



1

2

3

4

**Document Identifier: DSP1108**

**Date: 2017-03-18**

**Version: 1.0.2**

5

## **Physical Computer System View Profile**

6

**Supersedes: 1.0.1**

7

**Document Class: Normative**

8

**Document Status: Published**

9

**Document Language: en-US**

10 Copyright Notice

11 Copyright © 2017 Distributed Management Task Force, Inc. (DMTF). All rights reserved.

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

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26 withdrawn or modified after publication, and shall be indemnified and held harmless by any party  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

29 For information about patents held by third-parties which have notified 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>.

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 .....	8
39	4 Symbols and abbreviated terms.....	9
40	5 Synopsis .....	10
41	6 Description .....	11
42	7 Implementation.....	11
43	7.1 Representing a physical computer system view.....	11
44	8 Methods.....	22
45	8.1 CIM_PhysicalComputerSystemView.RequestStateChange( ) .....	22
46	8.2 CIM_PhysicalComputerSystemView.ClearLog( ) .....	23
47	8.3 CIM_PhysicalComputerSystemView.InstallSoftwareFromURI( ) .....	23
48	8.4 CIM_PhysicalComputerSystemView.SetOneTimeBootSource( ) .....	24
49	8.5 Profile conventions for operations .....	25
50	8.6 CIM_PhysicalComputerSystemView .....	25
51	8.7 CIM_ElementView .....	25
52	9 Use cases.....	26
53	9.1 Miscellaneous object diagrams.....	26
54	9.2 Small footprint managed computer system with additional CIM Schema implemented.....	26
55	9.3 Large managed computer system with additional CIM Profiles implemented.....	27
56	9.4 Managed system exposing multiple temperature sensors .....	28
57	9.5 Determine the enabled state of the physical computer system .....	30
58	9.6 Change the enabled state of the physical computer system .....	30
59	9.7 Get properties of a specific record log of the physical computer system .....	30
60	9.8 Browse the records of a log of the physical computer system .....	30
61	9.9 Monitor temperature sensor readings of the physical computer system .....	31
62	9.10 Configure a source of the physical computer system for next reboot only.....	31
63	9.11 Update the BIOS firmware of the physical computer system .....	31
64	10 CIM Elements.....	32
65	10.1 CIM_PhysicalComputerSystemView .....	32
66	10.2 CIM_RegisteredProfile.....	34
67	10.3 CIM_ElementView .....	35
68	10.4 CIM_ElementConformsToProfile .....	35
69	ANNEX A (informative) Change log.....	36

## Figures

72	Figure 1 – Physical Computer System View Profile: Class diagram .....	11
73	Figure 2 – Simple PhysicalComputerSystemView implementation .....	26
74	Figure 3 – Small footprint PhysicalComputerSystemView implementation .....	27
75	Figure 4 – Rich PhysicalComputerSystemView implementation .....	28
76	Figure 5 – PhysicalComputerSystemView implementation of multiple temperature sensors .....	29

79 **Tables**

80	Table 1 – Referenced profiles .....	10
81	Table 2 – CIM_PhysicalComputerSystemView property model correspondence .....	13
82	Table 3 – Property origins for processor .....	17
83	Table 4 – Property origins for memory .....	17
84	Table 5 – Property origins for current BIOS or EFI firmware .....	18
85	Table 6 – Property origins for current management firmware .....	18
86	Table 7 – Property origins for the operating system .....	19
87	Table 8 – Property origins for the power allocation limit .....	19
88	Table 9 – Property origins for numeric sensors .....	20
89	Table 10 – Property origins for record logs .....	21
90	Table 11 – Property origin for boot sources .....	21
91	Table 12 – CIM_PhysicalComputerSystemView.RequestStateChange( ) method: Return code values ...	22
92	Table 13 – CIM_PhysicalComputerSystemView.RequestStateChange( ) method: Parameters .....	23
93	Table 14 – CIM_PhysicalComputerSystemView.ClearLog( ) method: Return code values .....	23
94	Table 15 – CIM_PhysicalComputerSystemView.ClearLog( ) method: Parameters .....	23
95	Table 16 – CIM_PhysicalComputerSystemView.InstallSoftwareFromURI( ) method: Return code values	24
96	Table 17 – CIM_PhysicalComputerSystemView.InstallSoftwareFromURI( ) method: Parameters.....	24
97	Table 18 – CIM_PhysicalComputerSystemView.SetOneTimeBootSource( ) method: Return code values	25
98	Table 19 – CIM_PhysicalComputerSystemView.SetOneTimeBootSource( ) method: Parameters .....	25
99	Table 20 – Operations: CIM_PhysicalComputerSystemView .....	25
100	Table 21 – CIM Elements: Physical Computer System View Profile .....	32
101	Table 22 – Class: CIM_PhysicalComputerSystemView .....	32
102	Table 23 – Class: CIM_RegisteredProfile .....	34
103	Table 24 – Class: CIM_ElementView .....	35
104	Table 25 – Class: CIM_ElementConformsToProfile .....	35
105		

106

## Foreword

107 The *Physical Computer System View Profile* (DSP1108) was prepared by the Server Desktop Mobile  
108 Platforms Working Group of the DMTF.

