



1

2

3

4

Document Number: DSP1013

Date: 2008-09-29

Version: 1.0.1

5 **Fan Profile**

6 **Document Type: Specification**

7 **Document Status: Final Standard**

8 **Document Language: E**

9 Copyright Notice

10 Copyright © 2008 Distributed Management Task Force, Inc. (DMTF). All rights reserved.

11 DMTF is a not-for-profit association of industry members dedicated to promoting enterprise and systems
12 management and interoperability. Members and non-members may reproduce DMTF specifications and
13 documents for uses consistent with this purpose, provided that correct attribution is given. As DMTF
14 specifications may be revised from time to time, the particular version and release date should always be
15 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26 withdrawn or modified after publication, and shall be indemnified and held harmless by any party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 the DMTF that, in their opinion,
30 such patent may relate to or impact implementations of DMTF standards, visit
31 <http://www.dmtf.org/about/policies/disclosures.php>.

CONTENTS

33	Foreword	6
34	Introduction	7
35	1 Scope	9
36	2 Normative References.....	9
37	2.1 Approved References	9
38	2.2 Other References.....	9
39	3 Terms and Definitions	9
40	4 Symbols and Abbreviated Terms	11
41	5 Synopsis	11
42	6 Description	11
43	6.1 Fan Redundancy.....	12
44	7 Implementation Requirements	13
45	7.1 CIM_Fan	13
46	7.2 CIM_Fan.DesiredSpeed	13
47	7.3 CIM_EnabledLogicalElementCapabilities.....	13
48	7.4 Fan State Management	13
49	7.5 CIM_Fan.RequestedState	14
50	7.6 CIM_Fan.EnabledState	14
51	7.7 CIM_SystemDevice and CIM_AssociatedCooling.....	15
52	7.8 Modeling Fan Redundancy.....	15
53	7.9 CIM_Fan.ElementName	16
54	7.10 Modeling Fan Speed Sensor	16
55	8 Methods.....	16
56	8.1 CIM_Fan.SetSpeed()	16
57	8.2 CIM_Fan.RequestStateChange()	17
58	8.3 CIM_RedundancySet.Failover()	18
59	8.4 Profile Conventions for Operations.....	19
60	8.5 CIM_AssociatedCooling	19
61	8.6 CIM_ElementCapabilities	20
62	8.7 CIM_EnabledLogicalElementCapabilities.....	20
63	8.8 CIM_Fan	20
64	8.9 CIM_HostedCollection	21
65	8.10 CIM_IsSpare	21
66	8.11 CIM_MemberOfCollection	21
67	8.12 CIM_NumericSensor.....	21
68	8.13 CIM_OwningCollectionElement.....	22
69	8.14 CIM_RedundancySet.....	22
70	8.15 CIM_Sensor	22
71	8.16 CIM_SystemDevice	22
72	9 Use Cases.....	22
73	9.1 Object Diagrams	22
74	9.2 Set the Fan Speed	25
75	9.3 Reset the Fan	26
76	9.4 Retrieve the Fan Redundancy Status.....	26
77	9.5 Find the Spare Fan Supply for the Given Active Fan	26
78	9.6 Show the Fan Sensor Information	26
79	9.7 Find the Managed System Elements Cooled by the Fan	27
80	9.8 Determine Whether the CIM_Fan.ElementName Is Modifiable	27
81	10 CIM Elements.....	27
82	10.1 CIM_AssociatedCooling	28
83	10.2 CIM_ElementCapabilities	28
84	10.3 CIM_EnabledLogicalElementCapabilities.....	28

85	10.4	CIM_Fan	29
86	10.5	CIM_HostedCollection	29
87	10.6	CIM_IsSpare	30
88	10.7	CIM_MemberOfCollection	30
89	10.8	CIM_NumericSensor.....	30
90	10.9	CIM_OwningCollectionElement	31
91	10.10	CIM_RedundancySet.....	31
92	10.11	CIM_RegisteredProfile.....	31
93	10.12	CIM_Sensor	32
94	10.13	CIM_SystemDevice	32
95	ANNEX A (informative) Change Log.....		33
96	ANNEX B (informative) Acknowledgments.....		34

97

98 Figures

99	Figure 1 – Fan Profile: Class Diagram.....	12
100	Figure 2 – Fan Profile: Object Diagram	23
101	Figure 3 – Fan Profile: Fan Redundancy before Failover() Method Execution	24
102	Figure 4 – Fan Profile: Fan Redundancy after Failover() Method Execution	24
103	Figure 5 – Fan Profile: Fan Redundancy	25
104	Figure 6 – Fan Profile: After the SetSpeed() Method Execution.....	26

105

106 Tables

107	Table 1 – Related Profiles.....	11
108	Table 2 – EnabledState Value Description	14
109	Table 3 – CIM_Fan.SetSpeed() Method: Return Code Values	17
110	Table 4 – CIM_Fan.SetSpeed() Method: Parameters	17
111	Table 5 – CIM_Fan.RequestStateChange() Method: Return Code Values.....	17
112	Table 6 – CIM_Fan.RequestStateChange() Method: Parameters.....	17
113	Table 7 – CIM_RedundancySet.Failover() Method: Return Code Values.....	18
114	Table 8 – CIM_RedundancySet.Failover() Method: Parameters.....	19
115	Table 9 – Operations: CIM_AssociatedCooling.....	19
116	Table 10 – Operations: CIM_ElementCapabilities	20
117	Table 11 – Operations: CIM_Fan.....	20
118	Table 12 – Operations: CIM_HostedCollection	21
119	Table 13 – Operations: CIM_IsSpare	21
120	Table 14 – Operations: CIM_MemberOfCollection.....	21
121	Table 15 – Operations: CIM_OwningCollectionElement	22
122	Table 16 – Operations: CIM_SystemDevice.....	22
123	Table 17 – CIM Elements: Fan Profile	27
124	Table 18 – Class: CIM_AssociatedCooling.....	28
125	Table 19 – Class: CIM_ElementCapabilities.....	28
126	Table 20 – Class: CIM_EnabledLogicalElementCapabilities.....	28
127	Table 21 – Class: CIM_Fan	29
128	Table 22 – Class: CIM_HostedCollection	29
129	Table 23 – Class: CIM_IsSpare	30
130	Table 24 – Class: CIM_MemberOfCollection.....	30

131 Table 25 – Class: CIM_NumericSensor..... 30

132 Table 26 – Class: CIM_OwningCollectionElement 31

133 Table 27 – Class: CIM_RedundancySet..... 31

134 Table 28 – Class: CIM_RegisteredProfile..... 31

135 Table 29 – Class: CIM_Sensor 32

136 Table 30 – Class: CIM_SystemDevice 32

137

138

Foreword

139 The *Fan Profile* (DSP1013) was prepared by the Server Management Working Group.

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

142

Introduction

143 This document defines the classes used to describe the fans and the possible redundancy of the fans in a
144 managed system. The document also defines association classes that describe the relationship of the fan
145 to the fan's physical aspects (such as FRU data) to the sensors monitoring the fans, to other cooling
146 devices, to redundancy status, and to DMTF profile version information. The information in this
147 specification is intended to be sufficient for a provider or consumer of this data to identify unambiguously
148 the classes, properties, methods, and values that are mandatory to be instantiated and manipulated to
149 represent and manage fans and redundant fans of managed systems and subsystems that are modeled
150 using the DMTF CIM core and extended model definitions.

151 The target audience for this specification is implementers who are writing CIM-based providers or
152 consumers of management interfaces that represent the component described in this document.

153

Fan Profile

154 1 Scope

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

160 2 Normative References

161 The following referenced documents are indispensable for the application of this document. For dated
162 references, only the edition cited applies. For undated references, the latest edition of the referenced
163 document (including any amendments) applies.

