



1
2
3
4

Document Identifier: DSP1009

Date: 2024-01-05

Version: 1.2.0

5 **Sensors Profile**

- 6 **Supersedes: 1.1.1**
- 7 **Document Class: Normative**
- 8 **Document Status: Published**
- 9 **Document Language: en-US**

10 Copyright Notice

11 Copyright © 2006–2009, 2012, 2014, 2024 DMTF. All rights reserved.

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

16 Implementation of certain elements of this standard or proposed standard may be subject to third-party
17 patent rights, including provisional patent rights (herein “patent rights”). DMTF makes no representations
18 to users of the standard as to the existence of such rights and is not responsible to recognize, disclose, or
19 identify any or all such third-party patent right owners or claimants, nor for any incomplete or inaccurate
20 identification or disclosure of such rights, owners, or claimants. DMTF shall have no liability to any party,
21 in any manner or circumstance, under any legal theory whatsoever, for failure to recognize, disclose, or
22 identify any such third-party patent rights, or for such party’s reliance on the standard or incorporation
23 thereof in its products, protocols, or testing procedures. DMTF shall have no liability to any party
24 implementing such standard, whether such implementation is foreseeable or not, nor to any patent owner
25 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26 withdrawn or modified after publication, and shall be indemnified and held harmless by any party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 patents may relate to or impact implementations of DMTF standards, visit
31 <https://www.dmtf.org/about/policies/disclosures>.

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

CONTENTS

34	Foreword	5
35	Introduction.....	6
36	1 Scope	7
37	2 Normative references	7
38	3 Terms and definitions	7
39	4 Symbols and abbreviated terms.....	8
40	5 Synopsis	8
41	6 Description	9
42	7 Implementation requirements.....	9
43	7.1 CIM_Sensor and CIM_NumericSensor	9
44	7.2 CIM_Sensor.PossibleStates	9
45	7.3 CIM_NumericSensor.PossibleStates.....	10
46	7.4 CIM_Sensor.CurrentState and CIM_NumericSensor.CurrentState	11
47	7.5 CIM_NumericSensor.LowerThresholdNonCritical.....	11
48	7.6 CIM_NumericSensor.UpperThresholdNonCritical.....	11
49	7.7 CIM_NumericSensor.LowerThresholdCritical	11
50	7.8 CIM_NumericSensor.UpperThresholdCritical	12
51	7.9 CIM_NumericSensor.LowerThresholdFatal	12
52	7.10 CIM_NumericSensor.UpperThresholdFatal	12
53	7.11 CIM_NumericSensor.SupportedThresholds.....	12
54	7.12 CIM_NumericSensor.SettableThresholds	12
55	7.13 CIM_EnabledLogicalElementCapabilities.....	12
56	7.14 Sensor state management.....	13
57	7.15 CIM_Sensor.RequestedState and CIM_NumericSensor.RequestedState	13
58	7.16 CIM_Sensor.EnabledState and CIM_NumericSensor.EnabledState	14
59	7.17 CIM_Sensor.OtherSensorTypeDescription and	
60	CIM_NumericSensor.OtherSensorTypeDescription	14
61	7.18 CIM_SystemDevice and CIM_AssociatedSensor	14
62	7.19 CIM_Sensor.ElementName and CIM_NumericSensor.ElementName	14
63	7.20 CIM_Sensor.SensorContext	15
64	8 Methods.....	15
65	8.1 CIM_Sensor.RequestStateChange() and CIM_NumericSensor.RequestStateChange()	15
66	8.2 CIM_NumericSensor.RestoreDefaultThresholds()	16
67	8.3 Profile conventions for operations	16
68	8.4 CIM_Sensor	17
69	8.5 CIM_NumericSensor.....	17
70	8.6 CIM_EnabledLogicalElementCapabilities.....	18
71	8.7 CIM_ElementCapabilities	18
72	8.8 CIM_SystemDevice	19
73	8.9 CIM_AssociatedSensor	19
74	9 Use cases.....	19
75	9.1 Object diagrams.....	19
76	9.2 Show all current states of the monitored devices in the computer system.....	20
77	9.3 Find sensors associated with a specific component.....	20
78	9.4 Change upper non-critical threshold of numeric sensor	20
79	9.5 Reset sensor to threshold defaults	20
80	9.6 Determining whether ElementName can be modified	21
81	9.7 Determining whether state management is supported	21
82	9.8 Find inlet temperature sensor instance.....	21
83	10 CIM Elements	21
84	10.1 CIM_Sensor	22
85	10.2 CIM_NumericSensor.....	23

86	10.3	CIM_EnabledLogicalElementCapabilities.....	24
87	10.4	CIM_ElementCapabilities	24
88	10.5	CIM_SystemDevice	24
89	10.6	CIM_AssociatedSensor	25
90	10.7	CIM_RegisteredProfile.....	25
91	ANNEX A (informative)	Change log.....	26

92

93 Figures

94	Figure 1 – Sensors Profile: Class diagram	9
95	Figure 2 – Sensors Profile: Object diagram	20

96

97 Tables

98	Table 1 – Related profiles	8
99	Table 2 – CIM_Sensor.PossibleStates values for discrete sensors	10
100	Table 3 – CIM_NumericSensor.PossibleStates values for numeric sensors	11
101	Table 4 – EnabledState value description	14
102	Table 5 – Sensor Context Values	15
103	Table 6 – RequestStateChange() method: Return code values	15
104	Table 7 – RequestStateChange() method: Parameters	16
105	Table 8 – CIM_NumericSensor.RestoreDefaultThresholds() Method: Return code values.....	16
106	Table 9 – Operations: CIM_Sensor	17
107	Table 10 – Operations: CIM_NumericSensor.....	17
108	Table 11 – Operations: CIM_ElementCapabilities	18
109	Table 12 – Operations: CIM_SystemDevice.....	19
110	Table 13 – Operations: CIM_AssociatedSensor.....	19
111	Table 14 – CIM Elements: Sensors Profile Profile.....	21
112	Table 15 – Class: CIM_Sensor	22
113	Table 16 – Class: CIM_NumericSensor.....	23
114	Table 17 – Class: CIM_EnabledLogicalElementCapabilities.....	24
115	Table 18 – Class: CIM_ElementCapabilities.....	24
116	Table 19 – Class: CIM_SystemDevice	24
117	Table 20 – Class: CIM_AssociatedSensor	25
118	Table 21 – Class: CIM_RegisteredProfile	25

119

120

Foreword

121 The *Sensors Profile* (DSP1009) was prepared by the CIM Profiles for Platforms and Services (CPPS)
122 Task Force of DMTF.

