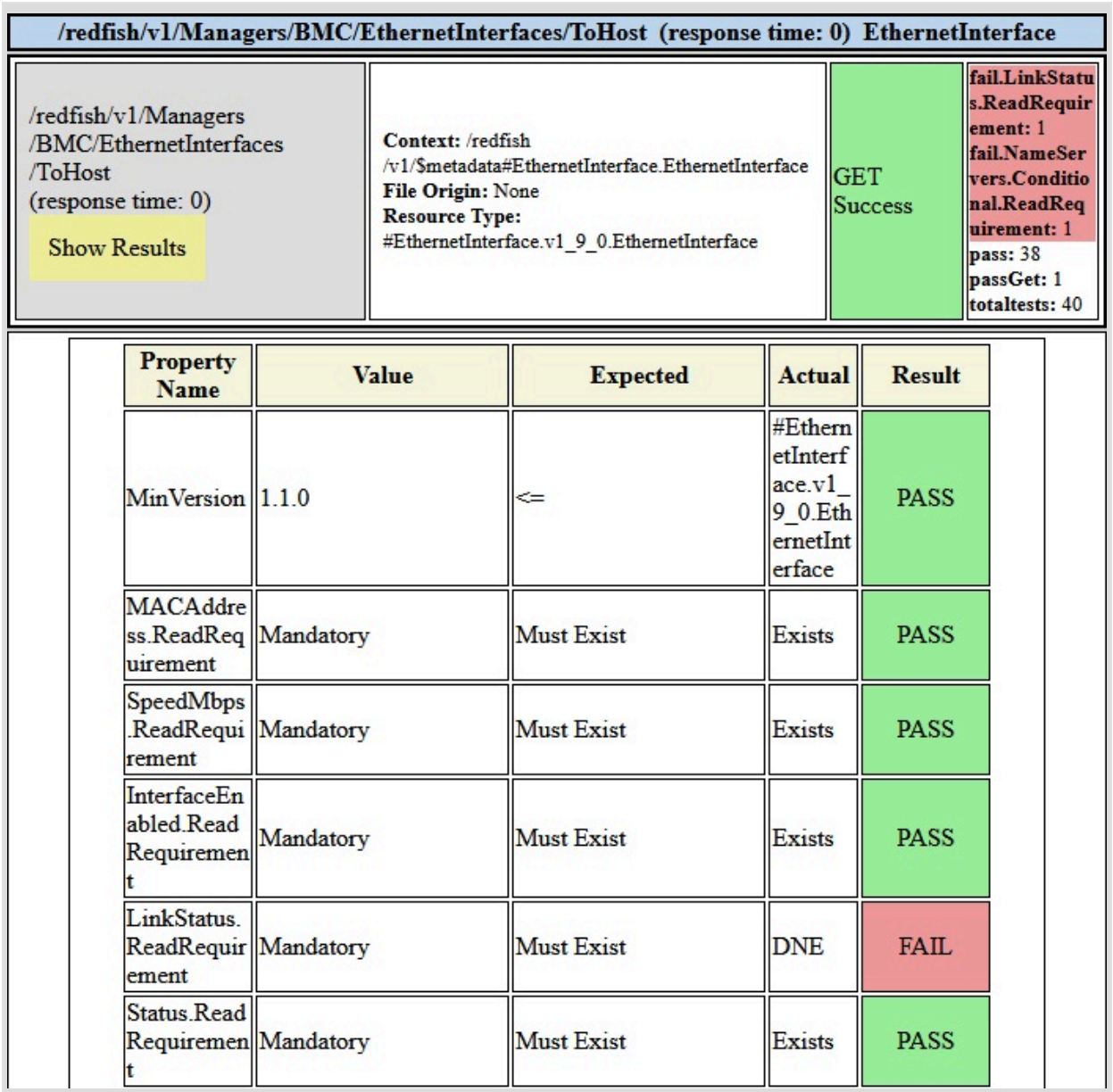
System:

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

authtype: Basic, certificatebundle: None, certificatecheck: False, config: None
configuri: , debugging: False, forceauth: False, ip:
logdir: ./logs, oemcheck: True, online_profiles: True, payload: None
required_profiles_dir: None, timeout: 10, token: None, username:
usessl: False, verbose: 0, warnrecommended: False, writecheck: False

fail.InterfaceEnabled.ReadRequirement: 6	fail.LinkStatus.ReadRequirement: 4
fail.NameServers.Conditional.ReadRequirement: 2	fail.ReadingCelsius.ReadRequirement: 2
inaccessibleResource: 2	pass: 657
passGet: 477	totaltests: 654

Figure 5



/redfish/v1/Managers/BMC/EthernetInterfaces/ToHost (response time: 0) EthernetInterface

/redfish/v1/Managers/BMC/EthernetInterfaces/ToHost (response time: 0)

Show Results

Context: /redfish/v1/$metadata#EthernetInterface.EthernetInterface
File Origin: None
Resource Type: #EthernetInterface.v1_9_0.EthernetInterface

GET Success

fail.LinkStatus.ReadRequirement: 1
fail.NameServers.Conditional.ReadRequirement: 1
pass: 38
passGet: 1
totaltests: 40

Property Name	Value	Expected	Actual	Result
MinVersion	1.1.0	<=	#EthernetInterface.v1_9_0.EthernetInterface	PASS
MACAddress.ReadRequirement	Mandatory	Must Exist	Exists	PASS
SpeedMbps.ReadRequirement	Mandatory	Must Exist	Exists	PASS
InterfaceEnabled.ReadRequirement	Mandatory	Must Exist	Exists	PASS
LinkStatus.ReadRequirement	Mandatory	Must Exist	DNE	FAIL
Status.ReadRequirement	Mandatory	Must Exist	Exists	PASS

Figure 6

6 Appendix A: References

- Redfish Protocol Validator, https://github.com/DMTF/Redfish-Protocol-Validator
- Redfish Service Validator, https://github.com/DMTF/Redfish-Service-Validator
- Redfish Interop Validator, https://github.com/DMTF/Redfish-Interop-Validator
- DMTF DSP0266, *Redfish Specification*, https://www.dmtf.org/dsp/DSP0266
- DMTF DSP0272, *Redfish Interoperability Profiles Specification*, https://www.dmtf.org/dsp/DSP0272
- DMTF DSP8010, *Redfish Schema Bundle*, https://www.dmtf.org/dsp/DSP8010
- OCP Hardware Management Profiles, https://github.com/opencomputeproject/HWMgmt-OCP-Profiles "https://github.com/opencomputeproject/HWMgmt-OCP-Profiles"
- The Open Group, Open Process Automation Forum Profiles, https://open-process-automation.projects.opengroup.org/twg/part-5-system-management/profiles/ "https://open-process-automation.projects.opengroup.org/twg/part-5-system-management/profiles/"
- OpenStack Ironic Profiles, https://github.com/openstack/ironic/tree/master/redfish-interop-profiles "https://github.com/openstack/ironic/tree/master/redfish-interop-profiles"

7 Appendix B: Change log

Version	Date	Description
1.0.0	2024-01-25	Initial release.