164 2.1 Approved References

165 DMTF [DSP0200](#), *CIM Operations over HTTP 1.2.0*

166 DMTF [DSP0004](#), *CIM Infrastructure Specification 2.3.0*

167 DMTF [DSP1000](#), *Management Profile Specification Template 1.0.0*

168 DMTF [DSP1001](#), *Management Profile Specification Usage Guide 1.0.0*

169 DMTF [DSP1011](#), *Physical Asset Profile 1.0.0*

170 DMTF [DSP1009](#), *Sensors Profile 1.0.0*

171 DMTF [DSP1033](#), *Profile Registration Profile 1.0.0*

172 2.2 Other References

173 ISO/IEC Directives, Part 2, [Rules for the structure and drafting of International Standards](#)

174 OMG, [Unified Modeling Language \(UML\) from the Open Management Group \(OMG\)](#)

175 3 Terms and Definitions

176 For the purposes of this document, the following terms and definitions apply.

177 3.1

178 **can**

179 used for statements of possibility and capability, whether material, physical, or causal

180 3.2

181 **cannot**

182 used for statements of possibility and capability, whether material, physical, or causal

- 183 **3.3**
184 **conditional**
185 indicates requirements to be followed strictly to conform to the document when the specified conditions
186 are met
- 187 **3.4**
188 **mandatory**
189 indicates requirements to be followed strictly to conform to the document and from which no deviation is
190 permitted
- 191 **3.5**
192 **may**
193 indicates a course of action permissible within the limits of the document
- 194 **3.6**
195 **need not**
196 indicates a course of action permissible within the limits of the document
- 197 **3.7**
198 **optional**
199 indicates a course of action permissible within the limits of the document
- 200 **3.8**
201 **redundant fan**
202 any fan participating in the redundancy set
- 203 **3.9**
204 **referencing profile**
205 indicates a profile that owns the definition of this class and can include a reference to this profile in its
206 "Related Profiles" table
- 207 **3.10**
208 **shall**
209 indicates requirements to be followed strictly to conform to the document and from which no deviation is
210 permitted
- 211 **3.11**
212 **shall not**
213 indicates requirements to be followed strictly to conform to the document and from which no deviation is
214 permitted
- 215 **3.12**
216 **should**
217 indicates that among several possibilities, one is recommended as particularly suitable, without
218 mentioning or excluding others, or that a certain course of action is preferred but not necessarily required
- 219 **3.13**
220 **should not**
221 indicates that a certain possibility or course of action is deprecated but not prohibited
- 222 **3.14**
223 **Spare Fan**
224 an instance of CIM_Fan that represents a spare fan in any condition

225 4 Symbols and Abbreviated Terms

226 None.

227 5 Synopsis

228 **Profile Name:** Fan

229 **Version:** 1.0.1

230 **Organization:** DMTF

231 **CIM Schema Version:** 2.19.1

232 **Central Class:** CIM_Fan

233 **Scoping Class:** CIM_ComputerSystem

234 The *Fan Profile* extends the management capability of the referencing profiles by adding the capability to
235 describe fans and redundant fans of managed systems.

236 Table 1 identifies profiles that are related to this.

237

Table 1 – Related Profiles

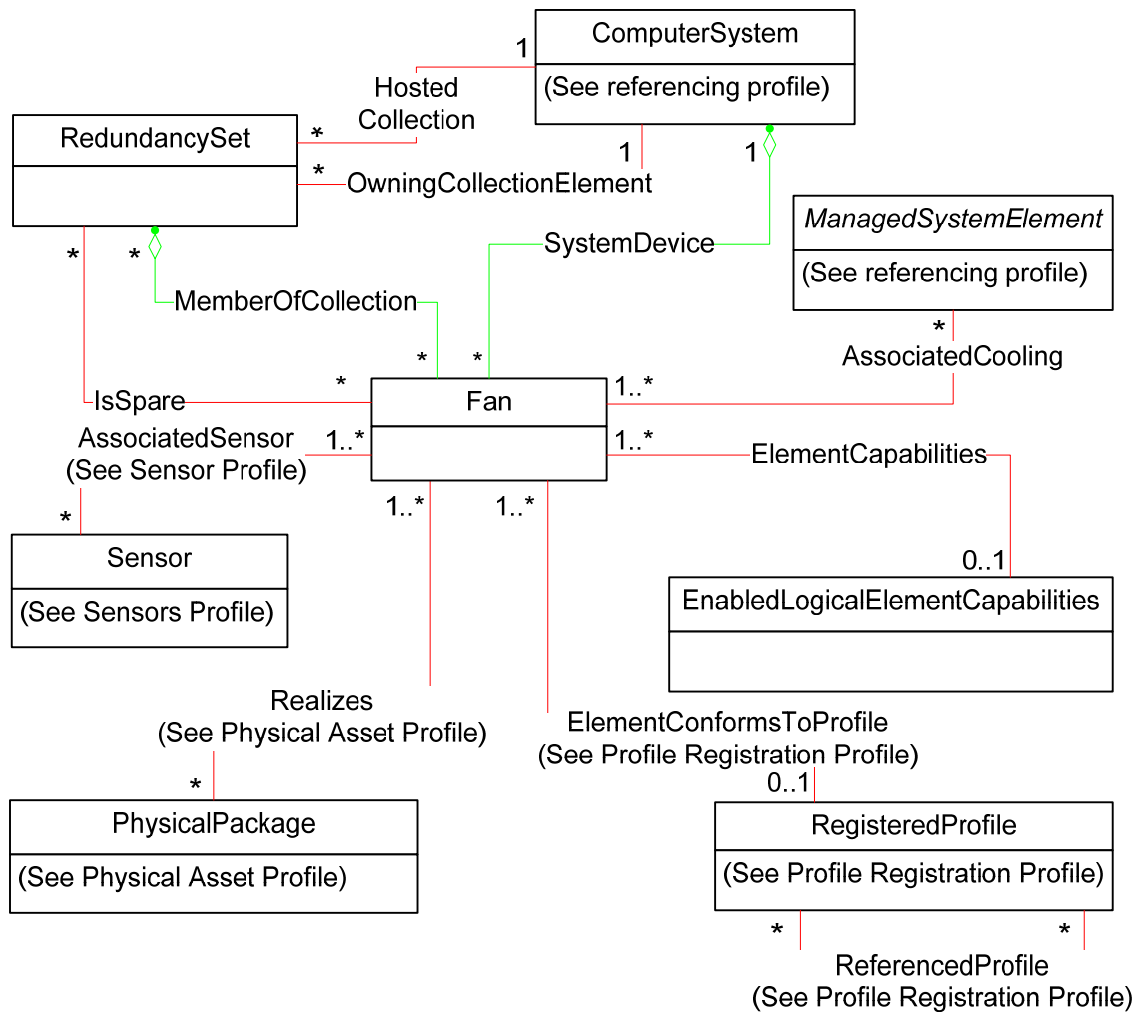
Profile Name	Organization	Version	Requirement	Description
Physical Asset	DMTF	1.0.0	Optional	
Sensors	DMTF	1.0.0	Optional	
Profile Registration	DMTF	1.0.0	Mandatory	

238 6 Description

239 The *Fan Profile* describes the properties and methods of fans and redundant fans in a managed system.

240 Figure 1 represents the class schema for the *Fan Profile*. For simplicity, the prefix CIM_ has been
241 removed from the names of the classes.

242 In the *Fan Profile*, the redundancy of fans is represented by a CIM_RedundancySet instance that
243 contains a collection of CIM_Fan instances (see section 7.8 for more details). When a fan has associated
244 sensors, a CIM_Fan instance is associated to a CIM_Sensor instance through the
245 CIM_AssociatedSensor class. The fan's physical description, such as FRU information, is represented by
246 an instance of the CIM_PhysicalPackage class through the CIM_Realizes association. The managed
247 system element for which the fan provides cooling is represented by the CIM_Fan instance that is
248 associated with a CIM_ManagedSystemElement subclass instance through the CIM_AssociatedCooling
249 association. When no CIM_AssociatedCooling association exists, the fan is providing cooling to the whole
250 system. The version of the *Fan Profile* implemented is represented through the CIM_RegisteredProfile
251 class.