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

125 **Acknowledgments**

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

127 **Editors:**

- 128 • Jim Davis – WBEM Solutions
- 129 • Jon Hass – Dell Inc.
- 130 • Steve Lee – Microsoft Corporation
- 131 • Khachatur Papanyan – Dell Inc.
- 132 • Hemal Shah – Broadcom Corporation

133 **Contributors:**

- 134 • Jim Davis – WBEM Solutions
- 135 • Jon Hass – Dell Inc.
- 136 • Jeff Hilland – Hewlett-Packard Company
- 137 • John Leung – Intel
- 138 • Jeff Lynch – IBM
- 139 • Divyanand Malavalli – AMD
- 140 • Linda Martinez – Dell Inc.
- 141 • Aaron Merkin – IBM
- 142 • Khachatur Papanyan – Dell Inc.
- 143 • Hemal Shah – Broadcom Corporation
- 144 • Christina Shaw – Hewlett-Packard Company
- 145 • Perry Vincent – Intel

146

Introduction

147 This document defines the classes used to describe the sensors in a managed system. The document
148 also defines association classes that describe the relationship of the sensors with the monitored devices
149 and with DMTF profile version information. The information in this specification is intended to be sufficient
150 for a provider or consumer of this data to identify unambiguously the classes, properties, methods, and
151 values that are mandatory to be instantiated and manipulated to represent and manage sensors of
152 managed systems and subsystems that are modeled using the DMTF CIM core and extended model
153 definitions.

154 The target audience for this specification is implementers who are writing CIM-based providers or
155 consumers of management interfaces that represent the component described in this document.
156

157 1 Scope

158 The *Sensors Profile* extends the management capabilities of referencing profiles by adding the capability
159 to represent sensors. The sensor's relationship with devices and the profile's registration for the schema
160 implementation version information are also described.

161 2 Normative references

162 The following referenced documents are indispensable for the application of this document. For dated or
163 versioned references, only the edition cited (including any corrigenda or DMTF update versions) applies.
164 For references without a date or version, the latest published edition of the referenced document
165 (including any corrigenda or DMTF update versions) applies.

166 DMTF DSP0004, *Common Information Model (CIM) Infrastructure 2.6*,
167 https://www.dmtf.org/standards/published_documents/DSP0004_2.6.pdf

168 DMTF DSP0200, *CIM Operations over HTTP 1.3*
169 https://www.dmtf.org/standards/published_documents/DSP0200_1.3.pdf

170 DMTF DSP1000, *Management Profile Specification Template 1.0*
171 <https://www.dmtf.org/sites/default/files/standards/documents/DSP1000.pdf>

172 DMTF DSP1001, *Management Profile Specification Usage Guide 1.1*,
173 https://www.dmtf.org/standards/published_documents/DSP1001_1.1.pdf

174 DMTF DSP1033, *Profile Registration 1.0*,
175 https://www.dmtf.org/standards/published_documents/DSP1033_1.0.pdf

176 ISO/IEC Directives, Part 2, *Principles and rules for the structure and drafting of ISO and IEC documents*,
177 <https://www.iso.org/sites/directives/current/part2/index.xhtml>

178 Unified Modeling Language (UML) from the Open Management Group (OMG), <https://www.uml.org/>

179 3 Terms and definitions

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

182 The terms "shall" ("required"), "shall not," "should" ("recommended"), "should not" ("not recommended"),
183 "may," "need not" ("not required"), "can" and "cannot" in this document are to be interpreted as described
184 in [ISO/IEC Directives, Part 2](#), Annex H. The terms in parenthesis are alternatives for the preceding term,
185 for use in exceptional cases when the preceding term cannot be used for linguistic reasons. Note that
186 [ISO/IEC Directives, Part 2](#), Annex H specifies additional alternatives. Occurrences of such additional
187 alternatives shall be interpreted in their normal English meaning.

188 The terms "clause," "subclause," "paragraph," and "annex" in this document are to be interpreted as
189 described in [ISO/IEC Directives, Part 2](#), Clause 5.

190 The terms "normative" and "informative" in this document are to be interpreted as described in [ISO/IEC](#)
191 [Directives, Part 2](#), Clause 3. In this document, clauses, subclauses, or annexes labeled "(informative)" do
192 not contain normative content. Notes and examples are always informative elements.

193 The terms defined in [DSP0004](#) and [DSP1001](#) apply to this document. The following additional terms are
194 used in this document.

- 195 **3.1**
 196 **conditional**
 197 indicates requirements to be followed strictly to conform to the document when the specified conditions
 198 are met
- 199 **3.2**
 200 **mandatory**
 201 indicates requirements to be followed strictly to conform to the document and from which no deviation is
 202 permitted
- 203 **3.3**
 204 **optional**
 205 indicates a course of action permissible within the limits of the document
- 206 **3.4**
 207 **referencing profile**
 208 indicates a profile that owns the definition of this class and can include a reference to this profile in its
 209 “Related Profiles” table

210 4 Symbols and abbreviated terms

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

212 5 Synopsis

213 **Profile name:** Sensors

214 **Version:** 1.2.0

215 **Organization:** DMTF

216 **CIM Schema version:** 2.32

217 **Central class:** CIM_Sensor

218 **Scoping class:** CIM_ComputerSystem

219 The *Sensors Profile* extends the management capability of the referencing profiles by adding the
 220 capability to represent sensors in a managed system.