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

## 111 Acknowledgments

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

113 Editors:

- 114 • Hemal Shah – Broadcom Limited
- 115 • Steve Lee – Microsoft Corporation

116 Contributors:

- 117 • Jeff Hilland – Hewlett-Packard Company
- 118 • John Leung – Intel Corporation
- 119 • Hemal Shah – Broadcom Corporation
- 120 • Satheesh Thomas – AMI
- 121 • Perry Vincent – Intel Corporation

122

123

## Introduction

124 The information in this specification should be sufficient for a provider or consumer of this data to identify  
125 unambiguously the classes, properties, methods, and values that shall be instantiated and manipulated to  
126 represent and manage a physical computer system and its associated management information.

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

### 129 Document conventions

#### 130 Typographical conventions

131 The following typographical conventions are used in this document:

- 132 • Document titles are marked in *italics*.
- 133 • Important terms that are used for the first time are marked in *italics*.
- 134 • Terms include a link to the term definition in the "Terms and definitions" clause, enabling easy  
135 navigation to the term definition.
- 136 • ABNF rules are in `monospaced font`.

#### 137 ABNF usage conventions

138 Format definitions in this document are specified using ABNF (see [RFC5234](#)), with the following  
139 deviations:

- 140 • Literal strings are to be interpreted as case-sensitive Unicode characters, as opposed to the  
141 definition in [RFC5234](#) that interprets literal strings as case-insensitive US-ASCII characters.

142

# Physical Computer System View Profile

## 143 1 Scope

144 The *Physical Computer System View Profile* describes a view of the management capability of  
145 referencing profiles by adding the capability to represent a physical computer system view of a managed  
146 computer system. This profile includes a specification of the physical computer system view, extrinsic  
147 methods for management operations, and its associated relationships to referencing profiles. This profile  
148 is not intended to provide all details of referenced profiles.

## 149 2 Normative references

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

154 DMTF DSP0004, *CIM Infrastructure Specification 2.6*,  
155 [http://www.dmtf.org/standards/published\\_documents/DSP0004\\_2.6.pdf](http://www.dmtf.org/standards/published_documents/DSP0004_2.6.pdf)

156 DMTF DSP0200, *CIM Operations over HTTP 1.3*,  
157 [http://www.dmtf.org/standards/published\\_documents/DSP0200\\_1.3.pdf](http://www.dmtf.org/standards/published_documents/DSP0200_1.3.pdf)

158 DMTF DSP0223, *Generic Operations 1.0*,  
159 [http://www.dmtf.org/standards/published\\_documents/DSP0223\\_1.0.pdf](http://www.dmtf.org/standards/published_documents/DSP0223_1.0.pdf)

160 DMTF DSP1001, *Management Profile Specification Usage Guide 1.0*,  
161 [http://www.dmtf.org/standards/published\\_documents/DSP1001\\_1.0.pdf](http://www.dmtf.org/standards/published_documents/DSP1001_1.0.pdf)

162 DMTF DSP1009, *Sensors Profile 1.0*,  
163 [http://dmf.org/sites/default/files/standards/documents/DSP1009\\_1.0.pdf](http://dmf.org/sites/default/files/standards/documents/DSP1009_1.0.pdf)

164 DMTF DSP1009, *Sensors Profile 1.1*,  
165 [http://dmf.org/sites/default/files/standards/documents/DSP1009\\_1.1.pdf](http://dmf.org/sites/default/files/standards/documents/DSP1009_1.1.pdf)

166 DMTF DSP1010, *Record Log Profile 1.0*,  
167 [http://www.dmtf.org/sites/default/files/standards/documents/DSP1010\\_1.0.pdf](http://www.dmtf.org/sites/default/files/standards/documents/DSP1010_1.0.pdf)

168 DMTF DSP1010, *Record Log Profile 2.0*,  
169 [http://www.dmtf.org/sites/default/files/standards/documents/DSP1010\\_2.0.pdf](http://www.dmtf.org/sites/default/files/standards/documents/DSP1010_2.0.pdf)

170 DMTF DSP1011, *Physical Asset Profile 1.0*,  
171 [http://www.dmtf.org/sites/default/files/standards/documents/DSP1011\\_1.0.pdf](http://www.dmtf.org/sites/default/files/standards/documents/DSP1011_1.0.pdf)

172 DMTF DSP1012, *Boot Control Profile 1.0*,  
173 [http://dmf.org/sites/default/files/standards/documents/DSP1012\\_1.0.pdf](http://dmf.org/sites/default/files/standards/documents/DSP1012_1.0.pdf)

174 DMTF DSP1022, *CPU Profile 1.0*,  
175 [http://dmf.org/sites/default/files/standards/documents/DSP1022\\_1.0.pdf](http://dmf.org/sites/default/files/standards/documents/DSP1022_1.0.pdf)

176 DMTF DSP1023, *Software Inventory Profile 1.0*,  
177 [http://dmf.org/sites/default/files/standards/documents/DSP1023\\_1.0.pdf](http://dmf.org/sites/default/files/standards/documents/DSP1023_1.0.pdf)