252

253

Figure 1 – Fan Profile: Class Diagram

254 **6.1 Fan Redundancy**

255 An instance of CIM_RedundancySet represents the redundancy of fans in a managed system. Each
 256 instance of CIM_Fan that corresponds to a redundant fan is associated with the instance of
 257 CIM_RedundancySet through the CIM_MemberOfCollection association. The Spare Fans within the
 258 redundancy are associated with the CIM_RedundancySet instance through the CIM_IsSpare association.

259 **7 Implementation Requirements**

260 This section details the requirements related to the arrangement of instances and properties of instances
261 for implementations of this profile.

262 **7.1 CIM_Fan**

263 Zero or more instances of CIM_Fan shall be instantiated.

264 **7.2 CIM_Fan.DesiredSpeed**

265 When the CIM_Fan.SetSpeed() method is supported and successfully executed, the DesiredSpeed
266 property shall be set to the value of the Speed parameter of the SetSpeed() method.

267 When the SetSpeed() method is supported, the initial value of CIM_Fan.DesiredSpeed shall be 0,
268 meaning that no change has been requested.

269 When the SetSpeed() method is not supported, the value of CIM_Fan.DesiredSpeed property shall be 0,
270 meaning not applicable.

271 **7.3 CIM_EnabledLogicalElementCapabilities**

272 When the CIM_EnabledLogicalElementCapabilities class is instantiated, the instance of
273 CIM_EnabledLogicalElementCapabilities shall be associated to the CIM_Fan instance through the
274 CIM_ElementCapabilities association, and used for advertising the capabilities of the CIM_Fan instance.

275 At most one instance of CIM_EnabledLogicalElementCapabilities shall be associated with a given
276 instance of CIM_Fan.

277 **7.3.1 CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported**

278 CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported is an array that contains the
279 supported requested states for the instance of CIM_Fan. This property shall be the super set of the
280 values to be used for the RequestedState parameter in the RequestStateChange() method (see
281 section 8.2). The value of the CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported
282 property shall be an empty array or contain any combination of the following values: 2 (Enabled), 3
283 (Disabled), or 11 (Reset).

284 **7.3.2 CIM_EnabledLogicalElementCapabilities.ElementNameEditSupported**

285 The ElementNameEditSupported property shall have a value of TRUE when the implementation supports
286 client modification of the ElementName property of the associated instance of CIM_Fan.

287 **7.3.3 CIM_EnabledLogicalElement.MaxElementNameLen**

288 The MaxElementNameLen property shall be implemented when the ElementNameEditSupported
289 property has a value of TRUE.

290 **7.4 Fan State Management**

291 Fan state management is optional. Fan state management consists of the
292 CIM_Fan.RequestStateChange() method being supported (see section 8.2) and the
293 CIM_Fan.RequestedState property having a value other than 12 (Not Applicable).

294 **7.4.1 Fan State Management Support**

295 When no CIM_EnabledLogicalElementCapabilities instance is associated with the CIM_Fan instance, fan
296 state management shall not be supported.

297 When a CIM_EnabledLogicalElementCapabilities instance is associated with the CIM_Fan instance but
298 the CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported property is an empty array, fan
299 state management shall not be supported.

300 When a CIM_EnabledLogicalElementCapabilities instance is associated with the CIM_Fan instance and
301 the CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported property is a non-empty array,
302 fan state management shall be supported.

303 **7.5 CIM_Fan.RequestedState**

304 The CIM_Fan.RequestedState property shall have a value of 12 (Not Applicable), a value of 5 (No
305 Change), or a value that is contained in the
306 CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported property array of the associated
307 CIM_EnabledLogicalElementCapabilities instance (see section 7.3.1).

308 When fan state management is supported and the RequestStateChange() method is successfully
309 executed, the RequestedState property shall be set to the value of the RequestedState parameter of the
310 RequestStateChange() method. After the RequestStateChange() method completes successfully, the
311 RequestedState and EnabledState properties shall have equal values with the exception of the
312 transitional requested state 11 (Reset). The value of the RequestedState property may also change as a
313 result of the request for a change to the fan's enabled state by a non-CIM implementation.

314 **7.5.1 RequestedState—12 (Not Applicable)**

315 When fan state management is not supported, the value of the CIM_Fan.RequestedState property shall
316 be 12 (Not Applicable).

317 **7.5.2 RequestedState—5 (No Change)**

318 When fan state management is supported, the initial value of the CIM_Fan.RequestedState property shall
319 be 5 (No Change).

320 **7.6 CIM_Fan.EnabledState**

321 Table 2 describes the mapping between the CIM_Fan.EnabledState property values and the
322 corresponding description of the state of the fan. The CIM_Fan.EnabledState property shall match the
323 values that are specified in Table 2. When the RequestStateChange() method does not complete
324 successfully and the fan is in an indeterminate state, the CIM_Fan.EnabledState property shall have a
325 value of 5 (Not Applicable). The value of the CIM_Fan.EnabledState property may also change as the
326 result of a change to the fan's enabled state by a non-CIM implementation.

327

Table 2 – EnabledState Value Description

Value	Description	Extended Description
0	Unknown	The fan state is indeterminate.
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.

328 **7.7 CIM_SystemDevice and CIM_AssociatedCooling**

329 An instance of CIM_SystemDevice that references the CIM_Fan instance and the Scoping Instance shall
330 always exist for each instance of CIM_Fan.

331 When no instance of CIM_AssociatedCooling references the instance of CIM_Fan, the fan represented
332 by CIM_Fan cools the whole managed system. In this case, the CIM_ComputerSystem instance and the
333 CIM_Fan instance shall be referenced only by the CIM_SystemDevice association.

334 When at least one instance of CIM_AssociatedCooling references the instance of CIM_Fan, the elements
335 cooled by the fan shall be only those referenced by the CIM_AssociatedCooling association and not the
336 whole managed system.

337 **7.8 Modeling Fan Redundancy**

338 Fan redundancy modeling is optional. Even when a managed system supports and implements the
339 redundancy, the redundant fans may co-exist with non-redundant fans. The conditions and requirements
340 described in this section refer only to the CIM_Fan instances that represent redundant fans.

341 Fan redundancy is modeled using CIM_RedundancySet, which is associated to the CIM_Fan instances
342 through the CIM_MemberOfCollection and CIM_IsSpare associations.

343 When fan redundancy is implemented, at least one instance of CIM_RedundancySet shall exist. An
344 instance of CIM_MemberOfCollection shall associate the CIM_RedundancySet instance with the
345 CIM_Fan instance.

346 The CIM_Fan instances that are associated with a particular CIM_RedundancySet instance shall be
347 associated with the same CIM_ComputerSystem instance through CIM_SystemDevice associations when
348 no CIM_AssociatedCooling association references any of the CIM_Fan instances. The CIM_Fan
349 instances that are associated with a particular CIM_RedundancySet instance shall be associated with the
350 same CIM_ManagedSystemElement subclass instance through CIM_AssociatedCooling associations
351 when a CIM_AssociatedCooling association references any of the CIM_Fan instances.

352 In addition to the CIM_MemberOfCollection association, the CIM_IsSpare association may be used to
353 associate the CIM_RedundancySet instance with the CIM_Fan instance, depending on the type of
354 redundancy implemented (see section 7.8.1).

355 **7.8.1 Requirements for Different Fan Redundancy Types**

356 When the CIM_RedundancySet.TypeOfSet property contains a value of 3 (Load Balanced), 2 (N+1), or
357 both, and does not contain any other values, the CIM_Fan instances associated with the
358 CIM_RedundancySet instance shall meet the following requirements:

- 359 • The CIM_Fan instances shall be associated with the CIM_RedundancySet instance through the
360 CIM_MemberOfCollection association.
- 361 • The CIM_Fan instances shall not be associated with the CIM_RedundancySet instance through
362 the CIM_IsSpare association.