221 Table 1 identifies profiles related to this profile.

222

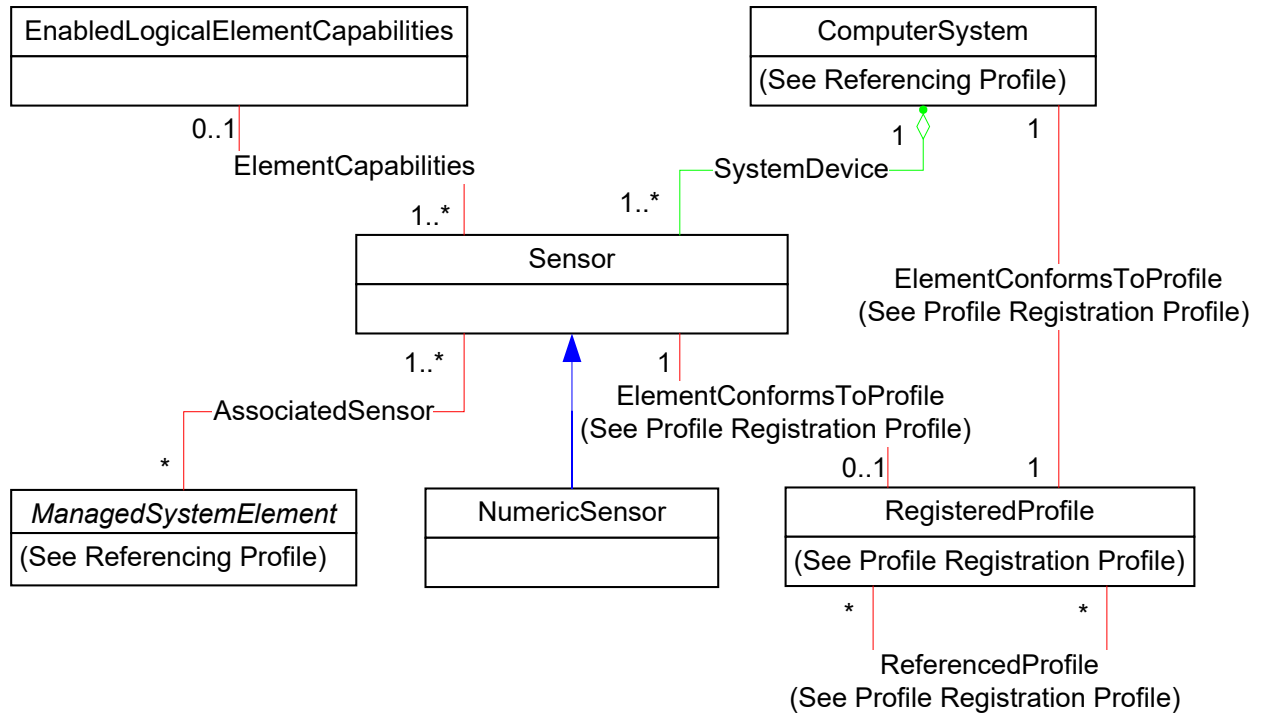
Table 1 – Related profiles

Profile Name	Organization	Version	Requirement	Description
Profile Registration	DMTF	1.0	Mandatory	None

223 **6 Description**

224 The *Sensors Profile* describes the properties and methods of sensors in a managed system.

225 Figure 1 represents the class schema for the *Sensors Profile*. For simplicity, the prefix CIM_ has been
 226 removed from the names of the classes.



227

228 **Figure 1 – Sensors Profile: Class diagram**

229 **7 Implementation requirements**

230 This section details the requirements related to the instantiation of instances and their properties for
 231 implementations of this profile. The requirements for the implementation of the methods are listed in 8,
 232 "Methods".

233 **7.1 CIM_Sensor and CIM_NumericSensor**

234 The implementation shall instantiate an instance of CIM_Sensor, including its subclass
 235 CIM_NumericSensor.

236 **7.2 CIM_Sensor.PossibleStates**

237 The CIM_Sensor.PossibleStates property shall represent an array of the possible string outputs of the
 238 sensor provided as a value of the CIM_Sensor.CurrentState property. The CIM_Sensor.SensorType
 239 property shall determine which CIM_Sensor.PossibleStates enumeration set to use.

240 The mappings between the CIM_Sensor.SensorType property values and the
 241 CIM_Sensor.PossibleStates property values are shown in Table 2. When the value of the
 242 CIM_Sensor.SensorType property matches a value in the "CIM_Sensor.SensorType" column of Table 2,
 243 the CIM_Sensor.PossibleStates property shall contain an array of values or an array of the subset of
 244 values specified in the corresponding "CIM_Sensor.PossibleStates" column. If the value of the
 245 CIM_Sensor.SensorType property is not listed in Table 2, the CIM_Sensor.PossibleStates property shall
 246 be defined by the implementation. The mapping between the values of CIM_Sensor.PossibleStates in
 247 Table 2 and the actual condition of the monitored device is implementation specific.

248 **Table 2 – CIM_Sensor.PossibleStates values for discrete sensors**

CIM_Sensor.SensorType	CIM_Sensor.PossibleStates
2 (Temperature)	"Bad", "Good", "Unknown"
3 (Voltage)	"Bad", "Good", "Unknown"
4 (Current)	"Bad", "Good", "Unknown"
5 (Tachometer)	"Bad", "Good", "Unknown"
7 (Switch)	"Closed", "Open", "Unknown"
8 (Lock)	"Locked", "Unlocked", "Unknown"
9 (Humidity)	"Humid", "Normal", "Unknown"
10 (Smoke Detection)	"Smoky", "Normal", "Unknown"
11 (Presence)	"Not Present", "Present", "Unknown"
12 (Air Flow)	"Bad", "Good", "Unknown"
13 (Power Consumption)	"Bad", "Good", "Unknown"
14 (Power Production)	"Bad", "Good", "Unknown"
15 (Pressure)	"Bad", "Good", "Unknown"
16 (GPS)	"Bad", "Good", "Unknown"

249 **7.3 CIM_NumericSensor.PossibleStates**

250 The CIM_NumericSensor.PossibleStates property shall represent an array of the possible string outputs
 251 of the sensor provided as a value of the CIM_NumericSensor.CurrentState property. The
 252 CIM_NumericSensor.SensorType property shall determine which CIM_NumericSensor.PossibleStates
 253 enumeration set to use.

254 The mappings between the CIM_NumericSensor.SensorType property values and the
 255 CIM_NumericSensor.PossibleStates property values are shown in Table 3. When the value of the
 256 CIM_NumericSensor.SensorType property matches a value in the “CIM_NumericSensor.SensorType”
 257 column of Table 3, the CIM_NumericSensor.PossibleStates property shall contain an array of values or
 258 an array of the subset of the values specified in the corresponding “CIM_NumericSensor.PossibleStates”
 259 column. If the value of the CIM_NumericSensor.SensorType property is not listed in Table 3, the
 260 CIM_NumericSensor.PossibleStates property shall be defined by the implementation. The mapping
 261 between the values of CIM_NumericSensor.PossibleStates in Table 3 and the actual condition of the
 262 monitored device is implementation specific.

263 **Table 3 – CIM_NumericSensor.PossibleStates values for numeric sensors**

CIM_NumericSensor.SensorType	CIM_NumericSensor.PossibleStates
2 (Temperature)	“Non-Critical”, “Lower Non-Critical”, “Upper Non-Critical”, “Critical”, “Lower Critical”, “Upper Critical”, “Fatal”, “Lower Fatal”, “Upper Fatal”, “Normal”, “Unknown”
3 (Voltage)	
4 (Current)	
5 (Tachometer)	
9 (Humidity)	
10 (Smoke Detection)	
12 (Air Flow)	
13 (Power Consumption)	
14 (Power Production)	
15 (Pressure)	

264 **7.4 CIM_Sensor.CurrentState and CIM_NumericSensor.CurrentState**

265 The CIM_Sensor.CurrentState property shall have a value of one of the elements in the
 266 CIM_Sensor.PossibleStates array.