- 178 DMTF DSP1025, *Software Update Profile 1.0*,  
179 [http://dmtf.org/sites/default/files/standards/documents/DSP1025\\_1.0.pdf](http://dmtf.org/sites/default/files/standards/documents/DSP1025_1.0.pdf)
- 180 DMTF DSP1026, *System Memory Profile 1.0*,  
181 [http://dmtf.org/sites/default/files/standards/documents/DSP1026\\_1.0.pdf](http://dmtf.org/sites/default/files/standards/documents/DSP1026_1.0.pdf)
- 182 DMTF DSP1029, *OS Status Profile 1.0*,  
183 [http://dmtf.org/sites/default/files/standards/documents/DSP1029\\_1.0.pdf](http://dmtf.org/sites/default/files/standards/documents/DSP1029_1.0.pdf)
- 184 DMTF DSP1029, *OS Status Profile 1.1*,  
185 [http://dmtf.org/sites/default/files/standards/documents/DSP1029\\_1.1.pdf](http://dmtf.org/sites/default/files/standards/documents/DSP1029_1.1.pdf)
- 186 DMTF DSP1033, *Profile Registration Profile 1.0*,  
187 [http://www.dmtf.org/standards/published\\_documents/DSP1033\\_1.0.pdf](http://www.dmtf.org/standards/published_documents/DSP1033_1.0.pdf)
- 188 DMTF DSP1052, *Computer System Profile 1.0*,  
189 [http://www.dmtf.org/sites/default/files/standards/documents/DSP1052\\_1.0.pdf](http://www.dmtf.org/sites/default/files/standards/documents/DSP1052_1.0.pdf)
- 190 DMTF DSP1085, *Power Utilization Management Profile 1.0*,  
191 [http://dmtf.org/sites/default/files/standards/documents/DSP1085\\_1.0.pdf](http://dmtf.org/sites/default/files/standards/documents/DSP1085_1.0.pdf)
- 192 IETF RFC5234, *ABNF: Augmented BNF for Syntax Specifications, January 2008*,  
193 <http://tools.ietf.org/html/rfc5234>
- 194 ISO/IEC Directives, Part 2, *Rules for the structure and drafting of International Standards*,  
195 <http://isotc.iso.org/livelink/livelink.exe?func=ll&objId=4230456&objAction=browse&sort=subtype>
- 196

### 197 3 Terms and definitions

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

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

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

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

211 The terms defined in [DSP0004](#), [DSP0223](#), [DSP1001](#), and [DSP1033](#) apply to this document. The  
212 following additional terms are used in this document.

#### 213 3.1

##### 214 **conditional**

215 indicates requirements to be followed strictly to conform to the document when the specified conditions  
216 are met



- 217 **3.2**  
218 **mandatory**  
219 indicates requirements to be followed strictly to conform to the document and from which no deviation is  
220 permitted
- 221 **3.3**  
222 **optional**  
223 indicates a course of action permissible within the limits of the document
- 224 **3.4**  
225 **referencing profile**  
226 indicates a profile that owns the definition of this class and can include a reference to this profile in its  
227 "Referenced Profiles" table
- 228 **3.5**  
229 **unspecified**  
230 indicates that this profile does not define any constraints for the referenced CIM element or operation

## 231 **4 Symbols and abbreviated terms**

232 The abbreviations defined in [DSP0004](#), [DSP0223](#), and [DSP1001](#) apply to this document. The following  
233 additional abbreviations are used in this document.

- 234 **4.1**  
235 **BIOS**  
236 Basic Input Output System
- 237 **4.2**  
238 **EFI**  
239 Extensible Firmware Interface
- 240 **4.3**  
241 **FRU**  
242 Field Replaceable Unit
- 243 **4.4**  
244 **SKU**  
245 Stock Keeping Unit  
246

247 **5 Synopsis**248 **Profile name:** Physical Computer System View249 **Version:** 1.0.2250 **Organization:** DMTF251 **CIM schema version:** 2.36252 **Central class:** CIM\_PhysicalComputerSystemView253 **Scoping class:** CIM\_ComputerSystem

254 The *Physical Computer System View Profile* extends the management capability of referencing profiles  
 255 by adding the capability to represent a physical computer system view of a managed computer system.  
 256 This profile includes a specification of the physical computer system view, extrinsic methods for  
 257 management operations, and its associated relationships to referencing profiles.