363 When the CIM_RedundancySet.TypeOfSet property contains a value of 4 (Sparing), 5 (Limited Sparing),
364 or both, a Spare Fan may exist. The Spare Fan shall be associated to the CIM_RedundancySet instance
365 through both the CIM_IsSpare association and the CIM_MemberOfCollection association. The Spare Fan
366 shall comply with one of the following requirements:

- 367 • When the value of the CIM_Fan.EnabledState property is set to 3 (Disabled), the value of the
368 referencing CIM_IsSpare instance's SpareStatus property shall be set to 3 (Cold Standby).
- 369 • When the value of the CIM_Fan.EnabledState property is set to a value other than 3 (Disabled),
370 the value of the referencing CIM_IsSpare instance's SpareStatus property shall be set to 0
371 (Unknown).

372 7.9 CIM_Fan.ElementName

373 CIM_Fan.ElementName shall be formatted as a free-form string of variable length (pattern “.*”).

374 7.9.1 CIM_Fan.ElementName Is Modifiable

375 Implementations may allow the CIM_Fan.ElementName to be modified by a client. This behavior is
376 conditional. This section describes the CIM elements and behavior requirements when an implementation
377 supports client modification of the CIM_Fan.ElementName property. CIM_Fan.ElementName property
378 shall be modifiable when the ElementNameEditSupported property of the associated
379 CIM_EnabledLogicalElementCapabilities instance has a value of TRUE.

380 7.10 Modeling Fan Speed Sensor

381 Modeling of a fan speed sensor is optional. This section describes constraints for representing a fan
382 speed sensor in addition to the constraints described in the [Sensors Profile](#).

383 A fan discrete speed sensor shall be represented by the CIM_Sensor class, which is associated with the
384 CIM_Fan instance through an instance of CIM_AssociatedSensor. The CIM_Sensor instance shall meet
385 the following requirement:

- 386 • The value of the CIM_Sensor.Type property is 5 (Tachometer).

387 A fan analog speed sensor shall be represented by the CIM_NumericSensor class, which is associated
388 with the CIM_Fan instance through an instance of CIM_AssociatedSensor. The CIM_NumericSensor
389 instance shall meet the following requirements:

- 390 • The value of the CIM_NumericSensor.Type property is 5 (Tachometer).
- 391 • The value of the CIM_NumericSensor.BaseUnits property is 19 (RPM).
- 392 • The value of the CIM_NumericSensor.RateUnits property is 0 (None).

393 Implementation may instantiate other CIM_Sensor or CIM_NumericSensor instances that are associated
394 with the CIM_Fan instance and which represent other types of sensors.

395 8 Methods

396 This section details the requirements for supporting intrinsic operations and extrinsic methods for the CIM
397 elements defined by this profile.

398 8.1 CIM_Fan.SetSpeed()

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

401 SetSpeed() return code values are specified in Table 3. Table 4 specifies SetSpeed() parameters.

402 When the CIM_Fan.VariableSpeed property has a value of FALSE or the CIM_Fan.EnabledState
403 property contains a value of 3 (Disabled), the CIM_Fan.SetSpeed() method shall not be implemented or
404 shall return a value of 1 (Not Supported).

405

Table 3 – CIM_Fan.SetSpeed() Method: Return Code Values

Value	Description
0	Request was successfully executed.
1	Method is not supported in the implementation.
2	Error occurred.

406

Table 4 – CIM_Fan.SetSpeed() Method: Parameters

Qualifiers	Name	Type	Description/Values
IN, REQ	DesiredSpeed	uint64	The desired speed for the fan in RPMs

407 **8.2 CIM_Fan.RequestStateChange()**

408 Invocation of the CIM_Fan.RequestStateChange() method changes the element’s state to the value
 409 specified in the RequestedState parameter.

410 RequestStateChange() return code values are specified in Table 5. RequestStateChange() parameters
 411 are specified in Table 6.

412 When fan state management is supported (see section 7.4.1), the RequestStateChange() method shall
 413 be implemented and shall not return a value of 1 (Not Supported).

414 Invoking the CIM_Fan.RequestStateChange() method multiple times could result in earlier requests
 415 being overwritten or lost.

416 No standard messages are defined for this method.

417

Table 5 – CIM_Fan.RequestStateChange() Method: Return Code Values

Value	Description
0	Request was successfully executed.
1	Method is not supported in the implementation.
2	Error occurred.
4096	Job started.

418

Table 6 – CIM_Fan.RequestStateChange() Method: Parameters

Qualifiers	Name	Type	Description/Values
IN	RequestedState	uint16	State : 2 (Enabled) 3 (Disabled) 11 (Reset)
OUT	Job	CIM_ConcreteJob REF	Returned if job started
IN	TimeoutPeriod	Datetime	Client specified maximum amount of time the transition to a new state is supposed to take: 0 or NULL – No time requirements <interval> – Maximum time allowed

419 **8.2.1 RequestStateChange() for the Spare Fan**

420 After the successful execution of the RequestStateChange() method on the Spare Fan with the
 421 RequestedState parameter set to 3 (Disabled), the SpareStatus of the referenced CIM_IsSpare
 422 association shall have a value of 3 (Cold Standby).

423 **8.3 CIM_RedundancySet.Failover()**

424 The CIM_RedundancySet.Failover() method forces a failover from one member of a
 425 CIM_RedundancySet collection to another. After the successful execution of the method, the fan that is
 426 represented by the CIM_Fan instance referenced by the FailoverFrom parameter becomes inactive. The
 427 fan that is represented by the CIM_Fan instance referenced by the FailoverTo parameter takes over as
 428 the active fan.

429 The Failover() method may be supported if the FailoverSupported property of at least one instance of
 430 CIM_IsSpare that references the CIM_RedundancySet instance contains a value of 3 (Manual) or 4 (Both
 431 Manual and Automatic).

432 The Failover() method shall not be supported if the FailoverSupported property of every instance of
 433 CIM_IsSpare that references the CIM_RedundancySet contains a value of 2 (Automatic).

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

- 436 • The CIM_Fan instance that is referenced by the FailoverTo parameter is not a Spare Fan.
- 437 • The CIM_Fan instance that is referenced by the FailoverFrom parameter is not associated with
 438 the CIM_RedundancySet only through an instance of CIM_MemberOfCollection.

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

- 440 • The CIM_Fan that is referenced by the FailoverTo parameter shall take over as the active fan.
- 441 • The CIM_Fan instance that is referenced by the FailoverTo parameter shall be associated with
 442 the CIM_RedundancySet only through an instance of CIM_MemberOfCollection.
- 443 • The CIM_Fan instance that is referenced by FailoverFrom parameter shall become a Spare
 444 Fan.
- 445 • When fan state management is supported, the CIM_Fan instance that is referenced by the
 446 FailoverFrom parameter shall not have an EnabledState property value of 2 (Enabled).

447 CIM_RedundancySet.Failover() return code values are specified in Table 7.

448 CIM_RedundancySet.Failover() parameters are specified in Table 8.

449 No standard messages are defined for this method.

450 **Table 7 – CIM_RedundancySet.Failover() Method: Return Code Values**

Value	Description
0	Method completed successfully.
1	Method is not supported in the implementation.
2	Error occurred.

451

Table 8 – CIM_RedundancySet.Failover() Method: Parameters

Qualifiers	Name	Type	Description/Values
IN, REQ	FailoverFrom	CIM_ManagedElement REF	The redundant element that will become inactive
IN, REQ	FailoverTo	CIM_ManagedElement REF	The redundant element that will become active and take over for the inactivated element

452 **8.4 Profile Conventions for Operations**

453 Support for operations for each profile class (including associations) is specified in the following
 454 subclauses. Each subclause includes either the statement “All operations in the default list in section 8.4
 455 are supported as described by [DSP0200 version 1.2](#)” or a table listing all of the operations that are not
 456 supported by this profile or where the profile requires behavior other than that described by [DSP0200](#)
 457 [version 1.2](#).

