



Profile Error: This profile contains 5 errors (search for 'Error:')

2



3

Document Number: XMP1013

4

Date: 2011-08-31

5

Version: 1.0.2m

6

Example Fan Profile

7

8

IMPORTANT: This specification is not a standard. It does not necessarily reflect the views of the DMTF or all of its members. Because this document is a Work in Progress, this specification may still change, perhaps profoundly. This document is available for public review and comment until the stated expiration date.

9

This document expires on: **2012-02-28**.

10

Target version for DMTF Standard: **1.0.2**.

11

Provide any comments through the DMTF Feedback Portal: <http://www.dmtf.org/standards/feedback>

12

Document Type: Specification

13

Document Status: Work in Progress

14

Document Language: en-US

15

Copyright notice

- 16 Copyright © 2006-2011 Distributed Management Task Force, Inc. (DMTF). All rights reserved.
- 17 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 for uses consistent with this purpose, 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.
- 18 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patent owner or claimant, and shall have no liability or responsibility for costs or losses incurred if a standard is withdrawn or modified after publication, and shall be indemnified and held harmless by any party implementing the standard from any and all claims of infringement by a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.
- 19 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>.

20

CONTENTS

Foreword	6
1 Scope	8
2 Normative references	8
3 Terms and definitions	9
3.1 General	9
4 Symbols and abbreviated terms	9
5 Synopsis	9
6 Description	13
7 Implementation	14
7.1 Features	14
7.1.1 Feature: FanRedundancyByBalancing	14
7.1.2 Feature: FanRedundancyBySparing	14
7.1.3 Feature: FanSpeedSensor	15
7.1.4 Feature: PhysicalAssetDescription	15
7.1.5 Feature: PartialCooling	15
7.1.6 Feature: FanCapabilities	16
7.1.7 Feature: FanElementNameModification	16
7.1.8 Feature: FanStateManagement	16
7.1.9 Feature: SettingFanSpeed	17
7.1.10 Feature: FanIndications	17
7.2 Adaptations	17
7.2.1 Conventions	17
7.2.2 Adaptation: ComputerSystem: CIM_ComputerSystem	18
7.2.3 Adaptation: SystemDevice: CIM_SystemDevice	18
7.2.4 Adaptation: FanMetricDefinition: CIM_BaseMetricDefinition	19
7.2.5 Adaptation: Fan: CIM_Fan	19
7.2.6 Adaptation: FanCapabilities: CIM_EnabledLogicalElementCapabilities	28
7.2.7 Adaptation: ElementCapabilities: CIM_ElementCapabilities	29
7.2.8 Adaptation: CooledElement: CIM_ManagedSystemElement	30
7.2.9 Adaptation: AssociatedCooling: CIM_AssociatedCooling	31
7.2.10 Adaptation: FanRedundancySet: CIM_RedundancySet	32
7.2.11 Adaptation: OwningCollectionElement: CIM_OwningCollectionElement	35
7.2.12 Adaptation: HostedRedundancySet: CIM_HostedCollection	36
7.2.13 Adaptation: MemberOfRedundancySet: CIM_MemberOfCollection	36
7.2.14 Adaptation: IsSpare: CIM_IsSpare	37
7.2.15 Adaptation: NumericFanSpeedSensor: CIM_NumericSensor	39
7.2.16 Adaptation: DiscreteFanSpeedSensor: CIM_Sensor	40
7.2.17 Adaptation: FanAddedIndication: CIM_InstCreation	41
7.2.18 Adaptation: FanRemovedIndication: CIM_InstDeletion	41
7.2.19 Adaptation: FanHealthIndication: CIM_AlertIndication	41

7.2.20 Adaptation: FanRedundancyIndication: CIM_AlertIndication 42

7.2.21 Adaptation: FanSpeedAlertIndicationFilter: CIM_IndicationFilter 42

7.2.22 Adaptation: FanAddedLifecycleIndicationFilter: CIM_IndicationFilter 43

8 Use cases and state descriptions 45

8.1 State description: ObjectDiagram 45

8.2 Use case: SetFanSpeed 45

8.3 Use case: ResetFan 45

8.4 Use case: GetFanRedundancyStatus 45

8.5 Use case: FindSpareFan 45

8.6 Use case: ShowFanSensorInfo 45

8.7 Use case: FindCooledElements 45

8.8 Use case: DetermineElementNameModifiability 45

ANNEX A (informative) Change log 47

Bibliography 48

21

Figures

Figure 1 – DMTF collaboration structure diagram 13

22

Tables

Table 1 – Profile references 10

Table 2 – Message registry references 10

Table 3 – Metric registry references 10

Table 4 – Features 11

Table 5 – Adaptations 11

Table 6 – Use cases and state descriptions 12

Table 7 – ComputerSystem: Element requirements 18

Table 8 – SystemDevice: Element requirements 18

Table 9 – FanMetricDefinition: Element requirements 19

Table 10 – Fan: Element requirements 20

Table 11 – EnabledState Value Description 22

Table 12 – RequestStateChange(): Parameter requirements 25

Table 13 – RequestStateChange(): Error reporting requirements 25

Table 14 – RequestStateChange: Return values 27

Table 15 – SetSpeed(): Parameter requirements 27

Table 16 – SetSpeed: Return values 28

Table 17 – FanCapabilities: Element requirements 28

Table 18 – ElementCapabilities: Element requirements 30

Table 19 – CooledElement: Element requirements 31

Table 20 – AssociatedCooling: Element requirements 31

Table 21 – FanRedundancySet: Element requirements 32

Table 22 – Failover(): Parameter requirements 34

Table 23 – Failover: Return values 34

Table 24 – OwningCollectionElement: Element requirements 35

Table 25 – HostedRedundancySet: Element requirements 36

Table 26 – MemberOfRedundancySet: Element requirements 37

Table 27 – IsSpare: Element requirements 38

Table 28 – NumericFanSpeedSensor: Element requirements 39

Table 29 – DiscreteFanSpeedSensor: Element requirements 40

Table 30 – FanAddedIndication: Element requirements 41

Table 31 – FanRemovedIndication: Element requirements 41

Table 32 – FanHealthIndication: Element requirements 42

Table 33 – FanRedundancyIndication: Element requirements 42

Table 34 – FanSpeedAlertIndicationFilter: Element requirements 43

Table 35 – FanAddedLifecycleIndicationFilter: Element requirements 44

Foreword

24 This document was prepared by the Physical Platform Profiles Working Group and Server Management Working Group.

25 DMTF is a not-for-profit association of industry members dedicated to promoting enterprise and systems management and interoperability.

26 **Design Note:** This document contains design notes (like this one), that provide information about the way the document is written, or to demonstrate certain things. Such design notes would not appear in a released version of this document.

27 **Design Note:** This document represents DSP1013 (Fan Profile) version 1.0.1 plus some additions as a machine readable profile in MRP 1.1 format. Since machine readable profiles need to be compliant to DSP1001 1.1, this document utilizes the newly introduced concepts, such as adaptations, features and collaboration diagrams. Relative to DSP1013 1.0, this machine readable profile adds the following, in order to demonstrate its use:

- 28 • The use of standard messages defined in DSP8016 and DSP8007 as error messages, in the RequestStateChange() method of the Fan adaptation.
- 29 • The use of standard metrics defined in an assumed metric registry DSPsamr, in two variants: (1) The definition of a metric 'Metric1' directly on the Fan adaptation, demonstrating the most simple approach to defining metrics; (2) The definition of a metric 'Metric2' on a metric definition represented by the FanMetricDefinition adaptation, demonstrating the more flexible approach of defining a metric using a metric definition. In both variants, the metrics are represented using the base metric model defined in DSP1053 (Base Metrics Profile).
- 30 • The use of alert indications, demonstrating the most simple approach to defining alert indications as defined in DSP1001 1.1 and DSP1054 1.2.

31 Acknowledgements

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

- 32 • Jon Hass, Dell (editor)
- 33 • Khachatur Papanyan, Dell (editor)
- 34 • Jeff Hilland, HP (editor)
- 35 • Jim Davis, WBEM Solutions (editor)
- 36 • Enoch Suen, Dell
- 37 • Christina Shaw, HP
- 38 • Aaron Merkin, IBM
- 39 • Perry Vincent, Intel
- 40 • John Leung, Intel
- 41 • John Ackerley, Sun Microsystems

42 Document conventions

43 Any text in this document is in normal text font, with the following exceptions:

- 44 • References to clause names use normal text font; if they consist of more than one word, the clause name is quoted using double quotes, such as in "CIM elements".
 - 45 • Important terms that are used for the first time are marked in *italics* .
 - 46 • The usage of terms link to the term definition defined in the "Terms and definitions" clause, enabling easy navigation to the term definition.
 - 47 • ABNF rules are in `monospaced font` .
- 48 Format definitions in this document are specified using ABNF (see [RFC5234](#)), with the following deviations:
- 49 • Literal strings are to be interpreted as case-sensitive Unicode characters, as opposed to the definition in [RFC5234](#) that interprets literal strings as case-insensitive US-ASCII characters.
- 50

Example Fan Profile

51

1 Scope

52 The Fan Profile extends the management capabilities of referencing profiles by adding the capability to represent fans for manageability and describe redundant fans . The fan as a logical device is modeled as referencing the fan physical package for physical asset information, a sensor for sensor reading information, and the profile registration for the schema implementation version information.

53

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated or versioned references, only the edition cited (including any corrigenda or DMTF update versions) applies. For undated and unversioned references, the latest published edition of the referenced document (including any corrigenda or DMTF update versions) applies.