258 Table 1 identifies profiles on which this profile has a dependency.

259

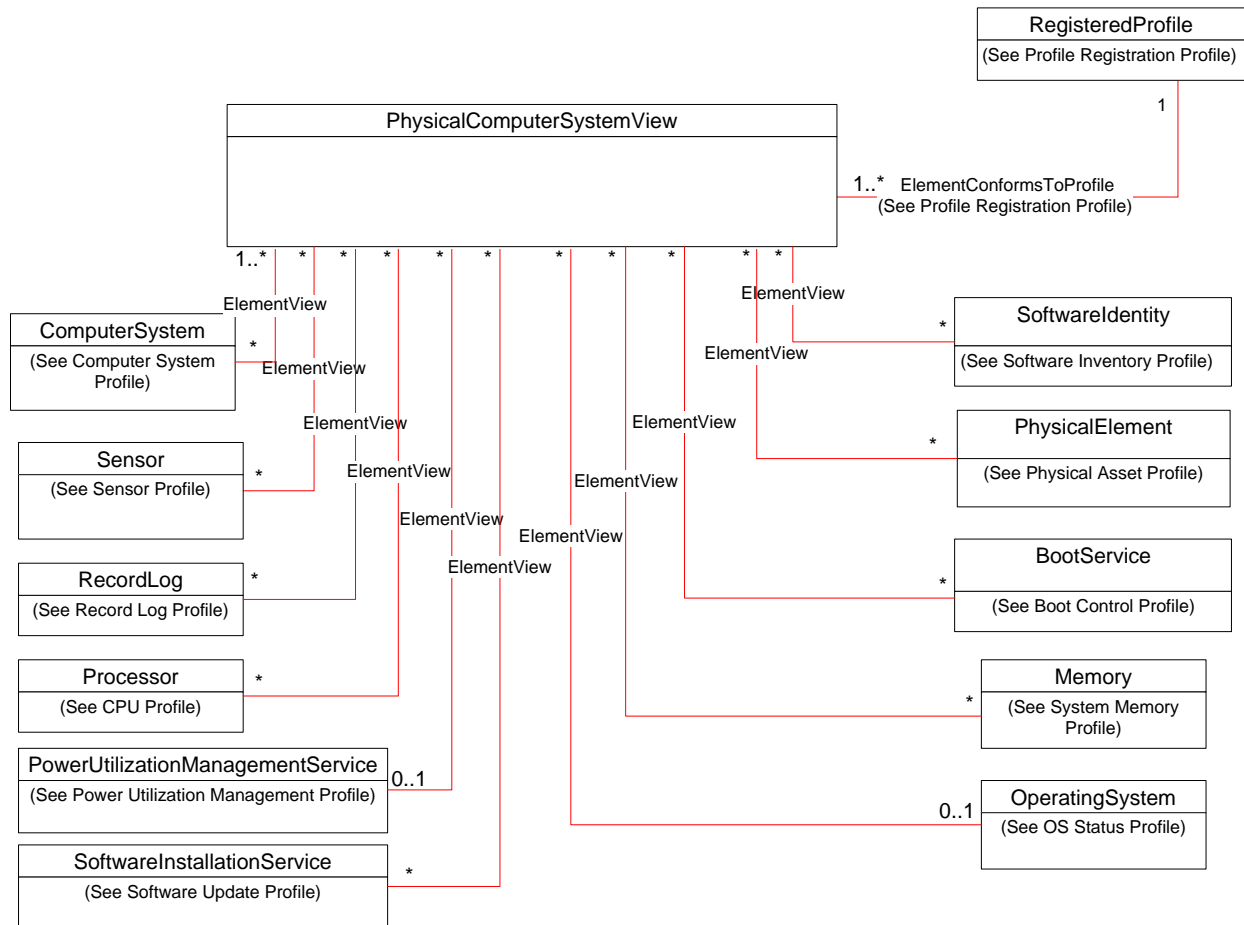
**Table 1 – Referenced profiles**

Profile Name	Organization	Version	Requirement	Description
Profile Registration	DMTF	1.0	Mandatory	None
Computer System	DMTF	1.0	Optional	None
Power Utilization Management	DMTF	1.0	Optional	None
Sensors	DMTF	1.0	Optional	None
Sensors	DMTF	1.1	Optional	None
Record Log	DMTF	1.0	Optional	None
Boot Control	DMTF	1.0	Optional	None
Software Inventory	DMTF	1.0	Optional	None
System Memory	DMTF	1.0	Optional	None
Physical Asset	DMTF	1.0	Optional	None
OS Status	DMTF	1.0	Optional	None
OS Status	DMTF	1.1	Optional	None
CPU	DMTF	1.0	Optional	None
Software Update	DMTF	1.0	Optional	None

## 260 6 Description

261 The *Physical Computer System View Profile* describes a physical computer system and associated  
 262 management information in a managed computer system.

263 Figure 1 represents the class schema for the *Physical Computer System View Profile*. For simplicity, the  
 264 CIM\_ prefix has been removed from the names of the classes.



265

266 **Figure 1 – Physical Computer System View Profile: Class diagram**

## 267 7 Implementation

268 This clause details the requirements related to the arrangement of instances and properties of instances  
 269 for implementations of this profile.

### 270 7.1 Representing a physical computer system view

271 A view of the managed computer system shall be represented by an instance of  
 272 CIM\_PhysicalComputerSystemView. The properties of the instance of  
 273 CIM\_PhysicalComputerSystemView shall reflect the current state and configuration of a managed  
 274 computer system conforming to the referenced profiles. When the ImplementedFeatures property of the  
 275 CIM\_RegisteredProfile instance contains the value listed in Table 2 below, the corresponding

276 CIM\_PhysicalComputerSystemView property shall conform to requirements for the corresponding  
277 property listed in the table as specified by the referenced profile.

### 278 **7.1.1 Representing information from multiple instances of the same class**

279 A subset of properties for CIM\_PhysicalComputerSystemView may represent information from multiple  
280 instances of same class. Indexed Arrays are used in CIM\_PhysicalComputerSystemView for such  
281 properties. The value in the array correspondence column in Table 2 identifies the class whose multiple  
282 instances are represented by the corresponding array property in CIM\_PhysicalComputerSystemView.

#### 283 **7.1.1.1 Properties of each instance**

284 The same index in properties with array correspondence with the same class shall represent a view of a  
285 single instance of that class.

#### 286 **7.1.1.2 Property correspondence**

287 The elements of properties with array correspondence with a class shall correspond to the respective  
288 properties of the corresponding instance of that class as in Table 2.

#### 289 **7.1.1.3 Matching property values to normalized instances**

290 For all properties with array correspondence with a class as in Table 2, the value at each index shall  
291 match the value of the corresponding property of the corresponding instance of that class.