458 The default list of operations is as follows:

- 459 • GetInstance
- 460 • EnumerateInstances
- 461 • EnumerateInstanceNames
- 462 • Associators
- 463 • AssociatorNames
- 464 • References
- 465 • ReferenceNames

466 A compliant implementation shall support all of the operations in the default list for each class, unless the
 467 “Requirement” column states something other than *Mandatory*.

468 **8.5 CIM_AssociatedCooling**

469 Table 9 lists operations that either have special requirements beyond those from [DSP0200 version 1.2](#) or
 470 shall not be supported.

471

Table 9 – Operations: CIM_AssociatedCooling

Operation	Requirement	Messages
EnumerateInstanceNames	Unspecified	None
Associators	Unspecified	None
AssociatorNames	Unspecified	None
References	Unspecified	None
ReferenceNames	Unspecified	None

472 8.6 CIM_ElementCapabilities

473 Table 10 lists operations that either have special requirements beyond those from [DSP0200 version 1.2](#)
474 or shall not be supported.

475 **Table 10 – Operations: CIM_ElementCapabilities**

Operation	Requirement	Messages
EnumerateInstanceNames	Unspecified	None
Associators	Unspecified	None
AssociatorNames	Unspecified	None
References	Unspecified	None
ReferenceNames	Unspecified	None

476 8.7 CIM_EnabledLogicalElementCapabilities

477 All operations in the default list in section 8.4 are supported as described by [DSP0200 version 1.2](#).

478 8.8 CIM_Fan

479 Table 11 lists operations that either have special requirements beyond those from [DSP0200 version 1.2](#)
480 or shall not be supported.

481 **Table 11 – Operations: CIM_Fan**

Operation	Requirement	Messages
ModifyInstance	Optional. See section 8.8.1.	None

482 8.8.1 CIM_Fan—ModifyInstance

483 This section details the requirements for the ModifyInstance operation applied to an instance of CIM_Fan.
484 The ModifyInstance operation may be supported.

485 The ModifyInstance operation shall be supported and the CIM_Fan.ElementName property shall be
486 modifiable when the ElementNameEditSupported property of the
487 CIM_EnabledLogicalElementCapabilities instance that is associated with the CIM_Fan instance has a
488 value of TRUE. See section 8.8.1.1.

489 8.8.1.1 CIM_Fan.ElementName

490 When the ElementNameEditSupported property of the CIM_EnabledLogicalElementCapabilities instance
491 that is associated with the CIM_Fan instance has a value of TRUE, the implementation shall allow the
492 ModifyInstance operation to change the value of the ElementName property of the CIM_Fan instance.
493 The ModifyInstance operation shall enforce the length restriction specified in the MaxElementNameLen
494 property of the CIM_EnabledLogicalElementCapabilities instance.

495 When the ElementNameEditSupported property of the CIM_EnabledLogicalElementCapabilities instance
496 has a value of FALSE or if there is no CIM_EnabledLogicalElementCapabilities associated with the
497 CIM_Fan instance through the CIM_ElementCapabilities association, the implementation shall not allow
498 the ModifyInstance operation to change the value of the ElementName property of the CIM_Fan instance.

499 8.9 CIM_HostedCollection

500 Table 14 lists operations that either have special requirements beyond those from [DSP0200 version 1.2](#)
501 or shall not be supported.

502 **Table 12 – Operations: CIM_HostedCollection**

Operation	Requirement	Messages
EnumerateInstanceNames	Unspecified	None
Associators	Unspecified	None
AssociatorNames	Unspecified	None
References	Unspecified	None
ReferenceNames	Unspecified	None

503 8.10 CIM_IsSpare

504 Table 13 lists operations that either have special requirements beyond those from [DSP0200 version 1.2](#)
505 or shall not be supported.

506 **Table 13 – Operations: CIM_IsSpare**

Operation	Requirement	Messages
EnumerateInstanceNames	Unspecified	None
Associators	Unspecified	None
AssociatorNames	Unspecified	None
References	Unspecified	None
ReferenceNames	Unspecified	None

507 8.11 CIM_MemberOfCollection

508 Table 14 lists operations that either have special requirements beyond those from [DSP0200 version 1.2](#)
509 or shall not be supported.

510 **Table 14 – Operations: CIM_MemberOfCollection**

Operation	Requirement	Messages
EnumerateInstanceNames	Unspecified	None
Associators	Unspecified	None
AssociatorNames	Unspecified	None
References	Unspecified	None
ReferenceNames	Unspecified	None

511 8.12 CIM_NumericSensor

512 The CIM_NumericSensor class is defined by the [Sensors Profile](#). There are no requirements in addition
513 to those mandated by the [Sensors Profile](#) (see section 2).

514 8.13 CIM_OwningCollectionElement

515 Table 14 lists operations that either have special requirements beyond those from [DSP0200 version 1.2](#)
516 or shall not be supported.

517 **Table 15 – Operations: CIM_OwningCollectionElement**

Operation	Requirement	Messages
EnumerateInstanceNames	Unspecified	None
Associators	Unspecified	None
AssociatorNames	Unspecified	None
References	Unspecified	None
ReferenceNames	Unspecified	None

518 8.14 CIM_RedundancySet

519 All operations in the default list in section 8.4 are supported as described by [DSP0200 version 1.2](#).

520 8.15 CIM_Sensor

521 The CIM_Sensor class is defined by the [Sensors Profile](#). There are no requirements in addition to those
522 mandated by the [Sensors Profile](#) (see section 2).

523 8.16 CIM_SystemDevice

524 Table 16 lists operations that either have special requirements beyond those from [DSP0200 version 1.2](#)
525 or shall not be supported.

526 **Table 16 – Operations: CIM_SystemDevice**

Operation	Requirement	Messages
EnumerateInstanceNames	Unspecified	None
Associators	Unspecified	None
AssociatorNames	Unspecified	None
References	Unspecified	None
ReferenceNames	Unspecified	None