267 The CIM_NumericSensor.CurrentState property shall have a value of one of the elements in the
 268 CIM_NumericSensor.PossibleStates array.

269 **7.5 CIM_NumericSensor.LowerThresholdNonCritical**

270 The CIM_NumericSensor.LowerThresholdNonCritical property shall be mandatory when the
 271 CIM_NumericSensor.SupportedThresholds array contains a value of 0 (LowerThresholdNonCritical).

272 The CIM_NumericSensor.LowerThresholdNonCritical property shall be settable only if the
 273 CIM_NumericSensor.SettableThresholds array contains a value of 0 (LowerThresholdNonCritical).

274 **7.6 CIM_NumericSensor.UpperThresholdNonCritical**

275 The CIM_NumericSensor.UpperThresholdNonCritical property shall be mandatory when the
 276 CIM_NumericSensor.SupportedThresholds array contains a value of 1 (UpperThresholdNonCritical).

277 The CIM_NumericSensor.UpperThresholdNonCritical property shall be settable only if the
 278 CIM_NumericSensor.SettableThresholds array contains a value of 1 (UpperThresholdNonCritical).

279 **7.7 CIM_NumericSensor.LowerThresholdCritical**

280 The CIM_NumericSensor.LowerThresholdCritical property shall be mandatory when the
 281 CIM_NumericSensor.SupportedThresholds array contains a value of 2 (LowerThresholdCritical).

282 The CIM_NumericSensor.LowerThresholdCritical property shall be settable only if the
283 CIM_NumericSensor.SettableThresholds array contains a value of 2 (LowerThresholdCritical).

284 **7.8 CIM_NumericSensor.UpperThresholdCritical**

285 The CIM_NumericSensor.UpperThresholdCritical property shall be mandatory when the
286 CIM_NumericSensor.SupportedThresholds array contains a value of 3 (UpperThresholdCritical).

287 The CIM_NumericSensor.UpperThresholdCritical property shall be settable only if the
288 CIM_NumericSensor.SettableThresholds array contains a value of 3 (UpperThresholdCritical).

289 **7.9 CIM_NumericSensor.LowerThresholdFatal**

290 The CIM_NumericSensor.LowerThresholdFatal property shall be mandatory when the
291 CIM_NumericSensor.SupportedThresholds array contains a value of 4 (LowerThresholdFatal).

292 The CIM_NumericSensor.LowerThresholdFatal property shall be settable only if the
293 CIM_NumericSensor.SettableThresholds array contains a value of 4 (LowerThresholdFatal).

294 **7.10 CIM_NumericSensor.UpperThresholdFatal**

295 The CIM_NumericSensor.UpperThresholdFatal property shall be mandatory when the
296 CIM_NumericSensor.SupportedThresholds array contains a value of 5 (UpperThresholdFatal).

297 The CIM_NumericSensor.UpperThresholdFatal property shall be settable only if the
298 CIM_NumericSensor.SettableThresholds array contains a value of 5 (UpperThresholdFatal).

299 **7.11 CIM_NumericSensor.SupportedThresholds**

300 The CIM_NumericSensor.SupportedThresholds property is an array that contains the list of the
301 implemented thresholds: LowerThresholdNonCritical, UpperThresholdNonCritical,
302 LowerThresholdCritical, UpperThresholdCritical, LowerThresholdFatal, and UpperThresholdFatal. When
303 the implementation does not support any of these threshold properties, the
304 CIM_NumericSensor.SupportedThresholds property shall be an empty array.

305 **7.12 CIM_NumericSensor.SettableThresholds**

306 The CIM_NumericSensor.SettableThresholds property is an array that contains the list of the settable
307 implemented thresholds: LowerThresholdNonCritical, UpperThresholdNonCritical,
308 LowerThresholdCritical, UpperThresholdCritical, LowerThresholdFatal, and UpperThresholdFatal. The
309 CIM_NumericSensor.SettableThresholds array shall contain the subset of values in the
310 CIM_NumericSensor.SupportedThresholds array (see 7.11). When the implementation does not support
311 any of the settable threshold properties, the CIM_NumericSensor.SettableThresholds property shall be an
312 empty array.

313 **7.13 CIM_EnabledLogicalElementCapabilities**

314 When the CIM_EnabledLogicalElementCapabilities class is instantiated, the instance of
315 CIM_EnabledLogicalElementCapabilities shall be associated with the Central Instance through the
316 CIM_ElementCapabilities association and used for advertising the capabilities of the Central Instance.

317 At most only one instance of CIM_EnabledLogicalElementCapabilities shall be associated with a given
318 instance of CIM_Sensor or CIM_NumericSensor.

319 **7.13.1 CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported**

320 The CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported property is an array that
321 contains the supported requested states for the instance of CIM_Sensor or CIM_NumericSensor. This
322 property shall be the super set of the values to be used as the RequestedState parameter in the
323 RequestStateChange() method.

324 The value of the CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported property shall be
325 an empty array or contain any combination of the following values: 2 (Enabled), 3 (Disabled), or
326 11 (Reset).

327 **7.13.2 CIM_EnabledLogicalElementCapabilities.ElementNameEditSupported**

328 The CIM_EnabledLogicalElementCapabilities.ElementNameEditSupported property shall have a value of
329 TRUE when the implementation supports client modification of the ElementName property of the
330 associated CIM_Sensor or CIM_NumericSensor instance.

331 **7.13.3 CIM_EnabledLogicalElement.MaxElementNameLen**

332 The CIM_EnabledLogicalElementCapabilities.MaxElementNameLen property shall be implemented when
333 the CIM_EnabledLogicalElementCapabilities.ElementNameEditSupported property has a value of TRUE.

334 **7.14 Sensor state management**

335 Sensor state management is optional. Sensor state management consists of the RequestStateChange()
336 method of the Central Instance being supported (see 8.1) and the RequestedState property of the Central
337 Instance having a value other than 12 (Not Applicable).

338 **7.14.1 Sensor state management support**

339 When no CIM_EnabledLogicalElementCapabilities instance is associated with the Central Instance,
340 sensor state management shall not be supported.