**Table 2 – CIM\_PhysicalComputerSystemView property model correspondence**

Implemented Feature Value	CIM_Physical ComputerSystem View Property	Origin Class/Property or Model Correspondence	Referenced Profile
DMTF:Physical AssetView	FRUInfoSupported	CIM_PhysicalAssetCapabilities.FRUInfoSupported for CIM_Chassis	<a href="#">DSP1011 1.0</a> Clause 10.2
	Tag	CIM_Chassis.Tag	<a href="#">DSP1011 1.0</a> Clause 10.2
	Manufacturer	CIM_Chassis.Manufacturer	<a href="#">DSP1011 1.0</a> Clause 10.2
	Model	CIM_Chassis.Model	<a href="#">DSP1011 1.0</a> Clause 10.2
	SKU	CIM_Chassis.SKU	<a href="#">DSP1011 1.0</a> Clause 10.2
	SerialNumber	CIM_Chassis.SerialNumber	<a href="#">DSP1011 1.0</a> Clause 10.2
	Version	CIM_Chassis.Version	<a href="#">DSP1011 1.0</a> Clause 10.2
	PartNumber	CIM_Chassis.PartNumber	<a href="#">DSP1011 1.0</a> Clause 10.2
DMTF:CPUView	NumberOfProcessors	Number of CIM_Processor instances associated to associated CIM_ComputerSystem	<a href="#">DSP1022 1.0</a> Clause 10.11
	NumberOfProcessor Cores	CIM_ProcessorCapabilities.NumberOfProcessorCores	<a href="#">DSP1022 1.0</a> Clause 10.12
	NumberOfProcessor Threads	CIM_ProcessorCapabilities.NumberOfHardwareThreads	<a href="#">DSP1022 1.0</a> Clause 10.12
	ProcessorFamily	CIM_Processor.Family	<a href="#">DSP1022 1.0</a> Clause 10.11
	ProcessorCurrent ClockSpeed	CIM_Processor.CurrentClockSpeed	<a href="#">DSP1022 1.0</a> Clause 10.11
	ProcessorMaxClock Speed	CIM_Processor.MaxClockSpeed	<a href="#">DSP1022 1.0</a> Clause 10.11
DMTF:System MemoryView	MemoryBlockSize	CIM_Memory.BlockSize	<a href="#">DSP1026 1.0</a> Clause 10.3
	MemoryNumberOf Blocks	CIM_Memory.NumberOfBlocks	<a href="#">DSP1026 1.0</a> Clause 10.3
	MemoryConsumable Blocks	CIM_Memory.ConsumableBlocks	<a href="#">DSP1026 1.0</a> Clause 10.3
DMTF:Software InventoryView	CurrentBIOSMajor Version	CIM_SoftwareIdentity.MajorVersion representing the BIOS/EFI	<a href="#">DSP1023 1.0</a> Clause 10.1
	CurrentBIOSMinor Version	CIM_SoftwareIdentity.MinorVersion representing the BIOS/EFI	<a href="#">DSP1023 1.0</a> Clause 10.1
	CurrentBIOSRevision Number	CIM_SoftwareIdentity.Revision Number representing the BIOS/EFI	<a href="#">DSP1023 1.0</a> Clause 10.1

Implemented Feature Value	CIM_Physical ComputerSystem View Property	Origin Class/Property or Model Correspondence	Referenced Profile
	CurrentBIOSBuildNumber	CIM_SoftwareIdentity.BuildNumber representing the BIOS/EFI	<a href="#">DSP1023 1.0</a> Clause 10.1
	CurrentBIOSVersionString	CIM_SoftwareIdentity.VersionString representing the BIOS/EFI	<a href="#">DSP1023 1.0</a> Clause 10.1
	CurrentManagementFirmwareMajorVersion	CIM_SoftwareIdentity.MajorVersion representing the management firmware	<a href="#">DSP1023 1.0</a> Clause 10.1
	CurrentManagementFirmwareMinorVersion	CIM_SoftwareIdentity.MinorVersion representing the management firmware	<a href="#">DSP1023 1.0</a> Clause 10.1
	CurrentManagementFirmwareRevisionNumber	CIM_SoftwareIdentity.RevisionNumber representing the management firmware	<a href="#">DSP1023 1.0</a> Clause 10.1
	CurrentManagementFirmwareBuildNumber	CIM_SoftwareIdentity.BuildNumber representing the management firmware	<a href="#">DSP1023 1.0</a> Clause 10.1
	CurrentManagementFirmwareElementName	CIM_SoftwareIdentity.ElementName representing the management firmware	<a href="#">DSP1023 1.0</a> Clause 10.1
	CurrentManagementFirmwareVersionString	CIM_SoftwareIdentity.VersionString representing the management firmware	<a href="#">DSP1023 1.0</a> Clause 10.1
DMTF:OSView	OSType	CIM_OperatingSystem.OSType	<a href="#">DSP1029 1.0</a> Clause 10.1
	OSEnabledState	CIM_OperatingSystem.EnabledState	<a href="#">DSP1029 1.0</a> Clause 10.1
	OSVersion	CIM_OperatingSystem.Version	<a href="#">DSP1029 1.1</a> Clause 10.1
DMTF:ComputerSystemView	OtherIdentifyingInfo	CIM_ComputerSystem.OtherIdentifyingInfo	<a href="#">DSP1052 1.0</a> Clause 10.1
	IdentifyingDescriptions	CIM_ComputerSystem.IdentifyingDescriptions	<a href="#">DSP1052 1.0</a> Clause 10.1
	Dedicated	CIM_ComputerSystem.Dedicated	<a href="#">DSP1108 1.0</a> Clause 7.1.2.10.1
	OtherDedicatedDescriptions	CIM_ComputerSystem.OtherDedicatedDescriptions	<a href="#">DSP1108 1.0</a> Clause 7.1.2.10.2
	EnabledState	CIM_ComputerSystem.EnabledState	<a href="#">DSP1052 1.0</a> Clause 10.1
	RequestedState	CIM_ComputerSystem.RequestedState	<a href="#">DSP1052 1.0</a> Clause 10.1
	OperationalStatus	CIM_ComputerSystem.OperationalStatus	<a href="#">DSP1052 1.0</a> Clause 10.1