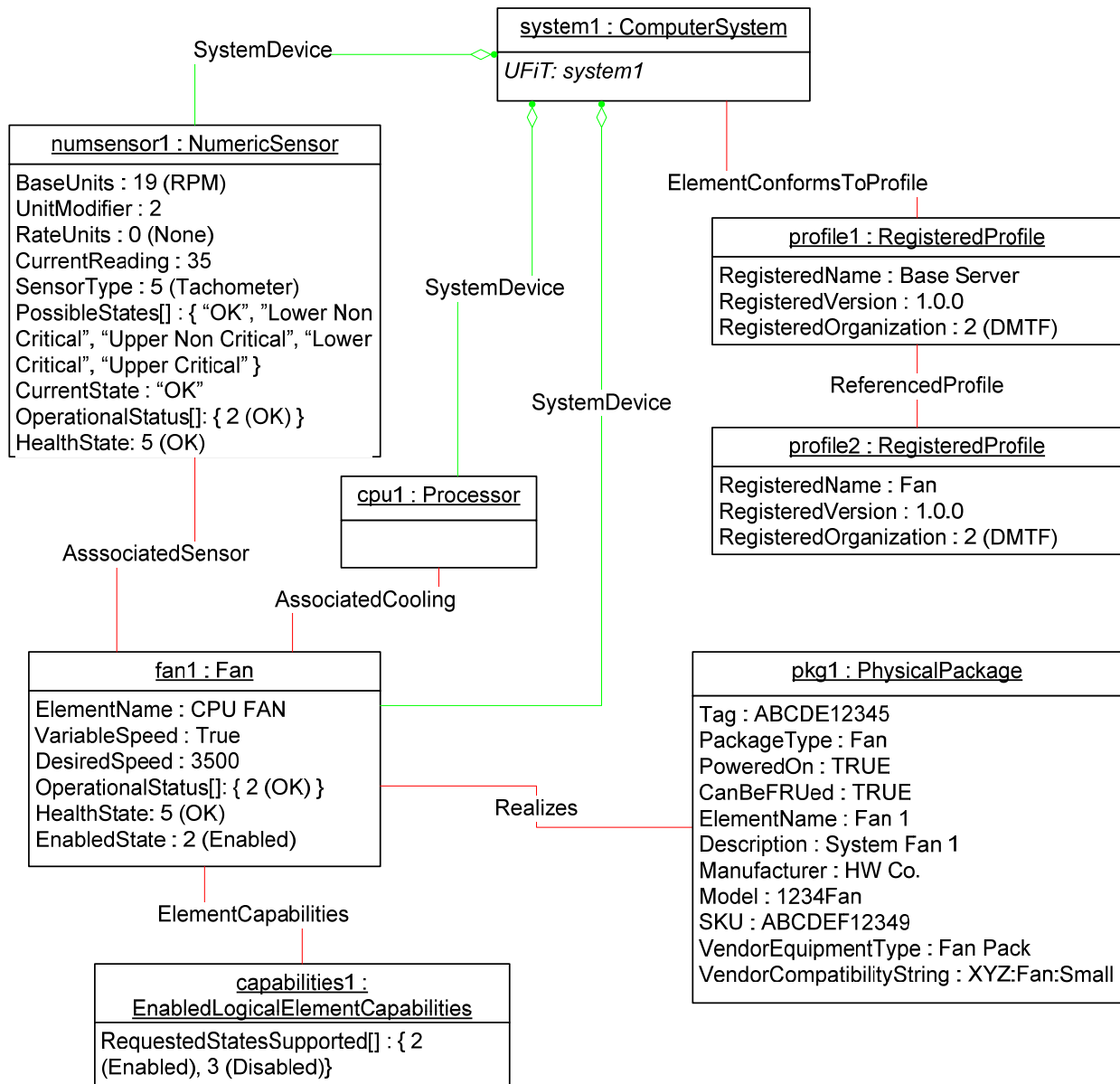
527 9 Use Cases

528 This section contains object diagrams and use cases for the *Fan Profile*.

529 9.1 Object Diagrams

530 Figure 2 represents a possible instantiation of the *Fan Profile*. In this instantiation, system1 has a
531 processor fan, fan1, which is variable speed. fan1 has an associated numeric sensor, numsensor1, that
532 reads the current speed of the fan. fan1's physical package information is represented with the
533 associated pkg1. profile2 shows the version of the current *Fan Profile* implementation. In this
534 instantiation, the fan cooled device, cpu1, is associated to fan1 through an instance of
535 CIM_AssociatedCooling. Because a CIM_AssociatedCooling instance references fan1, the
536 CIM_SystemDevice association between fan1 and system1 is for scoping purposes only; system1 is not
537 cooled by fan1.

538 For simplicity, the prefix CIM_ has been removed from the names of the classes.



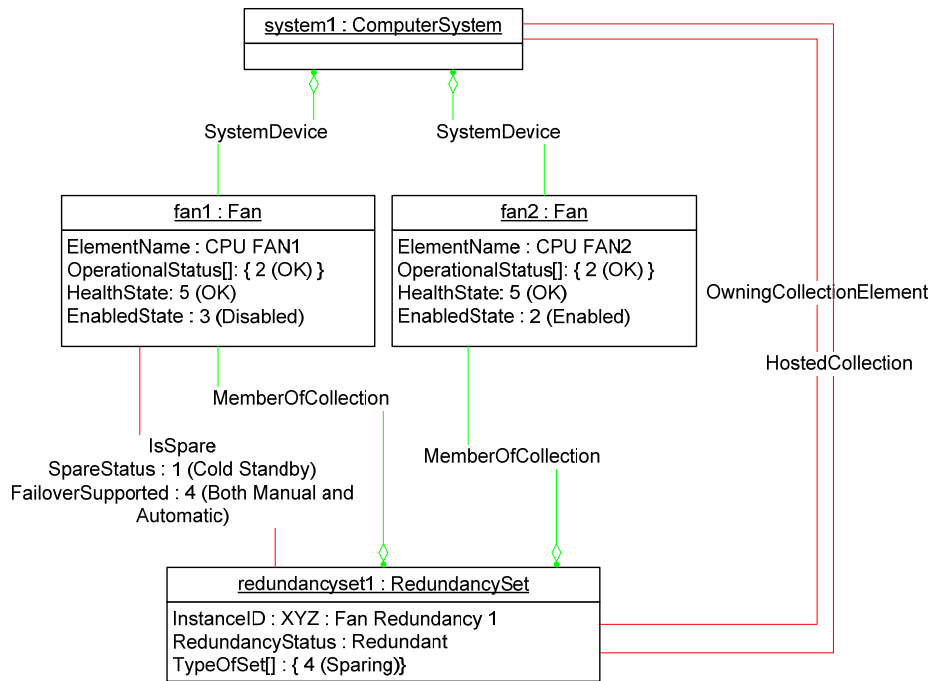
539

540

Figure 2 – Fan Profile: Object Diagram

541 Figure 3 represents a possible instantiation of the *Fan Profile* with redundancy. system1 has sparing fan
 542 redundancy. Because fan1 is associated with redundancysset1 through the CIM_IsSpare association and
 543 the value of the EnabledState property is 3 (Disabled), fan1 is the Spare Fan. fan2 is the active cooling
 544 fan of system1 because its EnabledState property has value of 2 (Enabled) and it is associated with
 545 redundancysset1 only through an instance of CIM_MemberOfCollection.

546 If redundancysset1 supports the Failover() method, a client can execute the Failover() method with the
 547 FailoverFrom parameter referencing fan2 and the FailoverTo parameter referencing fan1.