341 When a CIM_EnabledLogicalElementCapabilities instance is associated with the Central Instance but the
342 CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported property is an empty array, sensor
343 state management shall not be supported.

344 When a CIM_EnabledLogicalElementCapabilities instance is associated with the Central Instance and the
345 CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported property is a non-empty array,
346 sensor state management shall be supported.

347 **7.15 CIM_Sensor.RequestedState and CIM_NumericSensor.RequestedState**

348 The RequestedState property shall have a value of 12 (Not Applicable), a value of 5 (No Change), or a
349 value that is contained in the CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported
350 property array of the associated CIM_EnabledLogicalElementCapabilities instance (see 7.13.1).

351 When sensor state management is supported and the RequestStateChange() method is successfully
352 executed, the RequestedState property shall be set to the value of the RequestedState parameter of the
353 RequestStateChange() method. After the RequestStateChange() method completes successfully, the
354 RequestedState and EnabledState properties shall have equal values with the exception of the
355 transitional requested state 11 (Reset). The value of the RequestedState property may also change as a
356 result of the request for a change to the sensor's enabled state by a non-CIM implementation.

357 **7.15.1 RequestedState – 12 (Not Applicable)**

358 When sensor state management is not supported, the value of the RequestedState property of the
359 Central Instance shall be 12 (Not Applicable).

360 7.15.2 RequestedState – 5 (No Change)

361 When sensor state management is supported, the initial value of the RequestedState property of the
362 Central Instance shall be 5 (No Change).

363 7.16 CIM_Sensor.EnabledState and CIM_NumericSensor.EnabledState

364 Table 4 describes the mapping between the EnabledState property values and the corresponding
365 description of the state of the sensor. The EnabledState property shall match the values that are specified
366 in Table 4. The value of the EnabledState property may change as a result of a client execution of the
367 RequestStateChange() method or a change to the sensor's enabled state by a non-CIM implementation.

368

Table 4 – EnabledState value description

Value	Description	Extended Description
2	Enabled	Sensor shall be operational.
3	Disabled	Sensor shall be disabled.
5	Not Applicable	Sensor state is indeterminate, or sensor state management is not supported.

369 7.17 CIM_Sensor.OtherSensorTypeDescription and 370 CIM_NumericSensor.OtherSensorTypeDescription

371 The OtherSensorTypeDescription property shall be mandatory when the SensorType property is set to a
372 value of 1 (Other).

373 The OtherSensorTypeDescription property shall be formatted as a free-formed string of variable length
374 (pattern ".*").

375 7.18 CIM_SystemDevice and CIM_AssociatedSensor

376 When the Central Instance represents a sensor for the entire managed system, the instance of
377 CIM_ComputerSystem that is referenced by CIM_SystemDevice shall identify the managed system and
378 no instances of CIM_AssociatedSensor shall reference the Central Instance.

379 When the Central Instance represents a sensor for one or more discrete components and not for the
380 entire system, for each component an instance of CIM_AssociatedSensor shall reference the Central
381 Instance and the CIM_ManagedElement that identifies the component.

382 7.19 CIM_Sensor.ElementName and CIM_NumericSensor.ElementName

383 The ElementName property shall be formatted as a free-formed string of variable length (pattern ".*").

384 7.19.1 ElementName is modifiable

385 Implementations may allow the ElementName property to be modifiable by a client. This behavior is
386 conditional. This section describes the CIM elements and behavior requirements when an implementation
387 supports client modification of the ElementName property.

388 **7.20 CIM_Sensor.SensorContext**

389 The SensorContext property indicates the purpose and context of the sensor. The property may indicate
 390 what entity is being monitored or where the sensor is installed.

391 The property value shall be formatted using the following algorithm: <OrgID>:<LocalID> where <OrgID>
 392 and <LocalID> are separated by a colon (:). <OrgID> shall include a unique name that is owned by the
 393 business entity creating or defining the SensorContext and <OrgID> shall not contain a colon (:).
 394 <LocalID> is chosen by the business entity and should not be reused to identify different underlying (real-
 395 world) elements.

396 For DMTF defined identifiers, the <OrgID> shall match "DMTF". Table 5 shows the values of
 397 SensorContext defined in this profile.

398 **Table 5 – Sensor Context Values**

SensorContext	Description
DMTF:Inlet	Sensor located at inlet
DMTF:Exhaust	Sensor located at exhaust
DMTF:CPU	Sensor monitoring a CPU
DMTF:Base Board	Sensor monitoring base board

399 **8 Methods**

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

402 **8.1 CIM_Sensor.RequestStateChange() and**
 403 **CIM_NumericSensor.RequestStateChange()**

404 The RequestStateChange() method shall be implemented when sensor state management is supported
 405 (see 7.14.1).

406 Invocation of the RequestStateChange() method shall change the element's state to the value specified
 407 in the RequestedState parameter.

408 RequestStateChange() return code values are specified in Table 6. RequestStateChange() parameters
 409 are specified in Table 7.

410 Invoking the RequestStateChange() method multiple times could result in the earlier requests being
 411 overwritten or lost.

412 No standard messages are defined for this method.

413 **Table 6 – RequestStateChange() method: Return code values**

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

414

Table 7 – RequestStateChange() method: Parameters

Qualifiers	Name	Type	Description/Values
IN, REQ	RequestedState	uint16	Shall have a value from the CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported array (see 7.14)
OUT	Job	CIM_ConcreteJob REF	Returned if job started
IN, REQ	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

415 8.2 CIM_NumericSensor.RestoreDefaultThresholds()

416 The CIM_NumericSensor.RestoreDefaultThresholds() method shall be implemented and shall not return
417 a value of 1 (Unsupported) when the CIM_NumericSensor.SettableThresholds property is a non-empty
418 array (see 7.12).

419 Invocation of the CIM_NumericSensor.RestoreDefaultThresholds() method shall reset the values of the
420 thresholds of the sensor represented by the instance of CIM_NumericSensor to the hardware defaults.

421 Detailed requirements of the CIM_NumericSensor.RestoreDefaultThresholds() method are specified in
422 Table 8.

423 No method parameters are defined for this method.

424 No standard messages are defined for this method.

425 **Table 8 – CIM_NumericSensor.RestoreDefaultThresholds() Method: Return code values**

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

426 8.3 Profile conventions for operations

427 Support for operations for each profile class (including associations) is specified in the following
428 subclauses. Each subclause includes either the statement “All operations in the default list in 8.3 are
429 supported as described by [DSP0200](#) or a table listing all of the operations that are not supported by this
430 profile or where the profile requires behavior other than that described by [DSP0200](#).