Implemented Feature Value	CIM_Physical ComputerSystem View Property	Origin Class/Property or Model Correspondence	Referenced Profile
	HealthState	CIM_ComputerSystem.HealthState	<a href="#">DSP1052 1.0</a> Clause 10.1
DMTF:Power Utilization ManagementView	PowerUtilizationModes Supported	CIM_PowerUtilizationManagement Capabilities.PowerUtilizationModes Supported	<a href="#">DSP1085 1.0</a> Clause 10.2
	PowerUtilizationMode	CIM_PowerUtilizationManagement Service.PowerUtilizationMode	<a href="#">DSP1085 1.0</a> Clause 10.1
	PowerAllocationLimit	CIM_PowerAllocationSettingData.Limit	<a href="#">DSP1085 1.0</a> Clause 10.6
DMTF:Numeric SensorView	NumericSensor ElementName	CIM_NumericSensor.ElementName	<a href="#">DSP1009 1.0</a> Clause 10.2
	NumericSensor EnabledState	CIM_NumericSensor.EnabledState	<a href="#">DSP1009 1.0</a> Clause 10.2
	NumericSensor Context	CIM_NumericSensor.Sensor Context	<a href="#">DSP1009 1.1</a> Clause 10.2
	NumericSensorHealth State	CIM_NumericSensor.HealthState	<a href="#">DSP1009 1.0</a> Clause 10.2
	NumericSensor CurrentState	CIM_NumericSensor.CurrentState	<a href="#">DSP1009 1.0</a> Clause 10.2
	NumericSensor PrimaryStatus	CIM_NumericSensor.PrimaryStatus	<a href="#">DSP1009 1.0</a> Clause 10.2
	NumericSensorBase Units	CIM_NumericSensor.BaseUnits	<a href="#">DSP1009 1.0</a> Clause 10.2
	NumericSensorUnit Modifier	CIM_NumericSensor.UnitModifier	<a href="#">DSP1009 1.0</a> Clause 10.2
	NumericSensorRate Units	CIM_NumericSensor.RateUnits	<a href="#">DSP1009 1.0</a> Clause 10.2
	NumericSensor CurrentReading	CIM_NumericSensor.Current Reading	<a href="#">DSP1009 1.0</a> Clause 10.2
	NumericSensorSensor Type	CIM_NumericSensor.SensorType	<a href="#">DSP1009 1.0</a> Clause 10.2
	NumericSensorOther SensorType Description	CIM_NumericSensor.OtherSensor TypeDescription	<a href="#">DSP1009 1.0</a> Clause 10.2
	NumericSensorUpper ThresholdNonCritical	CIM_NumericSensor.Upper ThresholdNonCritical	<a href="#">DSP1009 1.0</a> Clause 10.2
	NumericSensorUpper ThresholdFatal	CIM_NumericSensor.Upper ThresholdFatal	<a href="#">DSP1009 1.0</a> Clause 10.2
NumericSensorUpper ThresholdCritical	CIM_NumericSensor.Upper ThresholdCritical	<a href="#">DSP1009 1.0</a> Clause 10.2	
NumericSensorLower ThresholdNonCritical	CIM_NumericSensor.Lower ThresholdNonCritical	<a href="#">DSP1009 1.0</a> Clause 10.2	

Implemented Feature Value	CIM_PhysicalComputerSystemView Property	Origin Class/Property or Model Correspondence	Referenced Profile
	NumericSensorLowerThresholdFatal	CIM_NumericSensor.LowerThresholdFatal	<a href="#">DSP1009 1.0</a> Clause 10.2
	NumericSensorLowerThresholdCritical	CIM_NumericSensor.LowerThresholdCritical	<a href="#">DSP1009 1.0</a> Clause 10.2
DMTF:Record LogView	LogInstanceID	CIM_RecordLog.InstanceID	<a href="#">DSP1010 1.0</a> Clause 10.5
	LogMaxNumberOfRecords	CIM_RecordLog.MaxNumberOfRecords	<a href="#">DSP1010 1.0</a> Clause 10.5
	LogCurrentNumberOfRecords	CIM_RecordLog.CurrentNumberOfRecords	<a href="#">DSP1010 1.0</a> Clause 10.5
	LogOverWritePolicy	CIM_RecordLog.OverwritePolicy	<a href="#">DSP1010 1.0</a> Clause 10.5
	LogState	CIM_RecordLog.LogState	<a href="#">DSP1010 1.0</a> Clause 10.5
DMTF:Boot ControlView	StructuredBootString	CIM_BootSourceSetting.StructuredBootString	<a href="#">DSP1012 1.0</a> Clause 10.6
	OneTimeBootSource	n/a	<a href="#">DSP1108 1.0</a> Clause 7.1.2.9.2

## 293 7.1.2 Additional requirements

294 This subclause details additional requirements for some properties of  
295 CIM\_PhysicalComputerSystemView.

### 296 7.1.2.1 CIM\_PhysicalComputerSystemView.InstanceID

297 The InstanceID is the property that shall be used to opaquely and uniquely identify an instance of this  
298 class within the scope of the instantiating Namespace. This property shall not correspond to the  
299 InstanceID property of CIM\_ComputerSystem.

### 300 7.1.2.2 Representing system processor information

301 When implemented according to Table 2, the intent of this set of properties is to model the central  
302 processing unit.