- 54 DMTF DSP0004, *CIM Infrastructure Specification 2.5*,
http://www.dmtf.org/standards/published_documents/DSP0004_2.5.pdf
- 55 DMTF DSP0223, *Generic Operations 1.0*,
http://www.dmtf.org/standards/published_documents/DSP0223_1.0.pdf
- 56 DMTF DSP1001, *Management Profile Specification Usage Guide 1.1*,
http://www.dmtf.org/standards/published_documents/DSP1001_1.1.pdf
- 57 DMTF XMP1009, *Example Sensors Profile (sample profile in DSP2023) 1.0*,
http://www.dmtf.org/standards/published_documents/DSP2023_1.0.zip
- 58 DMTF XMP1011, *Example Physical Asset Profile (sample profile in DSP2023) 1.0*,
http://www.dmtf.org/standards/published_documents/DSP2023_1.0.zip
- 59 DMTF XMP1033, *Example Profile Registration Profile (sample profile in DSP2023) 1.0*,
http://www.dmtf.org/standards/published_documents/DSP2023_1.0.zip
- 60 DMTF DSP1053, *Base Metric Profile 1.1*,
http://www.dmtf.org/standards/published_documents/DSP1053_1.1.pdf
- 61 DMTF DSP1054, *Indications Profile 1.2*,
http://www.dmtf.org/standards/published_documents/DSP1054_1.2.pdf
- 62 DMTF DSP8016, *WBEM Operations Message Registry 1.0*,
http://schemas.dmtf.org/wbem/messageregistry/1/dsp8016_1.0.xml
- 63 DMTF DSP8007, *Platform Message Registry 1.0*,
http://schemas.dmtf.org/wbem/messageregistry/1/dsp8007_1.0.xml
- 64 DMTF DSPsamr, *Sample Metric Registry 1.0*,
dpsamr_1.0.xml
- 65 IETF RFC5234, *Augmented BNF for Syntax Specifications: ABNF, 2008-01*,
<http://tools.ietf.org/html/rfc5234>
- 66 ISO/IEC Directives, Part 2, *Rules for the structure and drafting of International Standards*,
<http://isotc.iso.org/livelink/livelink?func=ll&objId=4230456&objAction=browse&sort=subtype>

67

3 Terms and definitions

In this document, some terms have a specific meaning beyond the normal English meaning. Those terms are defined in this clause.

68

3.1 General

69 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, Part2](#), Annex H. 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, Part2](#), Annex H specifies additional alternatives. Occurrences of such additional alternatives shall be interpreted in their normal English meaning in this document.

70 The terms "clause", "subclause", "paragraph", "annex" in this document are to be interpreted as described in [ISO/IEC Directives, Part2](#), Clause 5.

71 The terms "normative" and "informative" in this document are to be interpreted as described in [ISO/IEC Directives, Part2](#), Clause 3. In this document, clauses, subclauses or annexes indicated with "(informative)" as well as notes and examples do not contain normative content.

The terms defined in [DSP0004](#), [DSP0223](#), and [DSP1001](#) apply to this document.

72 The following additional terms are defined in this document.

73 3.2

fan

74 A device that provides thermal cooling by air flow to system elements.

75 3.3

redundant fan

76 A fan that is participating in a redundant set of fans.

77 3.4

spare fan

78 A fan that is not currently used, but is available for use in situations where currently used fans are no longer used for some reason.

79

4 Symbols and abbreviated terms

This clause defines the symbols and abbreviations used in this document.

80 The abbreviations defined in [DSP0004](#), [DSP0223](#), and [DSP1001](#) apply to this document.

81 This document does not define any additional abbreviations.

82

5 Synopsis

83 **Profile name:** Example Fan

84 **Version:** 1.0.2

85 **Organization:** DMTF

86 **Abstract:** No

87

Profile type: Component

88 **Schema:** DMTF CIM 2.19

89 **Central class adaptation:** Fan

90 **Scoping class adaptation:** ComputerSystem

91 **Scoping path:** SystemDevice

92 The Example Fan profile extends the management capabilities of referencing profiles by adding the capability to represent fans as logical devices, to monitor and control the state and speed of fans , to represent the relationship to elements cooled by fans , and to represent redundant fans as a redundancy group. Optionally, the Sensors profile can be implemented for fan speed sensors (SpeedSensors profile reference). Optionally, the Physical Asset profile can be implemented for fans (PhysicalAsset profile reference).

93 Table 1 identifies the profile references defined in this profile.

94 **Table 1 – Profile references**

Profile reference name	Profile name	Organization	Version	Relationship	Description
PhysicalAsset	Example Physical Asset	DMTF	1.0	Optional	Used to represent the physical packaging of fans.
SpeedSensors	Example Sensors	DMTF	1.0	Conditional	Used to represent fan speed sensors . Condition: The FanSpeedSensor feature is implemented.
PRP	Profile Registration	DMTF	1.0	Mandatory	Used to represent the implementation of this profile.
Indications	Indications	DMTF	1.2	Mandatory	Used for the indications defined by this profile.
BaseMetric	Base Metric	DMTF	1.1	Mandatory	Used for the metrics defined by this profile.

96 Table 2 identifies the message registry references defined in this profile.

97 **Table 2 – Message registry references**

Registry reference name	Registry ID	Organization	Version	Description
WBEMOperations	DSP8016	DMTF	1.0	
Platform	DSP8007	DMTF	1.0	

98 Table 3 identifies the metric registry references defined in this profile.

99 **Table 3 – Metric registry references**

Registry reference name	Registry ID	Organization	Version	Description
Sample	DSPsamr	DMTF	1.0	

100 Table 4 identifies the features defined in this profile.

101

Table 4 – Features

Feature	Requirement	Description
FanRedundancyByBalancing	Optional	See 7.1.1.
FanRedundancyBySparing	Optional	See 7.1.2.
FanSpeedSensor	Conditional	See 7.1.3.
PhysicalAssetDescription	Optional	See 7.1.4.
PartialCooling	Conditional	See 7.1.5.
FanCapabilities	Optional	See 7.1.6.
FanElementNameModification	Optional	See 7.1.7.
FanStateManagement	Optional	See 7.1.8.
SettingFanSpeed	Optional	See 7.1.9.
FanIndications	Optional	See 7.1.10.

102

Table 5 identifies the class adaptations defined in this profile.

103

Table 5 – Adaptations

Adaptation	Elements	Requirement	Description
Instantiated, embedded and abstract adaptations			
ComputerSystem	CIM_ComputerSystem	Mandatory	See 7.2.2.
SystemDevice	CIM_SystemDevice	Mandatory	See 7.2.3.
FanMetricDefinition	CIM_BaseMetricDefinition	Optional	See 7.2.4.
Fan	CIM_Fan	Mandatory	See 7.2.5.
FanCapabilities	CIM_EnabledLogicalElementCapabilities	Conditional	See 7.2.6.
ElementCapabilities	CIM_ElementCapabilities	Conditional	See 7.2.7.
CooledElement	CIM_ManagedSystemElement	Conditional	See 7.2.8.
AssociatedCooling	CIM_AssociatedCooling	Conditional	See 7.2.9.
FanRedundancySet	CIM_RedundancySet	Conditional	See 7.2.10.
OwningCollectionElement	CIM_OwningCollectionElement	Conditional	See 7.2.11.
HostedRedundancySet	CIM_HostedCollection	Conditional	See 7.2.12.
MemberOfRedundancySet	CIM_MemberOfCollection	Conditional	See 7.2.13.
IsSpare	CIM_IsSpare	Conditional	See 7.2.14.
NumericFanSpeedSensor	CIM_NumericSensor	Conditional	See 7.2.15.
DiscreteFanSpeedSensor	CIM_Sensor	Conditional	See 7.2.16.
FanSpeedAlertIndicationFilter	CIM_IndicationFilter	Conditional	See 7.2.21.
FanAddedLifecycleIndicationFilter	CIM_IndicationFilter	Conditional	See 7.2.22.
Indications and exceptions			
FanAddedIndication	CIM_InstCreation	Conditional	See 7.2.17.
FanRemovedIndication	CIM_InstDeletion	Conditional	See 7.2.18.
FanHealthIndication	CIM_AlertIndication	Conditional	See 7.2.19.
FanRedundancyIndication	CIM_AlertIndication	Conditional	See 7.2.20.

104

Table 6 identifies the use cases and state descriptions defined in this profile.