431 The default list of operations is as follows:

- 432 • GetInstance
- 433 • EnumerateInstances
- 434 • EnumerateInstanceNames
- 435 • Associators
- 436 • AssociatorNames
- 437 • References
- 438 • ReferenceNames

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

441 **8.4 CIM_Sensor**

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

444 **Table 9 – Operations: CIM_Sensor**

Operation	Requirement	Messages
ModifyInstance	Optional. See 8.4.1.	None

445 **8.4.1 CIM_Sensor—ModifyInstance**

446 This section details the requirements for the ModifyInstance operation applied to an instance of
 447 CIM_Sensor. The ModifyInstance operation may be supported.

448 The ModifyInstance operation shall be supported when the ElementNameEditSupported property of the
 449 CIM_EnabledLogicalElementCapabilities instance that is associated with the CIM_Sensor instance has a
 450 value of TRUE.

451 **8.4.1.1 CIM_Sensor.ElementName**

452 When the ElementNameEditSupported property of the CIM_EnabledLogicalElementCapabilities instance
 453 that is associated with the CIM_Sensor instance has a value of TRUE, the implementation shall allow the
 454 ModifyInstance operation to change the value of the ElementName property of the CIM_Sensor instance.
 455 The ModifyInstance operation shall enforce the length restriction specified in the MaxElementNameLen
 456 property of the CIM_EnabledLogicalElementCapabilities instance.

457 When the ElementNameEditSupported property of the CIM_EnabledLogicalElementCapabilities instance
 458 has a value of FALSE, the implementation shall not allow the ModifyInstance operation to change the
 459 value of the ElementName property of the CIM_Sensor instance.

460 **8.5 CIM_NumericSensor**

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

463 **Table 10 – Operations: CIM_NumericSensor**

Operation	Requirement	Messages
ModifyInstance	Optional. See 8.5.1.	None

464 **8.5.1 CIM_NumericSensor—ModifyInstance**

465 This section details the requirements for the ModifyInstance operation applied to an instance of
 466 CIM_NumericSensor. The ModifyInstance operation may be supported.

467 The ModifyInstance operation shall be supported when the ElementNameEditSupported property of the
 468 CIM_EnabledLogicalElementCapabilities instance that is associated with the CIM_NumericSensor
 469 instance has a value of TRUE.

470 The ModifyInstance operation shall be supported when CIM_NumericSensor.SettableThresholds property
 471 is a non-empty array. The ModifyInstance operation shall modify the following properties:

- 472 • LowerThresholdNonCritical when CIM_NumericSensor.SettableThresholds contains a value of 0
473 (LowerThresholdNonCritical)
- 474 • UpperThresholdNonCritical when CIM_NumericSensor.SettableThresholds contains a value of 1
475 (UpperThresholdNonCritical)
- 476 • LowerThresholdCritical when CIM_NumericSensor.SettableThresholds contains a value of 2
477 (LowerThresholdCritical)
- 478 • UpperThresholdCritical when CIM_NumericSensor.SettableThresholds contains a value of 3
479 (UpperThresholdCritical)
- 480 • LowerThresholdFatal when CIM_NumericSensor.SettableThresholds contains a value of 4
481 (LowerThresholdFatal)
- 482 • UpperThresholdFatal when CIM_NumericSensor.SettableThresholds contains a value of 5
483 (UpperThresholdFatal)

484 8.5.1.1 CIM_NumericSensor.ElementName

485 When the ElementNameEditSupported property of the CIM_EnabledLogicalElementCapabilities instance
486 that is associated with the CIM_NumericSensor instance has a value of TRUE, the implementation shall
487 allow the ModifyInstance operation to change the value of the ElementName property of the
488 CIM_NumericSensor instance. The ModifyInstance operation shall enforce the length restriction specified
489 in the MaxElementNameLen property of the CIM_EnabledLogicalElementCapabilities instance.

490 When the ElementNameEditSupported property of the CIM_EnabledLogicalElementCapabilities instance
491 has a value of FALSE, the implementation shall not allow the ModifyInstance operation to change the
492 value of the ElementName property of the CIM_NumericSensor instance.

493 8.6 CIM_EnabledLogicalElementCapabilities

494 All operations in the default list in 8.3 are supported as described by [DSP0200](#).

495 8.7 CIM_ElementCapabilities

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

498 **Table 11 – Operations: CIM_ElementCapabilities**

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

499 **8.8 CIM_SystemDevice**

500 Table 12 lists operations that either have special requirements beyond those from [DSP0200](#) or shall not
501 be supported.