303 The NumberOfProcessors property represents the number of homogenous processors on this physical  
304 computer system. Other types of processors (including GPUs) shall not be represented in the  
305 NumberOfProcessors property of CIM\_PhysicalComputerSystemView.

306 The NumberOfProcessorThreads property shall correspond to the NumberOfHardwareThreads property  
307 in the CIM\_ProcessorCapabilities class defined in [DSP1022](#).

308 The properties of the central processing unit of the physical computer system shall be represented as  
309 properties defined in Table 3 from [DSP1022](#). When one or more instances of CIM\_Processor are  
310 instantiated and represented in the view class, each CIM\_Processor instance should be associated with  
311 the CIM\_PhysicalComputerSystemView instance through an instance of CIM\_ElementView where the



312 Antecedent property is a reference to the corresponding CIM\_Processor instance and the Dependent  
 313 property is a reference to the CIM\_PhysicalComputerSystemView instance.

314 **Table 3 – Property origins for processor**

CIM_PhysicalComputerSystemView property name	CIM_Processor property name (origin)
ProcessorFamily	Family
ProcessorMaxClockSpeed	MaxClockSpeed

315 **7.1.2.3 Representing system memory information**

316 When implemented according to Table 2, the properties of the memory of the physical computer system  
 317 shall be represented as properties defined in Table 4 from [DSP1026](#). When the instance of CIM\_Memory  
 318 representing total system memory is instantiated and represented in the view class, the CIM\_Memory  
 319 instance should be associated with the CIM\_PhysicalComputerSystemView instance through an instance of  
 320 CIM\_ElementView where the Antecedent property is a reference to the CIM\_Memory instance and the  
 321 Dependent property is a reference to the CIM\_PhysicalComputerSystemView instance.

322 **Table 4 – Property origins for memory**

CIM_PhysicalComputerSystemView property name	CIM_Memory property name (origin)
MemoryBlockSize	BlockSize
MemoryNumberOfBlocks	NumberOfBlocks
MemoryConsumableBlocks	ConsumableBlocks

323 **7.1.2.4 Representing system software inventory**

324 When implemented according to Table 2, the properties of the current BIOS or EFI firmware of the  
 325 physical computer system shall be represented as properties defined in Table 5. The current BIOS or EFI  
 326 firmware property values shall correspond to an instance of CIM\_SoftwareIdentity where the  
 327 Classifications property contains a value of 10 (Firmware) or 11 (BIOS/FCODE). If instantiated, this  
 328 corresponding instance of CIM\_SoftwareIdentity shall be associated with the underlying instance of  
 329 CIM\_ComputerSystem by an instance of CIM\_ElementSoftwareIdentity where the  
 330 ElementSoftwareStatus property has a value of 2 (Current).

331 When an instance of CIM\_SoftwareIdentity representing the current BIOS or EFI firmware is instantiated  
 332 and represented in the view class, the CIM\_SoftwareIdentity instance should be associated with the  
 333 CIM\_PhysicalComputerSystemView instance through an instance of CIM\_ElementView where the  
 334 Antecedent is a reference to the CIM\_SoftwareIdentity instance and the Dependent property is a  
 335 reference to the CIM\_PhysicalComputerSystemView instance.

336

Table 5 – Property origins for current BIOS or EFI firmware

CIM_PhysicalComputerSystemView property name	CIM_SoftwareIdentity property name (origin)
CurrentBIOSMajorVersion	MajorVersion
CurrentBIOSMinorVersion	MinorVersion
CurrentBIOSRevisionNumber	RevisionNumber
CurrentBIOSBuildNumber	BuildNumber
CurrentBIOSVersionString	VersionString

337 The properties of the current management firmware of the physical computer system shall be represented  
 338 as properties defined in Table 6 from [DSP1023](#). The current management firmware property values shall  
 339 correspond to the instance of CIM\_SoftwareIdentity referenced by the instance of  
 340 CIM\_ElementSoftwareIdentity where the ElementSoftwareStatus property has a value of 2 (Current).

341 When an instance of CIM\_SoftwareIdentity representing the current management firmware is instantiated  
 342 and represented in the view class, the CIM\_SoftwareIdentity instance should be associated with the  
 343 CIM\_PhysicalComputerSystemView instance through an instance of CIM\_ElementView where the  
 344 Antecedent is a reference to the CIM\_SoftwareIdentity instance and the Dependent property is a  
 345 reference to the CIM\_PhysicalComputerSystemView instance.

346

Table 6 – Property origins for current management firmware

CIM_PhysicalComputerSystemView property name	CIM_SoftwareIdentity property name (origin)
CurrentManagementFirmwareMajorVersion	MajorVersion
CurrentManagementFirmwareMinorVersion	MinorVersion
CurrentManagementFirmwareRevisionNumber	RevisionNumber
CurrentManagementFirmwareBuildNumber	BuildNumber
CurrentManagementFirmwareVersionString	VersionString
CurrentManagementFirmwareElementName	ElementName

#### 347 7.1.2.5 Representing operating system information

348 This subclause describes the requirements for representing the running operating system for the  
 349 CIM\_PhysicalComputerSystemView class.

350 When implemented according to Table 2, the properties of the operating system of the physical computer  
 351 system shall be represented as properties as defined in Table 7 from [DSP1029](#). When an instance of  
 352 CIM\_OperatingSystem is instantiated and represented in the view class, the CIM\_OperatingSystem  
 353 instance should be associated with the CIM\_PhysicalComputerSystemView instance through an instance  
 354 of CIM\_ElementView where the Antecedent property is a reference to the CIM\_OperatingSystem  
 355 instance and the Dependent property is a reference to the CIM\_PhysicalComputerSystemView instance.