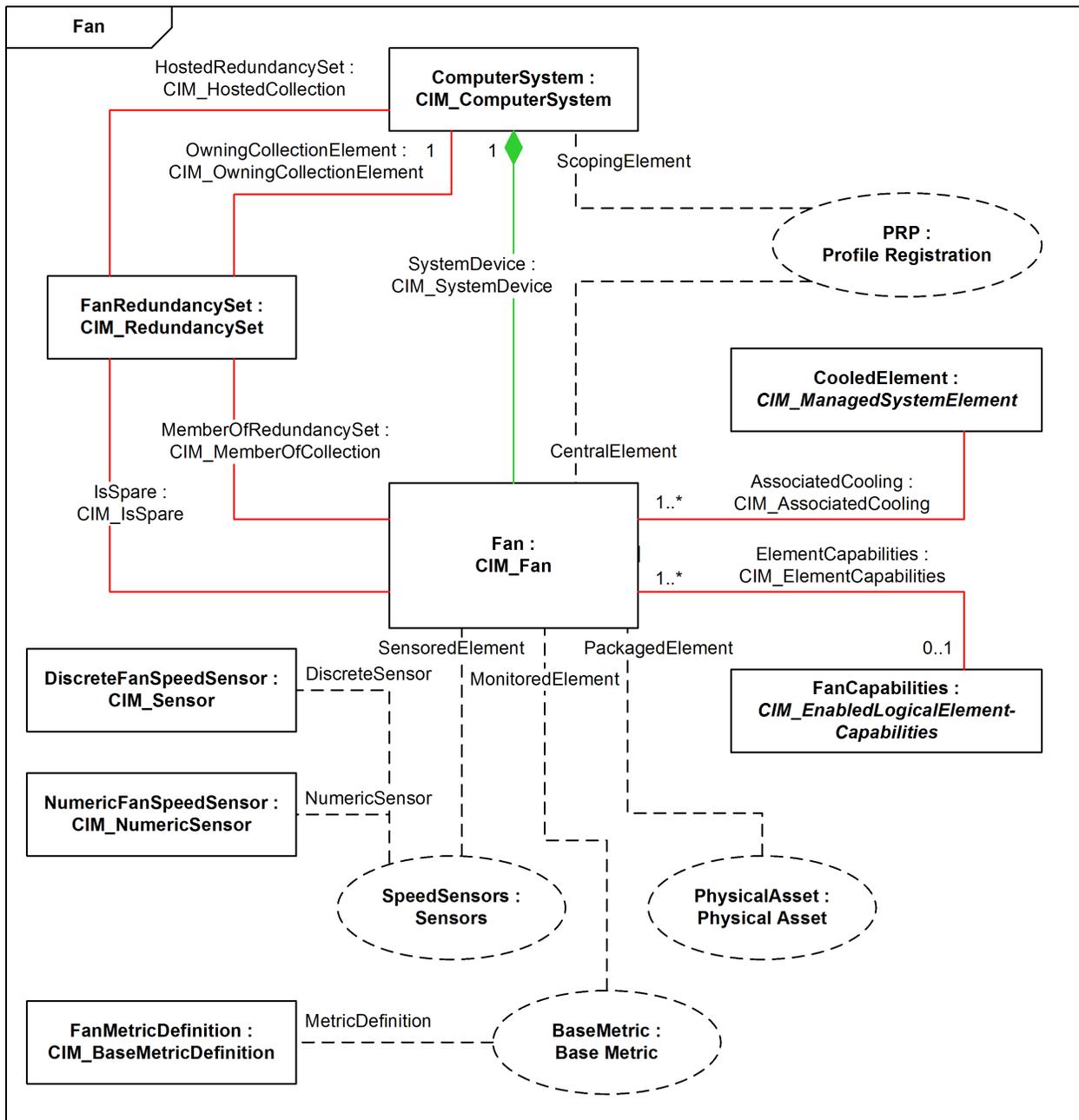
105

Table 6 – Use cases and state descriptions

Name	Description
State description: ObjectDiagram	See 8.1.
Use case: SetFanSpeed	See 8.2.
Use case: ResetFan	See 8.3.
Use case: GetFanRedundancyStatus	See 8.4.
Use case: FindSpareFan	See 8.5.
Use case: ShowFanSensorInfo	See 8.6.
Use case: FindCooledElements	See 8.7.
Use case: DetermineElementNameModifiability	See 8.8.

6 Description

107



108

Figure 1 – DMTF collaboration structure diagram

109

The logical aspect of fans in the managed environment is represented by instances of the Fan adaptation. The system hosting the fans is represented by an associated ComputerSystem instance.

110

The capability to support the representation of fan redundancy can be added by implementing the FanRedundancyBySparing or FanRedundancyByBalancing features, which are mutually exclusive for a given fan. If one of these features is implemented for a fan, each redundancy group this fan is a member of is represented by an associated FanRedundancySet instance.

111

The capability to support the sensing of the fan speed can be added by implementing the FanSpeedSensor feature for a fan, using the SpeedSensors profile.

112 The capability to expose physical asset information for a fan can be added by implementing the PhysicalAssetDescription feature for the fan, using the PhysicalAsset profile.

113 The capability to provide cooling only to specific elements of the system instead of the whole system can be added by implementing the PartialCooling feature. If it is implemented for a fan, the system element to which the fan provides cooling is represented by an associated CooledElement instance. If it is not implemented for a fan, the fan provides cooling to the entire system that hosts the fan.

114 The capability to expose the capabilities of a fan can be added by implementing the FanCapabilities feature. If it is implemented for a fan, its capabilities are represented by an associated FanCapabilities instance. Note that FanCapabilities instances can be shared between multiple Fan instances.

115 Conformance of an implementation to this profile is represented through the PRP profile.

116

7 Implementation

117

7.1 Features

118

7.1.1 Feature: FanRedundancyByBalancing

119

Requirement level: Optional

120 The implementation of this feature for a fan provides the ability to represent that a fan is redundant within a redundancy group of fans, such that all fans in the group run at the same time, balancing the cooling load between them. If one of these fan fails, the others remain running to provide cooling.

121 This feature can be made available to clients at the granularity of Fan instances.

122 It can be concluded that the feature is available for a Fan instance if:

- 124 • The following OCL derivation constraint evaluates to a Boolean value of True.

125 OCL context: A Fan instance.

```
126 derive: self.MemberOfRedundancySet::Collection->size() > 0 and  
self.IsSpare.Antecedent->size() = 0
```

127 Otherwise, it can be concluded that the feature is not available.

128

7.1.2 Feature: FanRedundancyBySparing

129

Requirement level: Optional

130 The implementation of this feature for a fan provides the ability to represent that a fan is redundant within a redundancy group of fans, such that some fans in the group run at the same time, balancing the cooling load between them, and some others do not normally run and act as spare fans . If one of the normally running fans fails, the other normally running fans remain running to provide cooling; and additional spare fans may be started to accomodate for the cooling capacity of the failed fan(s).

131 This feature can be made available to clients at the granularity of Fan instances.

132 It can be concluded that the feature is available for a Fan instance if:

- 134 • The following OCL derivation constraint evaluates to a Boolean value of True.

135 OCL context: A Fan instance.

136

```
derive: self.MemberOfRedundancySet::Collection->size() > 0 and
self.IsSpare.Antecedent->size() > 0
```

137 Otherwise, it can be concluded that the feature is not available.

138 7.1.3 Feature: FanSpeedSensor

139 **Requirement level:** Conditional

140 **Condition:**

Profile Error: Invalid type of condition "ManagedEnvironmentCondition" defined on an element.

142 This feature provides the ability to expose the speed of fans that have a speed sensor .

143 This feature can be made available to clients at the granularity of Fan instances.

144 It can be concluded that the feature is available for a Fan instance if:

- 146 • The following OCL derivation constraint evaluates to a Boolean value of True.

147 OCL context: A Fan instance.

```
148 derive: self.SpeedSensors::NumericSensor->size() > 0 or
self.SpeedSensors::DiscreteSensor->size() > 0
```

149 Otherwise, it can be concluded that the feature is not available.

150 7.1.4 Feature: PhysicalAssetDescription

151 **Requirement level:** Optional

152 This feature provides support for describing physical asset information of a fan.

153 This feature can be made available to clients at the granularity of an implementation of this profile.

154 It can be concluded that the feature is available if:

- 156 • The following OCL derivation constraint evaluates to a Boolean value of True.

157 OCL context: A RegisteredProfile instance for this profile.

```
158 derive: self.mrpIsReferencedProfileImplemented('PhysicalAsset')
```

159 Otherwise, it can be concluded that the feature is not available.

160 7.1.5 Feature: PartialCooling

161 **Requirement level:** Conditional

162 **Condition:**

Profile Error: Invalid type of condition "ManagedEnvironmentCondition" defined on an element.

164 The implementation of this feature for a fan allows representing that the fan provides cooling to a subset of the elements in a system, instead of to the entire system.

165 This feature can be made available to clients at the granularity of Fan instances.

166 It can be concluded that the feature is available for a Fan instance if:

- 168 • The following OCL derivation constraint evaluates to a Boolean value of True.

169

OCL context: A Fan instance.

170 `derive: self.AssociatedCooling::Antecedent->size() > 0`

171 Otherwise, it can be concluded that the feature is not available.

172 **7.1.6 Feature: FanCapabilities**

173 **Requirement level:** Optional

174 This feature allows a fan to expose its capabilities through an FanCapabilities instance.

175 This feature can be made available to clients at the granularity of Fan instances.

176 It can be concluded that the feature is available for a Fan instance if:

- 178 • The following OCL derivation constraint evaluates to a Boolean value of True.

179 OCL context: A Fan instance.

180 `derive: self.ElementCapabilities->size() = 1`

181 Explanation:

182 One instance exists of ElementCapabilities that is associated to the Fan instance.

183 Otherwise, it can be concluded that the feature is not available.

184 **7.1.7 Feature: FanElementNameModification**

185 **Requirement level:** Optional

186 This feature provides support for client modification of the CIM_Fan.ElementName property of a fan.

187 This feature can be made available to clients at the granularity of Fan instances.

188 It can be concluded that the feature is available for a Fan instance if:

- 190 • The following OCL derivation constraint evaluates to a Boolean value of True.

191 OCL context: A Fan instance.

192 `derive: self.ElementCapabilities::Capabilities.ElementNameEditSupported = true`

193 Otherwise, it can be concluded that the feature is not available.

194 **7.1.8 Feature: FanStateManagement**

195 **Requirement level:** Optional

196 This feature provides support for client management of the state of a fan.

197 This feature can be made available to clients at the granularity of Fan instances.

198 It can be concluded that the feature is available for a Fan instance if:

- 200 • The following OCL derivation constraint evaluates to a Boolean value of True.

201 OCL context: A Fan instance.

202 `derive: self.ElementCapabilities::Capabilities.RequestedStatesSupported->size()
203 > 0`

203 Otherwise, it can be concluded that the feature is not available.

204

7.1.9 Feature: SettingFanSpeed

205 **Requirement level:** Optional

206 This feature provides support for setting the speed of a fan, via the CIM_Fan.SetSpeed() method.

207 This feature can be made available to clients at the granularity of Fan instances.

208 Availability of this feature cannot be discovered by clients (other than trying the functionality provided by the feature).

7.1.10 Feature: FanIndications

210 **Requirement level:** Optional

211 **Design Note:** This feature has been defined to demonstrate the grouping of indication implementation decisions under one point of decision; this feature is not part of DSP1013 1.0. Note, a profile is free to define more granular features for indications, or to define indications without usage of any feature for grouping purposes.

212 This feature provides support for indications related to a fan.

213 This feature can be made available to clients at the granularity of an implementation of this profile.

214 Availability of this feature cannot be discovered by clients (other than trying the functionality provided by the feature).