502 **Table 12 – Operations: CIM_SystemDevice**

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

503 **8.9 CIM_AssociatedSensor**

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

506 **Table 13 – Operations: CIM_AssociatedSensor**

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

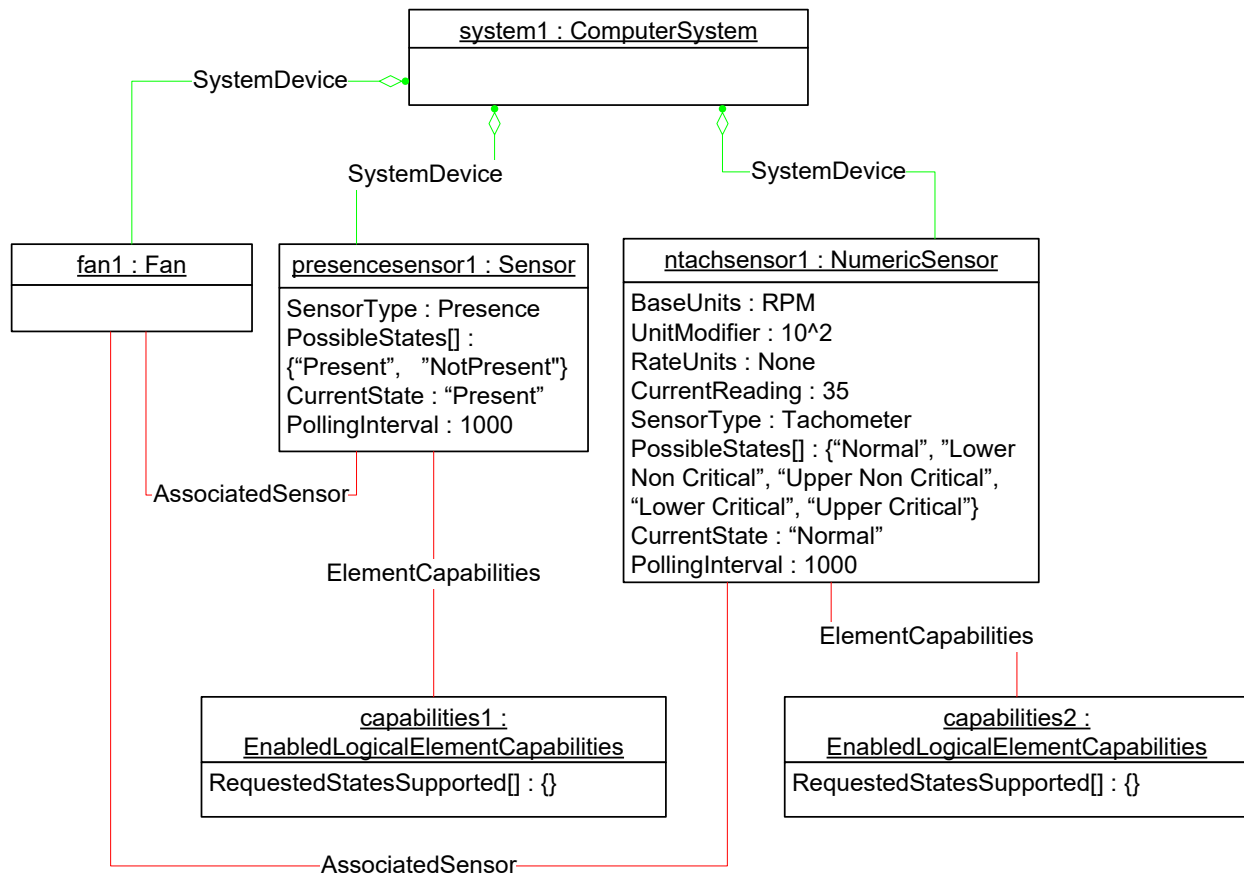
507 **9 Use cases**

508 All use cases are based on the implementation conformance to the DMTF *Sensors Profile*.

509 **9.1 Object diagrams**

510 Figure 2 represents a possible instantiation of the *Sensors Profile* classes. In the diagram, the instances
511 of CIM_Sensor and CIM_NumericSensor are associated with an instance of CIM_Fan through instances
512 of CIM_AssociatedSensor. In other cases, different instances of concrete classes derived from
513 CIM_ManagedElement can be associated through CIM_AssociatedSensor with an instance of
514 CIM_Sensor or CIM_NumericSensor.

515 Based on the SensorType property of the CIM_Sensor instance, presencesensor1 is a Presence sensor.
516 presencesensor1 is a discrete sensor and provides the value “Present” or “Not Present” based on the
517 PossibleStates property. Based on the SensorType property, ntachsensorn1 is a Tachometer sensor,
518 which is an analog sensor, and provides numeric reading of the fan1 speed. Based on the BaseUnits
519 property, the reading is in RPM units, and the RateUnit property shows no additional units. The
520 CurrentReading property in this object diagram has a value of 35, which is multiplied by the value of the
521 UnitModifier property (in this case 10^2), and therefore has a value of 3500 RPM.



522

523

Figure 2 – Sensors Profile: Object diagram

524 9.2 Show all current states of the monitored devices in the computer system

525 Starting from the CIM_ComputerSystem instance that represents the computer system, select all of the
 526 CIM_Sensor and CIM_NumericSensor instances that are associated through instances of
 527 CIM_SystemDevice, which represent the sensors of the computer system. Iterate through the instances
 528 and get the CIM_Sensor.CurrentState and CIM_NumericSensor.CurrentState property value, which will
 529 represent the states of the monitored devices.

530 9.3 Find sensors associated with a specific component

531 Select all of the CIM_Sensor and CIM_NumericSensor instances that are associated with the instance of
 532 a subclass of CIM_ManagedElement through an instance of CIM_AssociatedSensor.

533 9.4 Change upper non-critical threshold of numeric sensor

534 Select the instance of CIM_NumericSensor. Determine whether the SettableThresholds property contains
 535 a value of 1 (UpperThresholdNonCritical). If so, set the value of the UpperThresholdNonCritical property;
 536 otherwise, return an appropriate error.

537 9.5 Reset sensor to threshold defaults

538 Select the instance of CIM_NumericSensor. Execute the method RestoreDefaultThresholds().

539 **9.6 Determining whether ElementName can be modified**

540 For a given instance of CIM_Sensor or CIM_NumericSensor, a client can determine whether it can modify
 541 the ElementName as follows:

- 542 1) Find the CIM_EnabledLogicalElementCapabilities instance that is associated with the target
 543 instance.
- 544 2) Query the value of the ElementNameEditSupported property of the
 545 CIM_EnabledLogicalElementCapabilities instance. If the value is TRUE, the client can modify
 546 the ElementName property of the target instance.

547 **9.7 Determining whether state management is supported**

- 548 1) For a given instance of CIM_Sensor or CIM_NumericSensor, a client can determine whether
 549 state management is supported as follows:
- 550 2) Find the CIM_EnabledLogicalElementCapabilities instance that is associated with the instance.
- 551 3) Query the value of the RequestedStatesSupported property. If at least one value is specified,
 552 state management is supported.

553 **9.8 Find inlet temperature sensor instance**

554 Enumerate the instances of CIM_NumericSensor. Determine whether the SensorContext property
 555 contains a value of "DMTF:Inlet" and the SensorType has value 2 (Temperature) for one of the instance
 556 of CIM_NumericSensor. If so, that instance represents the sensor for the inlet.

557

558 **10 CIM Elements**

559 Table 14 shows the list of CIM Elements for this profile and details their requirements. The
 560 implementation requirements for the classes and properties described in this section are defined in 7,
 561 "Implementation Requirements".

562

Table 14 – CIM Elements: Sensors Profile Profile

Element Name	Requirement	Description
Classes		
CIM_Sensor	Conditional	See 7.1 and 10.1.
CIM_NumericSensor	Conditional	See 7.1 and 10.2.
CIM_EnabledLogicalElementCapabilities	Optional	See 7.13 and 10.3.
CIM_ElementCapabilities	Optional	See 10.4.
CIM_SystemDevice	Mandatory	See 7.18 and 10.5.
CIM_AssociatedSensor	Optional	See 7.18 and 10.6.
CIM_RegisteredProfile	Mandatory	See 10.7.
Indications		
None defined in this profile		

563 **10.1 CIM_Sensor**

564 CIM_Sensor is used to represent a discrete sensor. The CIM_Sensor class is mandatory if the
 565 CIM_NumericSensor class is not implemented. Table 15 provides information about the properties of
 566 CIM_Sensor.