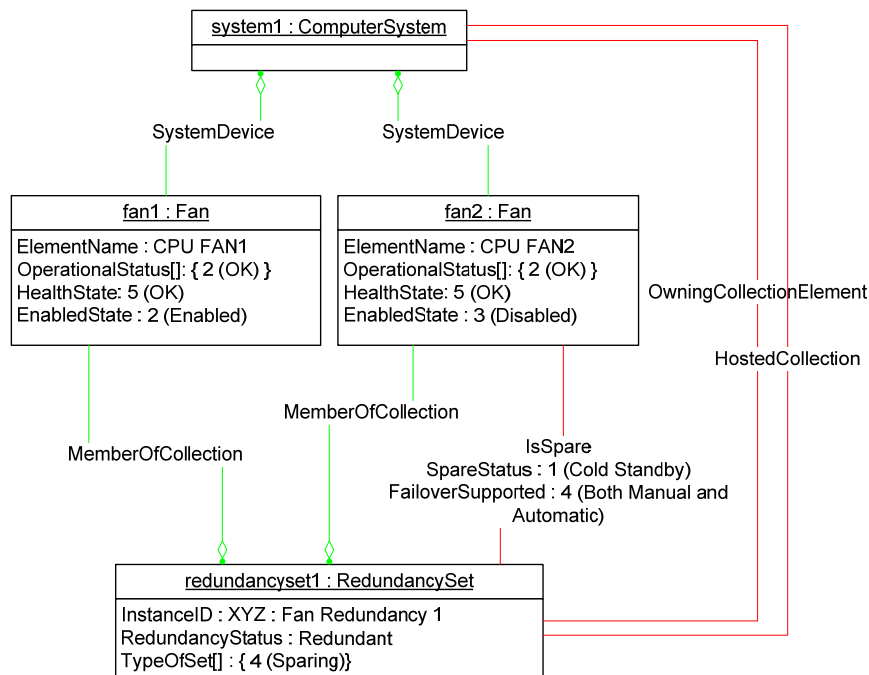


548

549

Figure 3 – Fan Profile: Fan Redundancy before Failover() Method Execution

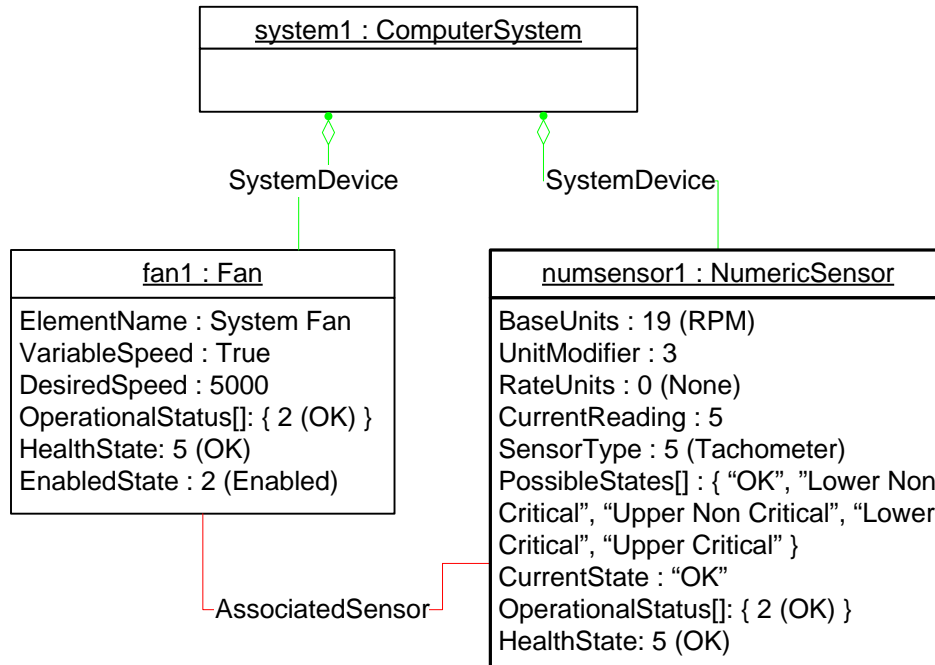
550 Figure 4 represents the instances of Figure 3 after the successful execution of the Failover() method.
 551 fan1 becomes the active fan for system1; its EnabledState property contains a value of 2 (Enabled) and it
 552 is not associated with redundancyset1 through the CIM_IsSpare association. fan2 has the EnabledState
 553 value of 3 (Disabled) and is associated with redundancyset1 through the CIM_IsSpare association.
 554 Because fan1 and fan2 are not referenced by the CIM_AssociatedCooling association, they are providing
 555 general cooling for system1, as denoted by the CIM_SystemDevice association.



556

557

Figure 4 – Fan Profile: Fan Redundancy after Failover() Method Execution



572

573

Figure 6 – Fan Profile: After the SetSpeed() Method Execution

574 **9.3 Reset the Fan**

575 For the given instance of CIM_Fan, find the associated instance of
 576 CIM_EnabledLogicalElementCapabilities. If the
 577 CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported property is a non-empty array that
 578 contains a value of 11 (Reset), execute the RequestStateChange() method with the RequestedState
 579 parameter set to 11 (Reset), which will disable and enable the fan represented by this instance.

580 **9.4 Retrieve the Fan Redundancy Status**

581 For the given instance of CIM_Fan, select the instance of CIM_RedundancySet that is associated through
 582 an instance of CIM_MemberOfCollection. Retrieve the value of the
 583 CIM_RedundancySet.RedundancyStatus property.

584 **9.5 Find the Spare Fan Supply for the Given Active Fan**

585 For the given instance of CIM_Fan, select the associated CIM_RedundancySet instance. Select the
 586 CIM_Fan instances that are associated with the CIM_RedundancySet instance through the CIM_IsSpare
 587 association.

588 **9.6 Show the Fan Sensor Information**

589 Select all the instances of CIM_AssociatedSensor that reference the given instance of CIM_Fan. Select
 590 the referenced CIM_NumericSensor or CIM_Sensor instances that are referenced by an instance of
 591 CIM_AssociatedSensor. Show the values of the properties of the selected CIM_NumericSensor and
 592 CIM_Sensor instances.

593 **9.7 Find the Managed System Elements Cooled by the Fan**

594 Select all the CIM_AssociatedCooling instances that reference the given instance of CIM_Fan.

595 If the CIM_AssociatedCooling instances exist, the CIM_AssociatedCooling.Dependent properties
 596 reference all the instances of the subclass of the CIM_ManagedSystemElement instance that is cooled by
 597 the fan.

598 If no CIM_AssociatedCooling instances exist, select the CIM_ComputerSystem instance that is
 599 associated with the given instance of the CIM_Fan through the CIM_SystemDevice association.

600 **9.8 Determine Whether the CIM_Fan.ElementName Is Modifiable**

601 A client can determine whether it can modify the CIM_Fan.ElementName property as follows:

- 602 1) Find the CIM_EnabledLogicalElementCapabilities instance that is associated with the CIM_Fan
 603 instance,
- 604 2) Query the value of the ElementNameEditSupported property of the instance. If the value is
 605 TRUE, the client can modify the CIM_Fan.ElementName property.

606 **10 CIM Elements**

607 Table 17 shows the instances of CIM Elements for this profile. Instances of the CIM Elements shall be
 608 implemented as described in Table 17. The implementation requirements for the classes and properties
 609 described in this section are defined in section 7 (“Implementation Requirements”).

610 **Table 17 – CIM Elements: Fan Profile**

Element Name	Requirement	Description
Classes		
CIM_AssociatedCooling	Optional	See sections 7.7 and 10.1.
CIM_ElementCapabilities	Optional	See section 10.2.
CIM_EnabledLogicalElementCapabilities	Optional	See sections 7.3 and 10.3.
CIM_Fan	Mandatory	See sections 7.1 and 10.4.
CIM_HostedCollection	Optional	See section 10.5.
CIM_IsSpare	Optional	See section 10.6.
CIM_MemberOfCollection	Optional	See section 10.7.
CIM_NumericSensor	Optional	See sections 7.10 and 10.8.
CIM_OwningCollectionElement	Optional	See section 10.9.
CIM_RedundancySet	Optional	See section 10.10.
CIM_RegisteredProfile	Mandatory	See section 10.11.
CIM_Sensor	Optional	See sections 7.10 and 10.12.
CIM_SystemDevice	Mandatory	See sections 7.7 and 10.13.
Indications		
None defined in this profile		

611 **10.1 CIM_AssociatedCooling**

612 CIM_AssociatedCooling associates CIM_Fan with a subclass of CIM_ManagedSystemElement. Table 18
613 provides information about the properties of CIM_AssociatedCooling.

614 **Table 18 – Class: CIM_AssociatedCooling**

Properties	Requirement	Notes
Antecedent	Mandatory	Key: Shall reference an instance of CIM_Fan Cardinality 1..* indicating one or more references
Dependent	Mandatory	Key: Shall reference an instance of a subclass of CIM_ManagedSystemElement for which the fan is providing cooling Cardinality * indicating zero or more references

615 **10.2 CIM_ElementCapabilities**

616 CIM_ElementCapabilities associates CIM_Fan with an instance of
617 CIM_EnabledLogicalElementCapabilities that describes the capabilities of CIM_Fan. Table 19 provides
618 information about the properties of CIM_ElementCapabilities.

619 **Table 19 – Class: CIM_ElementCapabilities**

Properties	Requirement	Notes
ManagedElement	Mandatory	Key: Shall reference an instance of CIM_Fan Cardinality 1..* indicating one or more references
Capabilities	Mandatory	Key: Shall reference an instance of CIM_EnabledLogicalElementCapabilities Cardinality 0..1 indicating zero or one reference

620 **10.3 CIM_EnabledLogicalElementCapabilities**

621 CIM_EnabledLogicalElementCapabilities represents the capabilities of the fan. Table 20 provides
622 information about the properties of CIM_EnabledLogicalElementCapabilities.