7.2 Adaptations

7.2.1 Conventions

217 This profile defines operation requirements based on [DSP0223](#).

218 For adaptations of ordinary classes and of associations, the requirements for operations are defined in adaptation-specific subclauses of subclause 7.2.

219 For association traversal operation requirements that are specified only in the elements table of an adaptation (i.e., without operation-specific subclauses), the names of the association adaptations to be traversed are listed in the elements table.

220 The default initialization requirement level for property requirements is optional.

221 The default modification requirement level for property requirements is optional.

222 This profile repeats the effective values of certain Boolean qualifiers as part of property, method parameter, or method return value requirements. The following convention is established: If the name of a qualifier is listed, its effective value is True; if the qualifier name is not listed, its effective value is False. The convention is applied in the following cases:

- 223 • In: indicates that the parameter is an input parameter
- 224 • Out: indicates that the parameter is an output parameter
- 225 • Key: indicates that the property is a key (that is, its value is part of the instance path)
- 226 • Required: indicates that the element value shall be non-Null
- 227 • Null OK: indicates explicitly that the element value may be Null for mandatory, conditional or conditional exclusive properties. This information is not specified as a qualifier in the schema but as an indicator in the profile.

228

7.2.2 Adaptation: ComputerSystem: CIM_ComputerSystem

229 This adaptation does not define a model description.

230 **Adaptation type:** Ordinary class

231 **Implementation type:** Instantiated

232 **Requirement level:** Mandatory

233 **Table 7 – ComputerSystem: Element requirements**

Element	Requirement	Description
Operations		
Associators()	Mandatory	
AssociatorNames()	Mandatory	
References()	Mandatory	
ReferenceNames()	Mandatory	

234 7.2.3 Adaptation: SystemDevice: CIM_SystemDevice

235 7.2.3.1 General

236 **Adaptation type:** Association class

237 **Implementation type:** Instantiated

238 **Requirement level:** Mandatory

239 This adaptation does not define a model description.

240 **Table 8 – SystemDevice: Element requirements**

Element	Requirement	Description
Properties		
GroupComponent	Mandatory	Key, see 7.2.3.2
PartComponent	Mandatory	Key, see 7.2.3.3
Operations		
GetInstance()	Mandatory	

241 7.2.3.2 Property: GroupComponent

242 **Requirement level:** Mandatory

243 **Reference kind:** REF-typed

244 **Constraints:**

- 245 • Referenced instances shall be of class adaptation ComputerSystem.
- 246 • The multiplicity of this association end is 1 .. 1

247 7.2.3.3 Property: PartComponent

248 **Requirement level:** Mandatory

249 **Reference kind:** REF-typed

250

Constraints:

- 251 • Referenced instances shall be of class adaptation Fan.
- 252 • The multiplicity of this association end is 1 .. *

7.2.4 Adaptation: FanMetricDefinition: CIM_BaseMetricDefinition

254 **Design Note:** This adaptation has been defined to demonstrate the definition of metrics through a metric definition; it is not part of DSP1013 1.0.

255 This adaptation models metric definitions for fans.

256 **Adaptation type:** Ordinary class

257 **Implementation type:** Instantiated

258 **Requirement level:** Optional

259 **Table 9 – FanMetricDefinition: Element requirements**

Element	Requirement	Description
Base adaptations		
BaseMetric::BaseMetricDefinition	Optional	See BaseMetric::BaseMetricDefinition.
Metrics		
Sample::Metric1	Optional	Sample metric #1

7.2.5 Adaptation: Fan: CIM_Fan**7.2.5.1 General**

262 **Adaptation type:** Ordinary class

263 **Implementation type:** Instantiated

264 **Requirement level:** Mandatory

265 This adaptation models fans in the managed environment.

266 **Design Note:** This adaptation defines an additional base adaptation 'MonitoredElement' and a metric 'Metric2' to demonstrate the definition of metrics; this is not part of DSP1013 1.0.

267 Each fan in the managed environment should be represented by a Fan instance.

Constraints:

- 270 • OCL constraint in the context of a Fan instance:

```
271 inv:
    ( mrpIsFeatureSupported('FanCapabilities',self) or
      mrpIsFeatureSupported('FanElementNameModification',self) )
    implies
        self.cIM_ElementCapabilities.Capabilities->size() = 1
```

272 Explanation:

273 If the FanCapabilities feature or the FanElementNameModification feature are supported for a fan, then there shall be one FanCapabilities instance associated via ElementCapabilities to the Fan instance representing that fan.

- 275 • OCL constraint in the context of a Fan instance:

```
inv:
self.SystemDevice.ComputerSystem->size() = 1
```

277 Explanation:
 278 There shall be one ComputerSystem instance associated via SystemDevice to the Fan
 instance representing that fan.

280 • OCL constraint in the context of a Fan instance:

```
281 inv:
mrpIsFeatureSupported('FanRedundancyByBalancing',self) or
mrpIsFeatureSupported('FanRedundancyBySparing',self)
implies
  let rgfans : Set(Fan) =
    self.MemberOfCollection.Collection.MemberOfCollection.Member
    /* rgfans is the set of fans in the redundancy group of the
       current redundant fan (i.e. self) */
  in
    if mrpIsFeatureSupported('PartialCooling',self)
    then rgfans->forall( rgfan |
      rgfan.AssociatedCooling.Dependent =
        self.AssociatedCooling.Dependent)
    else rgfans->forall( rgfan |
      rgfan.SystemDevice.System =
        self.SystemDevice.System)
    endif
```

282 Explanation:
 283 If feature 'fan redundancy' is supported for a fan and feature 'partial cooling' is supported
 for the same fan, the CIM_Fan instances in the redundancy group of that fan shall be
 associated with the same CIM_ManagedSystemElement instance through
 CIM_AssociatedCooling associations.

284 If feature 'fan redundancy' is supported for a fan and feature 'partial cooling' is not
 supported for the same fan, the CIM_Fan instances in the redundancy group of that fan
 shall be associated with the same CIM_ComputerSystem instance through
 CIM_SystemDevice associations.

285 **Table 10 – Fan: Element requirements**

Element	Requirement	Description
Base adaptations		
SpeedSensors::SensoredElement	Optional	See SpeedSensors::SensoredElement.
PhysicalAsset::PackagedElement	Optional	See PhysicalAsset::PackagedElement.
BaseMetric::MonitoredElement	Optional	See BaseMetric::MonitoredElement.
Metrics		
Sample::Metric2	Optional	Sample metric #2
Properties		
SystemCreationClassName	Mandatory	Key
SystemName	Mandatory	Key
CreationClassName	Mandatory	Key
DeviceID	Mandatory	Key

Element	Requirement	Description
OperationalStatus	Mandatory	
HealthState	Mandatory	
VariableSpeed	Mandatory	
DesiredSpeed	Mandatory	See 7.2.5.2
ActiveCooling	Mandatory	See 7.2.5.3
EnabledState	Mandatory	See 7.2.5.4
RequestedState	Mandatory	See 7.2.5.5
ElementName	Mandatory	See 7.2.5.6
Methods		
RequestStateChange()	Conditional	See 7.2.5.7
SetSpeed()	Conditional	See 7.2.5.8
Operations		
GetInstance()	Mandatory	
EnumerateInstances()	Mandatory	
EnumerateInstanceNames()	Mandatory	
ModifyInstance()	Conditional	See 7.2.5.9
Associators()	Mandatory	
AssociatorNames()	Mandatory	
References()	Mandatory	
ReferenceNames()	Mandatory	

- 286 **7.2.5.2 Property: DesiredSpeed**
- 287 **Requirement level:** Mandatory
- 288 If setting the fan speed is supported, the meaning of the value 0 is that no change in fan speed has been requested.
- 289 If setting the fan speed is not supported, the value of this property is meaningless.
- 290 **Constraints:**
- 292 • OCL constraint in the context of a Fan instance:
- 293

```
init:
mrpIsFeatureSupported('SettingFanSpeed',self)
implies
    self.DesiredSpeed = 0
```
- 294 **Explanation:**
- 295 If setting the fan speed is supported for the fan, the initial value of DesiredSpeed shall be 0.
- 297 • OCL constraint in the context of a Fan instance:
- 298

```
inv:
not mrpIsFeatureSupported('SettingFanSpeed',self)
implies
    self.DesiredSpeed = 0
```
- 299

Explanation:

300 If setting the fan speed is not supported for the fan, the value of DesiredSpeed shall be 0.

301 **7.2.5.3 Property: ActiveCooling**

302 **Requirement level:** Mandatory

303 **Constraint:**

304 OCL constraint in the context of a Fan instance:

```
305 inv:
self.ActiveCooling = True
```

306 Explanation:

307 ActiveCooling shall match True.

308 **7.2.5.4 Property: EnabledState**

309 **Requirement level:** Mandatory

310 Table 11 describes the mapping between the EnabledState property values and the corresponding description of the state of the fan.

311 **Table 11 – EnabledState Value Description**

Value	Description	Extended Description
2	Enabled	The fan shall be turned on.
3	Disabled	The fan shall be turned off.
5	Not Applicable	The fan state is indeterminate, or fan state management is not supported.

312 The value of the EnabledState property may change as the result of a change to the fan's enabled state by a non-CIM implementation.

313 **7.2.5.5 Property: RequestedState**

314 **Requirement level:** Mandatory

315 **Constraints:**

- 317 • OCL constraint in the context of a Fan instance:

```
318 inv:
mrpIsFeatureSupported('FanStateManagement',self)
implies
    Set { 5 /* No Change */, 12 /* Not Applicable */ }->
        union( self.ElementCapabilities.Capabilities->
            asOrderedSet()->at(1).RequestedStatesSupported)->
            includes(self.RequestedState)
```

319 Explanation:

320 If feature 'FanStateManagement' is supported for a fan, the value of RequestedState shall be 5 (No Change), 12 (Not Applicable), or one of the values in the RequestedStatesSupported array of the associated CIM_EnabledLogicalElementCapabilities instance.

