



Redfish

Document Identifier: DSP0285

Date: 2022-11-28

Version: 1.0.0

1
2
3
4

5 **RedPath Specification**

6 **Supersedes: None**
7 **Document Class: Normative**
8 **Document Status: Published**
9 **Document Language: en-US**

10 Copyright Notice

11 Copyright © 2022 DMTF. All rights reserved.

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

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26 withdrawn or modified after publication, and shall be indemnified and held harmless by any party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

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

32 This document’s normative language is English. Translation into other languages is permitted.

33

CONTENTS

34 Foreword 4
35 Acknowledgments 4
36 Introduction 4
37 1 Normative references 5
38 2 Terms, definitions, symbols, and abbreviated terms..... 5
39 3 RedPath query language..... 5
40 3.1 RedPath language constructs..... 7
41 4 RedPath interpreter design tenets 7
42 ANNEX A (informative) Change Log..... 8
43

44 Tables

45 Table 1 — RedPath Query Language Expressions 6
46

47

Foreword

48 The RedPath Specification was prepared by DMTF's Redfish Forum.

49 DMTF is a not-for-profit association of industry members that promotes enterprise and systems
50 management and interoperability. For information about DMTF, see [DMTF](#).

51 Acknowledgments

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

- 53 • Patrick Boyd — Dell Technologies
- 54 • Derek Chan — Google LLC

55

56

Introduction

57 RedPath is a string syntax to allow a client to specify a path to resources and properties in the Redfish
58 data model.

59 A RedPath interpreter is a Redfish client that is responsible for executing a sufficiently optimal query
60 strategy for a given RedPath string. The [Redfish Specification](#) defines optional query parameters that
61 might be available on a Redfish service for optimizing the sequence of client requests. A RedPath
62 interpreter implementation determines which query parameters are suitable against a particular Redfish
63 service.

64 This specification provides the RedPath syntax definition.

65 1 Normative references

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

69 DMTF DSP0266, *Redfish Specification*,
70 https://www.dmtf.org/sites/default/files/standards/documents/DSP0266_1.15.1.pdf

71 *XML Path Language (XPath) Version 1.0*,
72 <https://www.w3.org/TR/1999/REC-xpath-19991116/>

73 *ECMA-404, The JSON data interchange syntax, 2nd edition*,
74 https://www.ecma-international.org/wp-content/uploads/ECMA-404_2nd_edition_december_2017.pdf

75 2 Terms, definitions, symbols, and abbreviated terms

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

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

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

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

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

95 3 RedPath query language

96 RedPath is a query language based on [XPath 1.0](#). This language treats the entire Redfish service as
97 though it were a single JSON document. When an implementation encounters an `@odata.id` property, it
98 shall retrieve the resource specified by the `@odata.id` property if it is needed to satisfy the expression.
99 Implementations shall support the expressions in Table 1 — RedPath Query Language Expressions.

100

Table 1 — RedPath Query Language Expressions

Expression	Description
"nodename"	Selects the JSON entity with the name "nodename".
/	Selects from the root node.
["index"]	Selects the index number JSON entity from an array or object.
[last()]	Selects the last index number JSON entity from an array or object.
["nodename"]	Selects all the elements from an array or object that contain a property named "nodename".
["name"="value"]	Selects all the elements from an array or object where the property "name" is equal to "value".
["name"<"value"]	Selects all the elements from an array or object where the property "name" is less than "value".
["name"<="value"]	Selects all the elements from an array or object where the property "name" is less than or equal to "value".
["name">"value"]	Selects all the elements from an array or object where the property "name" is greater than "value".
["name">="value"]	Selects all the elements from an array or object where the property "name" is greater than or equal to "value".
["name"!="value"]	Selects all the elements from an array or object where the property "name" does not equal "value".
[*]	Selects all the elements from an array or object.
["node"."child"]	Selects all the elements from an array or object that contain a property named "node" that contains "child".

101 The following are example query expressions:

- 102 • /Chassis[1]: Returns the first Chassis resource.
- 103 • /Chassis[SKU=1234]: Returns all Chassis resources whose SKU property equals to 1234.
- 104 • /Systems[Storage]: Returns all the ComputerSystem resources that contain a Storage
- 105 property.
- 106 • /Systems[*]: Returns all the ComputerSystem resources.
- 107 • /Systems[*]/Processors[*]: Returns all Processor resources from all
- 108 ComputerSystem resources.
- 109 • /SessionService/Sessions[last()]: Returns the last Session resource.
- 110 • /Chassis[Location.Info]: Returns all the Chassis resources that contain a Location
- 111 property with an Info property.
- 112 • /Systems[Status.Health=OK]: Returns all ComputerSystem resources whose Health
- 113 property inside Status equals OK.
- 114 • /Systems[Status.Health=OK]/Memory[CapacityMiB>1024]: Returns all Memory
- 115 resources whose CapacityMiB property is greater than 1024 from all ComputerSystem
- 116 resources whose Health equals OK.

117 3.1 RedPath language constructs

118 RedPath is defined to be an absolute location path relative to the `/redfish/v1` node. A relative path is a
119 sequence of steps.

```
120 [1] RedPath ::= '/' RelativePath
121
122 [2] RelativePath ::= Step
123                    | RelativePath '/' Step
```

124 Each step selects a set of child nodes from the current node. The step can select a single specific child
125 node or provide a predicate that describes the criteria for selecting child nodes.

```
126 [3] Step ::= NodeName
127           | NodeName '[' Predicate ']'
128
129 [4] Predicate ::= Index
130                | '*'
131                | 'last()'
132                | NodeName Comparison NodeValue
133                | NodeName '.' NodeName
134
135 [5] Comparison ::= '=' | '<' | '<=' | '>' | '>=' | '!='
```

136 The node name is a qualified name referencing a Redfish property name. Its syntax matches valid syntax
137 for a Redfish property name as defined in the [Redfish Specification](#).

```
138 [6] NodeName ::= RedfishPropertyNameSyntax
```

139 The node value permits any valid JSON value as defined by the [JSON syntax](#).

```
140 [7] NodeValue ::= JsonValueSyntax
```

141 4 RedPath interpreter design tenets

142 Redpath is a declarative programming language designed to interoperate with any conformant interpreter
143 implementation that abstracts a Redfish Service as a single JSON document. Interpreters that implement
144 RedPath shall support interacting with Redfish services that do not support query parameters.

145
146
147
148

ANNEX A
(informative)
Change Log

Version	Date	Description
1.0.0	2009-03-16	Released as DMTF Standard

149