623 **Table 20 – Class: CIM_EnabledLogicalElementCapabilities**

Properties	Requirement	Notes
InstanceID	Mandatory	Key
RequestedStatesSupported	Mandatory	See section 7.3.1.
ElementNameEditSupported	Mandatory	See section 7.3.2.
MaxElementNameLen	Conditional	See section 7.3.3.

624 **10.4 CIM_Fan**

625 CIM_Fan represents a fan in a managed system. Table 21 provides information about the properties of
 626 CIM_Fan.

627 **Table 21 – Class: CIM_Fan**

Properties and Methods	Requirement	Notes
SystemCreationClassName	Mandatory	Key
SystemName	Mandatory	Key
CreationClassName	Mandatory	Key
DeviceID	Mandatory	Key
OperationalStatus	Mandatory	
HealthState	Mandatory	
VariableSpeed	Mandatory	
DesiredSpeed	Conditional	See section 7.2.
ActiveCooling	Mandatory	Shall match TRUE
EnabledState	Mandatory	See section 7.6.
RequestedState	Mandatory	See section 7.5.
ElementName	Mandatory	See section 7.9.
RequestStateChange()	Conditional	See section 8.2.

628 **10.5 CIM_HostedCollection**

629 HostedCollection defines a SystemSpecificCollection in the context of a scoping System. Table 22
 630 provides information about the properties of CIM_HostedCollection.

631 **Table 22 – Class: CIM_HostedCollection**

Properties	Requirement	Notes
Antecedent	Mandatory	Key: Shall reference a CIM_ComputerSystem instance of which a CIM_RedundancySet instance is a member Cardinality 1 indicating one reference
Dependent	Mandatory	Key: Shall reference a CIM_RedundancySet instance Cardinality * indicating zero or more references

632 **10.6 CIM_IsSpare**

633 CIM_IsSpare associates CIM_Fan with the CIM_RedundancySet instance of which the CIM_Fan is a
 634 member and where CIM_Fan is a Spare Fan. Table 23 provides information about the properties of
 635 CIM_IsSpare.

636 **Table 23 – Class: CIM_IsSpare**

Properties	Requirement	Notes
Antecedent	Mandatory	Key: Shall reference a CIM_RedundancySet instance of which a CIM_Fan instance is a member and where the CIM_Fan instance is a Spare Fan Cardinality * indicating zero or many references
Dependent	Mandatory	Key: Shall reference the Spare Fan Cardinality * indicating one or more references
SpareStatus	Mandatory	See section 7.8.1.
FailoverSupported	Mandatory	

637 **10.7 CIM_MemberOfCollection**

638 CIM_MemberOfCollection associates CIM_Fan with the CIM_RedundancySet instance of which CIM_Fan
 639 is a member. Table 24 provides information about the properties of CIM_MemberOfCollection.

640 **Table 24 – Class: CIM_MemberOfCollection**

Properties	Requirement	Notes
Collection	Mandatory	Key: Shall reference a CIM_RedundancySet instance of which a CIM_Fan instance is a member. Cardinality * indicating zero or many references
Member	Mandatory	Key: Shall reference a CIM_Fan instance Cardinality * indication one or many references

641 **10.8 CIM_NumericSensor**

642 The CIM_NumericSensor class is defined by the [Sensors Profile](#). The requirements denoted in Table 25
 643 are in addition to those mandated by the [Sensors Profile](#) (see section 2).

644 **Table 25 – Class: CIM_NumericSensor**

Properties	Requirement	Description
SensorType	Mandatory	SensorType shall be set to 5 (Tachometer).
BaseUnits	Mandatory	BaseUnits shall be set to 19 (RPM).
RateUnits	Mandatory	RateUnits shall be set to 0 (None).

645 **10.9 CIM_OwningCollectionElement**

646 CIM_OwningCollectionElement associates CIM_RedundancySet with the CIM_ComputerSystem instance of
 647 which CIM_RedundancySet is a member. Table 26 provides information about the properties of
 648 CIM_OwningCollectionElement.

649 **Table 26 – Class: CIM_OwningCollectionElement**

Properties	Requirement	Notes
OwningElement	Mandatory	Key: Shall reference a CIM_ComputerSystem instance of which a CIM_RedundancySet instance is a member Cardinality 1 indicating one reference
OwnedElement	Mandatory	Key: Shall reference a CIM_RedundancySet instance Cardinality * indicating zero or more references

650 **10.10 CIM_RedundancySet**

651 CIM_RedundancySet represents the aggregation of redundant fans. Table 27 provides information about
 652 the properties of CIM_RedundancySet.

653 **Table 27 – Class: CIM_RedundancySet**

Properties and Methods	Requirement	Notes
InstanceID	Mandatory	Key
RedundancyStatus	Mandatory	
TypeOfSet	Mandatory	See section 7.8.1.
MinNumberNeeded	Mandatory	Shall match 0 when the minimum number of fans needed for the redundancy is unknown
ElementName	Mandatory	Shall be formatted as a free-form string of variable length (pattern “.*”)
Failover()	Optional	See section 8.2.1.

654 **10.11 CIM_RegisteredProfile**

655 The CIM_RegisteredProfile class is defined by the [Profile Registration Profile](#). The requirements denoted
 656 in Table 28 are in addition to those mandated by the [Profile Registration Profile](#).

657 **Table 28 – Class: CIM_RegisteredProfile**

Properties	Requirement	Description
RegisteredName	Mandatory	This property shall have a value of “Fan”.
RegisteredVersion	Mandatory	This property shall have a value of “1.0.1”.
RegisteredOrganization	Mandatory	This property shall have a value of 2 (DMTF).

658 NOTE: Previous versions of this document included the suffix “Profile” for the RegisteredName value. If
 659 implementations querying for the RegisteredName value find the suffix “Profile”, they should ignore the suffix, with
 660 any surrounding white spaces, before any comparison is done with the value as specified in this document.

661 **10.12 CIM_Sensor**

662 The CIM_Sensor class is defined by the [Sensors Profile](#). The requirements denoted in Table 29 are in
 663 addition to those mandated by the [Sensors Profile](#) (see section 2).

664 **Table 29 – Class: CIM_Sensor**

Properties	Requirement	Description
SensorType	Mandatory	SensorType shall be set to 5 (Tachometer).

665 **10.13 CIM_SystemDevice**

666 CIM_SystemDevice associates CIM_Fan with the CIM_ComputerSystem instance of which CIM_Fan is a
 667 member. Table 30 provides information about the properties of CIM_SystemDevice.

668 **Table 30 – Class: CIM_SystemDevice**

Properties	Requirement	Notes
GroupComponent	Mandatory	Key: Shall reference a CIM_ComputerSystem instance of which a CIM_Fan instance is a member Cardinality 1 indicating one reference
PartComponent	Mandatory	Key: Shall reference a CIM_Fan instance Cardinality * indicating zero or more references

669

670
671
672
673

ANNEX A (informative)

Change Log

Version	Date	Description
1.0.0a	06/13/2006	Preliminary Standard
1.0.0	10/12/2007	Final Standard
1.0.1	09/23/2008	1.0.1 Final

674
675
676
677

ANNEX B (informative)

Acknowledgments

678 The authors wish to acknowledge the following people.

679 **Editors:**

- 680 • Jon Hass – Dell Inc.
- 681 • Khachatur Papanyan – Dell Inc.
- 682 • Jeff Hilland – HP
- 683 • Jim Davis – WBEM Solutions

684 **Contributors:**

- 685 • Jon Hass – Dell Inc.
- 686 • Khachatur Papanyan – Dell Inc.
- 687 • Enoch Suen – Dell
- 688 • Jeff Hilland – HP
- 689 • Christina Shaw – HP
- 690 • Aaron Merkin – IBM
- 691 • Perry Vincent – Intel
- 692 • John Leung – Intel
- 693 • John Ackerley – Sun Microsystems