- 322 • OCL constraint in the context of a Fan instance:

```

inv:
not mrpIsFeatureSupported('FanStateManagement',self)
implies
    self.RequestedState = 12 /* Not Applicable */

```

324 Explanation:

325 If feature 'fan state management' is not supported for a fan, the value of the RequestedState property shall be 12 (Not Applicable).

327 • OCL constraint in the context of a Fan instance:

```

328 init:
mrpIsFeatureSupported('FanStateManagement',self)
implies
    self.RequestedState = 5 /* No Change */

```

329 Explanation:

330 If feature 'FanStateManagement' is supported for a fan, the initial value of RequestedState shall be 5 (No Change).

332 • OCL constraint in the context of a Fan instance:

```

333 inv:
not mrpIsFeatureSupported('FanStateManagement',self)
implies
    self.RequestedState = 12 /* Not Applicable */

```

334 Explanation:

335 If feature 'FanStateManagement' is not supported for a fan, the value of the RequestedState property shall be 12 (Not Applicable).

337 • OCL constraint in the context of a Fan instance:

```

338 inv:
not mrpIsFeatureSupported('FanStateManagement',self)
implies
    self.RequestedState = 12 /* Not Applicable */

```

339 Explanation:

340 If feature 'fan state management' is supported for a fan, the value of the RequestedState property shall be 12 (Not Applicable).

341 7.2.5.6 Property: ElementName

342 Requirement level: Mandatory

343 Constraint:

344 OCL constraint in the context of a Fan instance:

```

345 inv:
self.ElementCapabilities.Capabilities.
    ElementNameEditSupported = True
implies
    self.ElementName.isModifiable()

```

346 Explanation:

347 The ElementName property shall be modifiable when the ElementNameEditSupported
property of the CIM_EnabledLogicalElementCapabilities instance that is associated with the
CIM_Fan instance has a value of True.

348 7.2.5.7 Method: RequestStateChange()

349 **Requirement level:** Conditional

350 **Condition:**

The FanStateManagement feature is implemented.

351 **Design Note:** This method defines error reporting requirements based on standard messages. It
represents the error situations 'method not supported' and 'timeout' as errors, instead of as return values
as defined in DSP1013 1.0.

352 If feature 'fan state management' is supported, the method shall be implemented.

353 When the RequestStateChange() method does not complete successfully and the fan is in an
indeterminate state, the EnabledState property shall have a value of 5 (Not Applicable).

355 Invocation of the RequestStateChange() method changes the fan's state to the value specified in the
RequestedState parameter.

356 Invoking this method multiple times may result in earlier requests being overwritten or lost.

357 **Constraints:**

- 359 • OCL constraint in the context of a Fan instance:

```
360 pre:
Set { 2 /* Enabled */, 3 /* Disabled */, 11 /* Reset */ }->
includes(self.RequestStateChange.RequestedState)
```

361 **Explanation:**

362 The RequestedState parameter shall have one of the following values: 2 (Enabled), 3
(Disabled), 11 (Reset).

- 364 • OCL constraint in the context of a Fan instance:

```
365 post:
self.RequestedState = self.RequestStateChange.RequestedState
```

366 **Explanation:**

367 After successful completion of the RequestStateChange() method, the RequestedState
property of the CIM_Fan instance for which the method was invoked, shall have the value
specified in the RequestedState parameter.

- 369 • OCL constraint in the context of a Fan instance:

```
370 post:
self.IsSpare->size() > 0 and
self.RequestStateChange.RequestedState = 3 /* Disabled */
implies
self.IsSpare.SpareStatus = 3 /* Cold Standby */
```

371 **Explanation:**

372 After successful completion of the RequestStateChange() method on a CIM_Fan instance
representing a spare fan with the RequestedState parameter set to 3 (Disabled), the

SpareStatus property of the CIM_IsSpare association referencing this CIM_Fan instance shall have a value of 3 (Cold Standby).

373

Table 12 – RequestStateChange(): Parameter requirements

Parameter	Description
RequestedState	In. For valid values, see method constraints.
Job	Out, see 7.2.5.7.2
TimeoutPeriod	In, see 7.2.5.7.3
Return value	See 7.2.5.7.4

374

Table 13 – RequestStateChange(): Error reporting requirements

Reporting mechanism	Requirement level	Description
375 376 377 378 WBEMOperations::WIPG0208, Platform::PLAT9001	Mandatory	The requested state is not supported for the fan. Design Note: The messages are: <ul style="list-style-type: none"> WIPG0208: Invalid input parameter value PLAT9001(example): Requested state not supported for the element
379 380 381 382 WBEMOperations::WIPG0208, Platform::PLAT9002	Mandatory	A non-Null value for the Timeout parameter is not supported. Design Note: The messages are: <ul style="list-style-type: none"> WIPG0208: Invalid input parameter value PLAT9002(example): Timeout not supported for the method
383 384 385 386 WBEMOperations::WIPG0219	Mandatory	Method is not implemented. Note: This error situation and its reporting through this message is defined already in DSP0223. This error situation is listed in this table only because it was reported through a return value in earlier versions of this profile. Design Note: The messages are: <ul style="list-style-type: none"> WIPG0219: Method not supported by class implementation
387 388 389 390 WBEMOperations::WIPG0227, Platform::PLAT9003	Mandatory	Fan cannot be disabled due to excessive temperature. The detail text of WIPG0227 should be omitted or should indicate that the next message details the error. Design Note: The messages are: <ul style="list-style-type: none"> WIPG0227: Other failure

Reporting mechanism	Requirement level	Description
		<ul style="list-style-type: none"> PLAT9003(example): Fan cannot be disabled due to excessive temperature
391 392 WBEMOperations::WIPG0227	Mandatory	<p>Any other failure. As defined in WIPG0227, the failure shall be described in its detail text.</p> <p>Note: This error situation and its reporting through this message is defined already in DSP0223. This error situation is listed in this table only because it was reported through a return value in earlier versions of this profile.</p> <p>Design Note: The messages are:</p> <ul style="list-style-type: none"> WIPG0227: Other failure
395 396 CIM_ERR_SERVER_LIMITS_EXCEEDED	Mandatory	<p>More element changes are under way than the configured limit of concurrent changes, or there is a resource shortage in the WBEM server.</p> <p>Design Note: This error situation demonstrates the possibility of mixing the usage of CIM status codes and messages. It is not recommended to define such a mixed usage in a single profile, but it may happen in merged profiles.</p>

397 **7.2.5.7.1 Parameter: RequestedState**

For valid values, see method constraints.

398 **7.2.5.7.2 Parameter: Job**

A non-Null instance path is returned if a job was started. If no job was started, Null is returned.

399 **Constraint:**

400 Referenced instances shall be of class adaptation ConcreteJob
Profile Error: A class adaptation "ConcreteJob" is referenced in a class adaptation link but is not defined or is defined more than once in this profile..

401 **7.2.5.7.3 Parameter: TimeoutPeriod**

402 Client-specified maximum amount of time the transition to a new state is supposed to take:

- 403 • 0 or Null – No maximum time is specified
- 404 • Non-Null – The value specifies the maximum time allowed

405 **7.2.5.7.4 Return value**

406 This method shall return one of the following return values:

407

Table 14 – RequestStateChange: Return values

Value	Description
0	The state change was successfully performed.
1	The method is not implemented.
2	An error has occurred.
4096	The request to change the state is being executed asynchronously, and the Job parameter references a ConcreteJob Profile Error: A class adaptation "ConcreteJob" is referenced in a class adaptation link but is not defined or is defined more than once in this profile. instance representing the request.

408 **7.2.5.8 Method: SetSpeed()**409 **Requirement level:** Conditional410 **Condition:**

The SettingFanSpeed feature is implemented.

If the feature is not supported, the method shall not be implemented or shall return a value of 1 (Not Supported).

412 The SetSpeed() method requests that the speed of the fan represented by CIM_Fan be set to the value specified in the method's DesiredSpeed input parameter.

413 **Constraints:**

- 415 • OCL constraint in the context of a Fan instance:

416

```

body:
if (self.VariableSpeed = false or
    self.EnabledState = 3 /* Disabled */)
then 1 /* Not Supported */

```

417 Explanation:

418 When the CIM_Fan.VariableSpeed property has a value of FALSE or the CIM_Fan.EnabledState property has a value of 3 (Disabled), the CIM_Fan.SetSpeed() method shall return a value of 1 (Not Supported).

- 420 • OCL constraint in the context of a Fan instance:

421

```

post:
self.DesiredSpeed = self.SetSpeed.Speed

```

422 Explanation:

423 When the CIM_Fan.SetSpeed() method successfully executed, the DesiredSpeed property shall be the value of the Speed parameter of the SetSpeed() method.

424 **Table 15 – SetSpeed(): Parameter requirements**

Parameter	Description
DesiredSpeed	In. The parameter shall be specified in a unit of RPMs (revolutions per minute)
Return value	See 7.2.5.8.2

426

7.2.5.8.1 Parameter: DesiredSpeed

427 The parameter shall be specified in a unit of RPMs (revolutions per minute)

7.2.5.8.2 Return value

429 This method shall return one of the following return values:

Table 16 – SetSpeed: Return values

Value	Description
0	The speed change was successfully performed.
1	The method is not implemented.
2	An error has occurred. Note, the meaning of this value differs from the definition in the CIM schema.

7.2.5.9 Operation: ModifyInstance()

432 **Requirement level:** Conditional

433 **Condition:**

The FanElementNameModification feature is implemented.

7.2.6 Adaptation: FanCapabilities: CIM_EnabledLogicalElementCapabilities

7.2.6.1 General

437 **Adaptation type:** Ordinary class

438 **Implementation type:** Instantiated

439 **Requirement level:** Conditional

440 **Condition:**

The FanCapabilities feature is implemented.