567

Table 15 – Class: CIM_Sensor

Properties and Methods	Requirement	Notes
SystemCreationClassName	Mandatory	Key
SystemName	Mandatory	Key
CreationClassName	Mandatory	Key
DeviceID	Mandatory	Key
SensorType	Mandatory	None
PossibleStates	Mandatory	See 7.2.
CurrentState	Mandatory	See 7.4.
ElementName	Mandatory	See 7.13.2.
OtherSensorTypeDescription	Conditional	See 7.17.
EnabledState	Mandatory	See 7.16.
SensorContext	Optional	See 7.20.
RequestedState	Mandatory	See 7.14.
OperationalStatus	Mandatory	None
HealthState	Mandatory	None
RequestStateChange()	Conditional	See 8.1.

568 **10.2 CIM_NumericSensor**

569 CIM_NumericSensor is used to represent an analog sensor. The CIM_NumericSensor class is mandatory
 570 when the CIM_Sensor class is not implemented. Table 16 provides information about the properties of
 571 CIM_NumericSensor.

572 **Table 16 – Class: CIM_NumericSensor**

Properties and Methods	Requirement	Notes
SystemCreationClassName	Mandatory	Key
SystemName	Mandatory	Key
CreationClassName	Mandatory	Key
DeviceID	Mandatory	Key
BaseUnits	Mandatory	None
UnitModifier	Mandatory	None
RateUnits	Mandatory	None
CurrentReading	Mandatory	None
LowerThresholdNonCritical	Conditional	See 7.5.
UpperThresholdNonCritical	Conditional	See 7.6.
LowerThresholdCritical	Conditional	See 7.7.
UpperThresholdCritical	Conditional	See 7.8.
LowerThresholdFatal	Conditional	See 7.9.
UpperThresholdFatal	Conditional	See 7.10.
SupportedThresholds	Mandatory	See 7.11.
SettableThresholds	Mandatory	See 7.12.
SensorType	Mandatory	None
PossibleStates	Mandatory	See 7.3.
CurrentState	Mandatory	See 7.4.
ElementName	Mandatory	See 7.13.2.
OtherSensorTypeDescription	Conditional	See 7.17.
SensorContext	Optional	See 7.20.
EnabledState	Mandatory	See 7.16.
RequestedState	Mandatory	See 7.14.
OperationalStatus	Mandatory	None
HealthState	Mandatory	None
RequestStateChange()	Conditional	See 8.1.
RestoreDefaultThresholds()	Conditional	See 8.2.

573 **10.3 CIM_EnabledLogicalElementCapabilities**

574 CIM_EnabledLogicalElementCapabilities is used to represent the capabilities of the sensor as it applies to
 575 the properties of CIM_Sensor or CIM_NumericSensor that are derived from CIM_EnabledLogicalElement,
 576 such as RequestedEnabledState. For implementation details, see 7.13.

577 Table 17 provides information about the properties of CIM_EnabledLogicalElementCapabilities.

578 **Table 17 – Class: CIM_EnabledLogicalElementCapabilities**

Properties	Requirement	Notes
InstanceID	Mandatory	Key
RequestedStatesSupported	Mandatory	See 7.13.1.
ElementNameEditSupported	Mandatory	See 7.13.2.
MaxElementNameLen	Conditional	See 7.13.3.

579 **10.4 CIM_ElementCapabilities**

580 CIM_ElementCapabilities is used to associate CIM_Sensor or CIM_NumericSensor with an instance of
 581 CIM_EnabledLogicalElementCapabilities that describes the capabilities of CIM_Sensor or
 582 CIM_NumericSensor. Table 18 provides information about the properties of CIM_ElementCapabilities.

583 **Table 18 – Class: CIM_ElementCapabilities**

Properties	Requirement	Notes
ManagedElement	Mandatory	Key
Capabilities	Mandatory	Key. See 7.13.

584 **10.5 CIM_SystemDevice**

585 CIM_SystemDevice is used to associate the instance of CIM_Sensor or CIM_NumericSensor with the
 586 instance of CIM_ComputerSystem of which the CIM_Sensor instance is a member. Table 19 provides
 587 information about the properties of CIM_SystemDevice.

588 **Table 19 – Class: CIM_SystemDevice**

Properties	Requirement	Notes
GroupComponent	Mandatory	Key: shall be a reference to the CIM_ComputerSystem instance of which the current CIM_Sensor or CIM_NumericSensor instance is a member.
PartComponent	Mandatory	Key: shall be a reference to the current CIM_Sensor or CIM_NumericSensor instance.

589 **10.6 CIM_AssociatedSensor**

590 CIM_AssociatedSensor is used to associate the instance of CIM_Sensor or CIM_NumericSensor with the
 591 instance of a subclass of CIM_ManagedElement. Table 20 provides information about the properties of
 592 CIM_AssociatedSensor.

593 **Table 20 – Class: CIM_AssociatedSensor**

Properties	Requirement	Notes
Antecedent	Mandatory	shall be a reference to a specific instance of CIM_Sensor or CIM_NumericSensor.
Dependent	Mandatory	shall be a reference to the instance of a subclass of CIM_ManagedElement that the sensor is monitoring.

594 **10.7 CIM_RegisteredProfile**

595 CIM_RegisteredProfile is defined by the Profile Registration Profile. The requirements denoted in
 596 Table 21 are in addition to those mandated by the Profile Registration Profile.

597 **Table 21 – Class: CIM_RegisteredProfile**

Properties	Requirement	Description
RegisteredName	Mandatory	This property shall have a value of "Sensors".
RegisteredVersion	Mandatory	This property shall have a value of "1.2.0".
RegisteredOrganization	Mandatory	This property shall have a value of 2 (DMTF).

598 NOTE: Previous versions of this document included the suffix 'Profile' for the RegisteredName value. If
 599 implementations querying for the RegisteredName value find the suffix 'Profile', they should ignore the suffix, with any
 600 surrounding white spaces, before any comparison is done with the value as specified in this document.

601
602
603
604
605

ANNEX A (informative)

Change log

Version	Date	Description
1.0.0c	2006-05-16	Initial Preliminary Release
1.0.0	2007-11-06	Final Specification Release
1.0.1	2008-09-25	Errata 1.0.1
1.0.2	2009-10-05	Errata 1.0.2
1.1.0	2012-04-30	Added SensorContext property
1.1.1	2014-04-25	Errata removing unnecessary reference to DSP0223
1.2.0	2024-01-05	Add GPS sensor type