441 This adaptation models the capabilities of fans modeled with Fan .

Table 17 – FanCapabilities: Element requirements

Element	Requirement	Description
Properties		
InstanceID	Mandatory	Key
RequestedStatesSupported	Mandatory	See 7.2.6.2
ElementNameEditSupported	Mandatory	See 7.2.6.3
MaxElementNameLen	Conditional	See 7.2.6.4
Operations		
GetInstance()	Mandatory	
EnumerateInstances()	Mandatory	
EnumerateInstanceNames()	Mandatory	
Associators()	Mandatory	
AssociatorNames()	Mandatory	

Element	Requirement	Description
References()	Mandatory	
ReferenceNames()	Mandatory	

443 7.2.6.2 Property: RequestedStatesSupported

444 **Requirement level:** Mandatory

445 **Constraint:**

446 OCL constraint in the context of a FanCapabilities instance:

```
447 inv:
if mrpIsFeatureSupported('fan ElementName modification',
    self.cIM_ElementCapabilities.Element)
then self.RequestedStatesSupported =
    Set { 2 /* Enabled */, 3 /* Disabled */, 11 /* Reset */ } )
else self.RequestedStatesSupported->isEmpty() )
```

448 **Explanation:**

449 If feature 'fan ElementName modification' is supported, the RequestedStatesSupported array property shall contain any combination of the values: 2 (Enabled), 3 (Disabled), 11 (Reset). If feature 'fan ElementName modification' is not supported, the RequestedStatesSupported property shall be an empty array.

450 7.2.6.3 Property: ElementNameEditSupported

451 **Requirement level:** Mandatory

452 **Constraint:**

453 OCL constraint in the context of a FanCapabilities instance:

```
454 inv:
mrpIsFeatureSupported('fan ElementName modification',
    self.cIM_ElementCapabilities.Element)
implies
    self.ElementNameEditSupported = True
```

455 **Explanation:**

456 If client modification of the CIM_Fan.ElementName property is supported, the ElementNameEditSupported property shall have a value of True.

457 7.2.6.4 Property: MaxElementNameLen

458 **Requirement level:** Conditional

459 **Condition:**

The FanElementNameModification feature is implemented.

461 7.2.7 Adaptation: ElementCapabilities: CIM_ElementCapabilities

462 7.2.7.1 General

463 **Adaptation type:** Association class

464

Implementation type: Instantiated

465 **Requirement level:** Conditional

466 **Condition:**

The FanCapabilities feature is implemented.

468 This adaptation does not define a model description.

469 **Table 18 – ElementCapabilities: Element requirements**

Element	Requirement	Description
Properties		
ManagedElement	Mandatory	Key, see 7.2.7.2
Capabilities	Mandatory	Key, see 7.2.7.3
Operations		
GetInstance()	Mandatory	

470 **7.2.7.2 Property: ManagedElement**

471 **Requirement level:** Mandatory

472 **Reference kind:** REF-typed

473 Shall reference an instance of CIM_Fan representing the fan.

474 **Constraints:**

- 475 • Referenced instances shall be of class adaptation Fan.
- 476 • The multiplicity of this association end is 1 .. ***Profile Error: Reference "ManagedElement" defined in association class adaptation "ElementCapabilities" constrains its maximum multiplicity to "unbounded", which is invalid because it is greater than the maximum multiplicity "1" defined in schema association class "CIM_ElementCapabilities".**

477 **7.2.7.3 Property: Capabilities**

478 **Requirement level:** Mandatory

479 **Reference kind:** REF-typed

480 Shall reference an instance of CIM_EnabledLogicalElementCapabilities that describes the capabilities of CIM_Fan.

481 **Constraints:**

- 482 • Referenced instances shall be of class adaptation FanCapabilities.
- 483 • The multiplicity of this association end is 0 .. 1

484 **7.2.8 Adaptation: CooledElement: CIM_ManagedSystemElement**

485 This adaptation does not define a model description.

486 **Adaptation type:** Ordinary class

487 **Implementation type:** Instantiated

488 A concrete subclass of the abstract schema class CIM_ManagedSystemElement needs to be implemented.

489

Requirement level: Conditional

490 **Condition:**

The PartialCooling feature is implemented.

492 **Table 19 – CooledElement: Element requirements**

Element	Requirement	Description
Operations		
Associators()	Mandatory	
AssociatorNames()	Mandatory	
References()	Mandatory	
ReferenceNames()	Mandatory	

493 **7.2.9 Adaptation: AssociatedCooling: CIM_AssociatedCooling**

494 **7.2.9.1 General**

495 **Adaptation type:** Association class

496 **Implementation type:** Instantiated

497 **Requirement level:** Conditional

498 **Condition:**

The PartialCooling feature is implemented.

500 This adaptation does not define a model description.

501 The managed system element for which the fan provides cooling is represented by the CIM_Fan instance that is associated with a CIM_ManagedSystemElement subclass instance through the CIM_AssociatedCooling association. When no instance of CIM_AssociatedCooling references the instance of CIM_Fan, the fan represented by CIM_Fan cools the whole managed system represented by the CIM_System instance associated with the CIM_Fan instance via CIM_SystemDevice. When at least one instance of CIM_AssociatedCooling references the instance of CIM_Fan, the elements cooled by the fan shall be only those referenced by the CIM_AssociatedCooling association and not the whole managed system.

502 **Table 20 – AssociatedCooling: Element requirements**

Element	Requirement	Description
Properties		
Antecedent	Mandatory	Key, see 7.2.9.2
Dependent	Mandatory	Key, see 7.2.9.3
Operations		
GetInstance()	Mandatory	

503 **7.2.9.2 Property: Antecedent**

504 **Requirement level:** Mandatory

505 **Reference kind:** REF-typed

506 Shall reference an instance of CIM_Fan representing the fan.

507

Constraints:

- 508 • Referenced instances shall be of class adaptation Fan.
- 509 • The multiplicity of this association end is 1 .. *

7.2.9.3 Property: Dependent

Requirement level: Mandatory

Reference kind: REF-typed

Shall reference an instance of a subclass of CIM_ManagedSystemElement for which the fan is providing cooling.

Constraints:

- 515 • Referenced instances shall be of class adaptation CooledElement.
- 516 • The multiplicity of this association end is 1 .. *

7.2.10 Adaptation: FanRedundancySet: CIM_RedundancySet

7.2.10.1 General

Adaptation type: Ordinary class

Implementation type: Instantiated

Requirement level: Conditional

Condition:

At least one of the following is true:

- 523 • The FanRedundancyBySparing feature is implemented.
- 524 • The FanRedundancyByBalancing feature is implemented.

This adaptation models fan redundancy groups for which the feature 'fan redundancy' is implemented.

Table 21 – FanRedundancySet: Element requirements

Element	Requirement	Description
Properties		
InstanceID	Mandatory	Key
RedundancyStatus	Mandatory	
TypeOfSet	Mandatory	See 7.2.10.2
MinNumberNeeded	Mandatory	See 7.2.10.3
ElementName	Mandatory	See 7.2.10.4
Methods		
Failover()	Optional	See 7.2.10.5
Operations		
GetInstance()	Mandatory	
EnumerateInstances()	Mandatory	
EnumerateInstanceNames()	Mandatory	
Associators()	Mandatory	

Element	Requirement	Description
AssociatorNames()	Mandatory	
References()	Mandatory	
ReferenceNames()	Mandatory	

528

7.2.10.2 Property: TypeOfSet

529

Requirement level: Mandatory

530

Constraints:

532

- OCL constraint in the context of a FanRedundancySet instance:

533

```

inv:
mrpIsFeatureSupported('fan redundancy type \'load-balanced\'',self)
implies
    self.TypeOfSet->forall( v | Set { 3 /* Load Balanced */,
        2 /* N+1 */ }->includes(v))

```

534

Explanation:

535

If feature 'fan redundancy type "load-balanced"' is supported for a fan redundancy group represented by a CIM_RedundancySet instance, its TypeOfSet array property shall contain the values 3 (Load Balanced), 2 (N+1), or both, and shall not contain any other values.

537

- OCL constraint in the context of a FanRedundancySet instance:

538

```

inv:
mrpIsFeatureSupported('fan redundancy type \'sparing\'',self)
implies
    self.TypeOfSet->forall( v | Set { 4 /* Sparing */,
        5 /* Limited Sparing */ }->includes(v))

```

539

Explanation:

540

If feature 'fan redundancy type "sparing"' is supported for a fan redundancy group represented by a CIM_RedundancySet instance, its TypeOfSet array property shall contain the values 4 (Sparing), 5 (Limited Sparing), or both, and shall not contain any other values.

541

7.2.10.3 Property: MinNumberNeeded

542

Requirement level: Mandatory

543

Shall match 0 when the minimum number of fans needed for the redundancy is unknown.

544

7.2.10.4 Property: ElementName

545

Requirement level: Mandatory

546

Shall be formatted as a free-form string of variable length, using the pattern ".*".

547

7.2.10.5 Method: Failover()

548

Requirement level: Optional

549

The Failover() method forces a failover from one member of a CIM_RedundancySet collection to another. After the successful execution of the method, the fan that is represented by the CIM_Fan instance

referenced by the FailoverFrom parameter becomes inactive. The fan that is represented by the CIM_Fan instance referenced by the FailoverTo parameter takes over as the active fan.

550 The Failover() method may be supported if the FailoverSupported property of at least one instance of CIM_IsSpare that references the CIM_RedundancySet instance has a value of 3 (Manual) or 4 (Both Manual and Automatic). The Failover() method shall not be supported if the FailoverSupported property of every instance of CIM_IsSpare that references the CIM_RedundancySet has a value of 2 (Automatic).

551 The execution of the Failover() method shall return a value of 2 (Error Occurred) under the following conditions:

- 552 • The CIM_Fan instance that is referenced by the FailoverTo parameter is not a spare fan .
- 553 • The CIM_Fan instance that is referenced by the FailoverFrom parameter is not associated with the CIM_RedundancySet only through an instance of CIM_MemberOfCollection.

554 After the successful execution of the Failover() method, the following actions occur:

- 555 • The CIM_Fan that is referenced by the FailoverTo parameter shall take over as the active fan.
- 556 • The CIM_Fan instance that is referenced by the FailoverTo parameter shall be associated with the CIM_RedundancySet only through an instance of CIM_MemberOfCollection.
- 557 • The CIM_Fan instance that is referenced by FailoverFrom parameter shall become a spare fan .
- 558 • When fan state management is supported, the CIM_Fan instance that is referenced by the FailoverFrom parameter shall not have an EnabledState property value of 2 (Enabled).

559 **Table 22 – Failover(): Parameter requirements**

Parameter	Description
FailoverFrom	In, see 7.2.10.5.1
FailoverTo	In, see 7.2.10.5.2
Return value	See 7.2.10.5.3

560 **7.2.10.5.1 Parameter: FailoverFrom**

561 The redundant fan that will become inactive.

562 **Constraint:**

563 Referenced instances shall be of class adaptation Fan.

564 **7.2.10.5.2 Parameter: FailoverTo**

565 The redundant fan that will become active and take over for the inactivated fan.

566 **Constraint:**

567 Referenced instances shall be of class adaptation Fan.

568 **7.2.10.5.3 Return value**

569 This method shall return one of the following return values:

570 **Table 23 – Failover: Return values**

Value	Description
0	The failover was successfully performed.
1	The method is not implemented.

Value	Description
2	An error has occurred.

571 **7.2.11 Adaptation: OwingCollectionElement: CIM_OwingCollectionElement**

572 **7.2.11.1 General**

573 **Adaptation type:** Association class

574 **Implementation type:** Instantiated

575 **Requirement level:** Conditional

576 **Condition:**

At least one of the following is true:

- 577 • The FanRedundancyBySparing feature is implemented.
- 578 • The FanRedundancyByBalancing feature is implemented.

580 This adaptation does not define a model description.

581 **Table 24 – OwingCollectionElement: Element requirements**

Element	Requirement	Description
Properties		
OwingElement	Mandatory	Key, see 7.2.11.2
OwnedElement	Mandatory	Key, see 7.2.11.3
Operations		
GetInstance()	Mandatory	

582 **7.2.11.2 Property: OwingElement**

583 **Requirement level:** Mandatory

584 **Reference kind:** REF-typed

585 **Constraints:**

- 586 • Referenced instances shall be of class adaptation ComputerSystem.
- 587 • The multiplicity of this association end is 1 .. 1

588 **7.2.11.3 Property: OwnedElement**

589 **Requirement level:** Mandatory

590 **Reference kind:** REF-typed

591 **Constraints:**

- 592 • Referenced instances shall be of class adaptation FanRedundancySet.
- 593 • The multiplicity of this association end is 0 .. *

594

7.2.12 Adaptation: HostedRedundancySet: CIM_HostedCollection

595 **7.2.12.1 General**

596 **Adaptation type:** Association class

597 **Implementation type:** Instantiated

598 **Requirement level:** Conditional

599 **Condition:**

At least one of the following is true:

- 600 • The FanRedundancyBySparing feature is implemented.
- 601 • The FanRedundancyByBalancing feature is implemented.

603 This adaptation does not define a model description.

604 **Table 25 – HostedRedundancySet: Element requirements**

Element	Requirement	Description
Properties		
Antecedent	Mandatory	Key, see 7.2.12.2
Dependent	Mandatory	Key, see 7.2.12.3
Operations		
GetInstance()	Mandatory	

605 **7.2.12.2 Property: Antecedent**

606 **Requirement level:** Mandatory

607 **Reference kind:** REF-typed

608 **Constraints:**

- 609 • Referenced instances shall be of class adaptation ComputerSystem.
- 610 • The multiplicity of this association end is 1 .. 1

611 **7.2.12.3 Property: Dependent**

612 **Requirement level:** Mandatory

613 **Reference kind:** REF-typed

614 **Constraints:**

- 615 • Referenced instances shall be of class adaptation FanRedundancySet.
- 616 • The multiplicity of this association end is 0 .. *

617 **7.2.13 Adaptation: MemberOfRedundancySet: CIM_MemberOfCollection**

618 **7.2.13.1 General**

619 **Adaptation type:** Association class

620 **Implementation type:** Instantiated

621

Requirement level: Conditional

622 **Condition:**

At least one of the following is true:

- 623 • The FanRedundancyBySparing feature is implemented.
- 624 • The FanRedundancyByBalancing feature is implemented.

626 This adaptation does not define a model description.

627 **Table 26 – MemberOfRedundancySet: Element requirements**

Element	Requirement	Description
Properties		
Collection	Mandatory	Key, see 7.2.13.2
Member	Mandatory	Key, see 7.2.13.3
Operations		
GetInstance()	Mandatory	

628 **7.2.13.2 Property: Collection**

629 **Requirement level:** Mandatory

630 **Reference kind:** REF-typed

631 **Constraints:**

- 632 • Referenced instances shall be of class adaptation FanRedundancySet.
- 633 • The multiplicity of this association end is 0 .. *

634 **7.2.13.3 Property: Member**

635 **Requirement level:** Mandatory

636 **Reference kind:** REF-typed

637 **Constraints:**

- 638 • Referenced instances shall be of class adaptation Fan.
- 639 • The multiplicity of this association end is 1 .. *

640 **7.2.14 Adaptation: IsSpare: CIM_IsSpare**

641 **7.2.14.1 General**

642 **Adaptation type:** Association class

643 **Implementation type:** Instantiated

644 **Requirement level:** Conditional

645 **Condition:**

The FanRedundancyBySparing feature is implemented.

647 This adaptation models the relationship between spare fans and their redundancy group.

648 **Constraints:**

650 • OCL constraint in the context of a IsSpare instance:
 651 `inv:`
`self.Antecedent.EnabledState = 3 /* Disabled */`
`implies`
`self.SpareStatus = 3 /* Cold Standby */`

652 Explanation:
 653 If the CIM_Fan instance (representing the spare fan) referenced from this association instance has an EnabledState property value of 3 (Disabled), the value of the referencing CIM_IsSpare instance's SpareStatus property shall be 3 (Cold Standby).

655 • OCL constraint in the context of a IsSpare instance:
 656 `inv:`
`self.Antecedent.EnabledState != 3 /* Disabled */`
`implies`
`self.SpareStatus = 0 /* Unknown */`

657 Explanation:
 658 If the CIM_Fan instance (representing the spare fan) referenced from this association instance has an EnabledState property value other than 3 (Disabled), the value of the referencing CIM_IsSpare instance's SpareStatus property shall be 0 (Unknown).

659 **Table 27 – IsSpare: Element requirements**

Element	Requirement	Description
Properties		
Antecedent	Mandatory	Key, see 7.2.14.2
Dependent	Mandatory	Key, see 7.2.14.3
Operations		
GetInstance()	Mandatory	

660 **7.2.14.2 Property: Antecedent**

661 **Requirement level:** Mandatory

662 **Reference kind:** REF-typed

663 **Constraints:**

- 664 • Referenced instances shall be of class adaptation Fan.
- 665 • The multiplicity of this association end is 1 .. *

666 **7.2.14.3 Property: Dependent**

667 **Requirement level:** Mandatory

668 **Reference kind:** REF-typed

669 **Constraints:**

- 670 • Referenced instances shall be of class adaptation FanRedundancySet.
- 671 • The multiplicity of this association end is 0 .. *

672

7.2.15 Adaptation: NumericFanSpeedSensor: CIM_NumericSensor

673 7.2.15.1 General

674 **Adaptation type:** Ordinary class

675 **Implementation type:** Instantiated

676 **Requirement level:** Conditional

677 **Condition:**

The FanSpeedSensor feature is implemented.

678 **Design Note:** DSP1013 1.0 defines the requirement level as optional. However, clause 7.10 of DSP1013 1.0 states the condition, so this machine readable profile has consistently defined the requirement level to be conditional.

679 This adaptation models analog speed sensors .

680 **Table 28 – NumericFanSpeedSensor: Element requirements**

Element	Requirement	Description
Base adaptations		
SpeedSensors::NumericSensor	Optional	See SpeedSensors::NumericSensor.
Properties		
SensorType	Mandatory	See 7.2.15.2
BaseUnits	Mandatory	See 7.2.15.3
RateUnits	Mandatory	See 7.2.15.4
CurrentReading	Mandatory	

681 7.2.15.2 Property: SensorType

682 **Requirement level:** Mandatory

683 **Constraint:**

684 OCL constraint in the context of a NumericFanSpeedSensor instance:

```
685 inv:
self.SensorType = 5 /* Tachometer */
```

686 **Explanation:**

The value of the SensorType property shall be 5 (Tachometer).

687 7.2.15.3 Property: BaseUnits

688 **Requirement level:** Mandatory

689 **Constraint:**

690 OCL constraint in the context of a NumericFanSpeedSensor instance:

```
691 inv:
self.BaseUnits = 19 /* RPM */
```

692 **Explanation:**

The value of the BaseUnits property shall be 19 (RPM).

693 **7.2.15.4 Property: RateUnits**

694 **Requirement level:** Mandatory

695 **Constraint:**

696 OCL constraint in the context of a NumericFanSpeedSensor instance:

```
697 inv:
    self.RateUnits = 0 /* None */
```

698 **Explanation:**

The value of the RateUnits property shall be 0 (None).

699 **7.2.16 Adaptation: DiscreteFanSpeedSensor: CIM_Sensor**

700 **7.2.16.1 General**

701 **Adaptation type:** Ordinary class

702 **Implementation type:** Instantiated

703 **Requirement level:** Conditional

704 **Condition:**

The FanSpeedSensor feature is implemented.

705 **Design Note:** DSP1013 1.0 defines the requirement level as optional. However, clause 7.10 of DSP1013 1.0 states the condition, so this machine readable profile has consistently defined the requirement level to be conditional.

706 This adaptation models discrete fan speed sensors .

707 **Table 29 – DiscreteFanSpeedSensor: Element requirements**

Element	Requirement	Description
Base adaptations		
SpeedSensors::DiscreteSensor	Optional	See SpeedSensors::DiscreteSensor.
Properties		
SensorType	Mandatory	See 7.2.16.2

708 **7.2.16.2 Property: SensorType**

709 **Requirement level:** Mandatory

710 **Constraint:**

711 OCL constraint in the context of a DiscreteFanSpeedSensor instance:

```
712 inv:
    self.SensorType = 5 /* Tachometer */
```

713 **Explanation:**

The value of the SensorType property shall be 5 (Tachometer).

714

7.2.17 Adaptation: FanAddedIndication: CIM_InstCreation

715 This adaptation does not define a model description.

716 **Design Note:** This adaptation has been defined to demonstrate the definition of lifecycle indications; it is not part of DSP1013 1.0.

717 **Adaptation type:** Indication class

718 **Implementation type:** Indication

719 **Requirement level:** Conditional

720 **Condition:**

The FanIndications feature is implemented.

722 **Table 30 – FanAddedIndication: Element requirements**

Element	Requirement	Description
Base adaptations		
Indications::LifecycleIndication	Mandatory	See Indications::LifecycleIndication.

7.2.18 Adaptation: FanRemovedIndication: CIM_InstDeletion

724 This adaptation does not define a model description.

725 **Design Note:** This adaptation has been defined to demonstrate the definition of lifecycle indications; it is not part of DSP1013 1.0.

726 **Adaptation type:** Indication class

727 **Implementation type:** Indication

728 **Requirement level:** Conditional

729 **Condition:**

The FanIndications feature is implemented.

731 **Table 31 – FanRemovedIndication: Element requirements**

Element	Requirement	Description
Base adaptations		
Indications::LifecycleIndication	Mandatory	See Indications::LifecycleIndication.

7.2.19 Adaptation: FanHealthIndication: CIM_AlertIndication

This adaptation models alert indications for reporting the health state of fans.

733 **Design Note:** This adaptation has been defined to demonstrate the definition of alert indications; it is not part of DSP1013 1.0.

734 Indications related to the redundancy of fans are defined in the FanRedundancyIndication adaptation.

735 **Adaptation type:** Indication class

736 **Implementation type:** Indication

737 **Requirement level:** Conditional

738 **Condition:**

The FanIndications feature is implemented.

740

Table 32 – FanHealthIndication: Element requirements

Element	Requirement	Description
Base adaptations		
Indications::AlertIndication	Mandatory	See Indications::AlertIndication.
Alert messages		
Platform::PLAT0458	Mandatory	Fan failed
Platform::PLAT0459	Mandatory	Fan return to OK
Platform::PLAT0460	Mandatory	Fan degraded
Platform::PLAT0462	Mandatory	Fan speed high
Platform::PLAT0463	Mandatory	Fan speed normal

741

7.2.20 Adaptation: FanRedundancyIndication: CIM_AlertIndication

742

This adaptation models alert indications related to the redundancy of fans.

743

Design Note: This adaptation has been defined to demonstrate the definition of alert indications; it is not part of DSP1013 1.0.

744

Adaptation type: Indication class

745

Implementation type: Indication

746

Requirement level: Conditional

747

Condition:

All of the following is true:

748

- The FanIndications feature is implemented.

749

- At least one of the following is true:

750

- The FanRedundancyByBalancing feature is implemented.

751

- The FanRedundancyBySparing feature is implemented.

753

Table 33 – FanRedundancyIndication: Element requirements

Element	Requirement	Description
Base adaptations		
Indications::AlertIndication	Mandatory	See Indications::AlertIndication.
Alert messages		
Platform::PLAT0452	Mandatory	Fan Redundacy Lost (sufficient)
Platform::PLAT0454	Mandatory	Fan Redundancy Lost (insufficient)
Platform::PLAT0455	Mandatory	Fan Redundancy Restored

754

7.2.21 Adaptation: FanSpeedAlertIndicationFilter: CIM_IndicationFilter

755

7.2.21.1 General

756

Adaptation type: Ordinary class

757

Implementation type: Instantiated

758

Requirement level: Conditional

759 **Condition:**

The FanIndications feature is implemented.

761 This adaptation does not define a model description.

762 A static filter for the fan speed related alert indications.

763 **Design Note:** This adaptation has been defined to demonstrate the definition of an indication specific (i.e. static) indication filter for certain alert indications; it is not part of DSP1013 1.0.

764 **Table 34 – FanSpeedAlertIndicationFilter: Element requirements**

Element	Requirement	Description
Base adaptations		
Indications::IndicationSpecificIndicationFilter	Mandatory	See Indications::IndicationSpecificIndicationFilter.
Properties		
Name	Mandatory	Key, see 7.2.21.2
Query	Mandatory	Required, see 7.2.21.3
QueryLanguage	Mandatory	Required, see 7.2.21.4

765 **7.2.21.2 Property: Name**

766 **Requirement level:** Mandatory

767 **Constraint:**

768 OCL constraint in the context of a FanSpeedAlertIndicationFilter instance:

769 `inv: self.Name = 'DMTF:Fan:SpeedAlertIndicationFilter'`

770 **7.2.21.3 Property: Query**

771 **Requirement level:** Mandatory

772 **Constraint:**

773 OCL constraint in the context of a FanSpeedAlertIndicationFilter instance:

774 `inv: self.Query = 'SELECT * FROM CIM_AlertIndication WHERE MessageID IN (`
 775 `"PLAT0462", "PLAT0463")'`

775 **7.2.21.4 Property: QueryLanguage**

776 **Requirement level:** Mandatory

777 **Constraint:**

778 OCL constraint in the context of a FanSpeedAlertIndicationFilter instance:

779 `inv: self.QueryLanguage = 'DMTF:CQL'`

780 **7.2.22 Adaptation: FanAddedLifecycleIndicationFilter: CIM_IndicationFilter**

781 **7.2.22.1 General**

782 **Adaptation type:** Ordinary class

783

Implementation type: Instantiated

784 **Requirement level:** Conditional

785 **Condition:**

The FanIndications feature is implemented.

787 This adaptation does not define a model description.

788 A static filter for the fan added lifecycle indication.

789 **Design Note:** This adaptation has been defined to demonstrate the definition of an indication specific (i.e. static) indication filter for a single lifecycle indication; it is not part of DSP1013 1.0.

790 **Table 35 – FanAddedLifecycleIndicationFilter: Element requirements**

Element	Requirement	Description
Base adaptations		
Indications::IndicationSpecificIndicationFilter	Mandatory	See Indications::IndicationSpecificIndicationFilter.
Properties		
Name	Mandatory	Key, see 7.2.22.2
Query	Mandatory	Required, see 7.2.22.3
QueryLanguage	Mandatory	Required, see 7.2.22.4

791 **7.2.22.2 Property: Name**

792 **Requirement level:** Mandatory

793 **Constraint:**

794 OCL constraint in the context of a FanAddedLifecycleIndicationFilter instance:

795 `inv: self.Name = 'DMTF:Fan:AddedLifecycleIndicationFilter'`

796 **7.2.22.3 Property: Query**

797 **Requirement level:** Mandatory

798 **Constraint:**

799 OCL constraint in the context of a FanAddedLifecycleIndicationFilter instance:

800 `inv: self.Query = 'SELECT * FROM CIM_InstCreation WHERE SourceInstance ISA CIM_Fan'`

801 **7.2.22.4 Property: QueryLanguage**

802 **Requirement level:** Mandatory

803 **Constraint:**

804 OCL constraint in the context of a FanAddedLifecycleIndicationFilter instance:

805 `inv: self.QueryLanguage = 'DMTF:CQL'`

806

8 Use cases and state descriptions

807

8.1 State description: ObjectDiagram

808

Section 9.1 of Fan Profile (DSP1013) would be inserted here.

809

8.2 Use case: SetFanSpeed

810

Section 9.2 of Fan Profile (DSP1013) would be inserted here.

811

The main flow for this use case consists of the following step:

812

1. Steps from Section 9.2

813

8.3 Use case: ResetFan

814

Section 9.3 of Fan Profile (DSP1013) would be inserted here.

815

The main flow for this use case consists of the following step:

816

1. Steps from Section 9.3

817

8.4 Use case: GetFanRedundancyStatus

818

Section 9.4 of Fan Profile (DSP1013) would be inserted here.

819

The main flow for this use case consists of the following step:

820

1. Steps from Section 9.4

821

8.5 Use case: FindSpareFan

822

Section 9.5 of Fan Profile (DSP1013) would be inserted here.

823

The main flow for this use case consists of the following step:

824

1. Steps from Section 9.5

825

8.6 Use case: ShowFanSensorInfo

826

Section 9.6 of Fan Profile (DSP1013) would be inserted here.

827

The main flow for this use case consists of the following step:

828

1. Steps from Section 9.6

829

8.7 Use case: FindCooledElements

830

Section 9.7 of Fan Profile (DSP1013) would be inserted here.

831

The main flow for this use case consists of the following step:

832

1. Steps from Section 9.7

833

8.8 Use case: DetermineElementNameModifiability

834

Section 9.8 of Fan Profile (DSP1013) would be inserted here.

835

The main flow for this use case consists of the following step:

836
837

1. Steps from Section 9.8

ANNEX A (informative)

Change log

838

Version	Date	Description
1.0.0a	2006-06-13	DSP1013: Released as a Preliminary Standard
1.0.0	2007-10-12	DSP1013: Released as a Final Standard
1.0.1	2008-09-23	DSP1013: Released as a Final Standard
1.0.2m	2011-08-31	XMP1013: Included as a sample profile into DSP2023

839

Bibliography

This clause lists references that are helpful for the application of this document.

- 840 DMTF DSP1000, *Management Profile Specification Template 1.1*,
http://www.dmtf.org/standards/published_documents/DSP1000_1.1.pdf