356

**Table 7 – Property origins for the operating system**

CIM_PhysicalComputerSystemView property name	CIM_OperatingSystem property name (origin)
OSEnabledState	EnabledState
OSVersion	Version
OSType	OSType

357 **7.1.2.6 Representing power utilization information**

358 When implemented according to Table 2, the PowerAllocationLimit property shall correspond to the Limit  
 359 property of the CIM\_PowerAllocationSettingData class defined in [DSP1085](#) with the added  
 360 “PowerAllocation” prefix to logically group properties related to power utilization and avoid naming  
 361 collision.

362 The power allocation limit of the physical computer system shall be represented as the property as  
 363 defined in Table 8 from [DSP1085](#). When an instance of CIM\_PowerAllocationSettingData is instantiated  
 364 and represented in the view class, the CIM\_PowerUtilizationManagementService instance should be  
 365 associated with the CIM\_PhysicalComputerSystemView instance through an instance of  
 366 CIM\_ElementView where the Antecedent property is a reference to the  
 367 CIM\_PowerUtilizationManagementService instance and the Dependent property is a reference to the  
 368 CIM\_PhysicalComputerSystemView instance.

369

**Table 8 – Property origins for the power allocation limit**

CIM_PhysicalComputerSystemView property name	CIM_PowerAllocationSettingData property name (origin)
PowerAllocationLimit	Limit

370 **7.1.2.7 Representing system numeric sensors**

371 When implemented according to Table 2, the properties of a numeric sensor of the physical computer  
 372 system shall be represented as elements of a group of indexed array properties as defined in Table 9  
 373 from [DSP1009](#). For these properties, the array elements with same index shall present a view of the  
 374 same numeric sensor. When an instance of CIM\_NumericSensor is instantiated and represented in the  
 375 view class, the CIM\_NumericSensor instance should be associated with the  
 376 CIM\_PhysicalComputerSystemView instance through an instance of CIM\_ElementView where the  
 377 Antecedent property is a reference to the CIM\_NumericSensor instance and the Dependent property is a  
 378 reference to the CIM\_PhysicalComputerSystemView instance.

379

Table 9 – Property origins for numeric sensors

CIM_PhysicalComputerSystemView property name	CIM_NumericSensor property name (origin)
NumericSensorElementName[]	ElementName
NumericSensorEnabledState[]	EnabledState
NumericSensorHealthState[]	HealthState
NumericSensorCurrentState[]	CurrentState
NumericSensorPrimaryStatus[]	PrimaryStatus
NumericSensorBaseUnits[]	BaseUnits
NumericSensorUnitModifier[]	UnitModifier
NumericSensorRateUnits[]	RateUnits
NumericSensorCurrentReading[]	CurrentReading
NumericSensorSensorType[]	SensorType
NumericSensorOtherSensorTypeDescription[]	OtherSensorTypeDescription
NumericSensorUpperThresholdNonCritical[]	UpperThresholdNonCritical
NumericSensorUpperThresholdFatal[]	UpperThresholdFatal
NumericSensorUpperThresholdCritical[]	UpperThresholdCritical
NumericSensorLowerThresholdNonCritical[]	LowerThresholdNonCritical
NumericSensorLowerThresholdFatal[]	LowerThresholdFatal
NumericSensorLowerThresholdCritical[]	LowerThresholdCritical
NumericSensorContext[]	SensorContext

### 380 7.1.2.8 Representing system record logs

381 When implemented according to Table 2, the properties of a record log of the physical computer system  
 382 shall be represented as elements of a group of indexed array properties as defined in Table 10 from  
 383 [DSP1010](#). For these properties, the array elements with same index shall present a view of the same  
 384 record log.

385 Instances of CIM\_RecordLog that contain information about the underlying computer system should be  
 386 represented in the view class.

387 When instantiated, the CIM\_RecordLog instance shall be associated with the  
 388 CIM\_PhysicalComputerSystemView instance through an instance of CIM\_ElementView where the  
 389 Antecedent property is a reference to the CIM\_RecordLog instance and the Dependent property is a  
 390 reference to the CIM\_PhysicalComputerSystemView instance.

391

**Table 10 – Property origins for record logs**

CIM_PhysicalComputerSystemView property name	CIM_RecordLog property name (origin)
LogInstanceID[]	InstanceID
LogMaxNumberOfRecords[]	MaxNumberOfRecords
LogCurrentNumberOfRecords[]	CurrentNumberOfRecords
LogOverWritePolicy[]	OverwritePolicy
LogState[]	LogState

392 **7.1.2.9 Representing system boot configuration**

393 When implemented according to Table 2, the enabled boot sources of the boot configuration of the  
 394 physical computer system shall be represented as elements of an ordered array property as defined in  
 395 Table 11 from [DSP1012](#).

396 When an instance of CIM\_BootSourceSetting is instantiated and represented in the view class, the  
 397 CIM\_BootService instance should be associated with the CIM\_PhysicalComputerSystemView instance  
 398 through an instance of CIM\_ElementView where the Antecedent property is a reference to the  
 399 CIM\_BootService instance and the Dependent property is a reference to the  
 400 CIM\_PhysicalComputerSystemView instance.

401

**Table 11 – Property origin for boot sources**

CIM_PhysicalComputerSystemView property name	CIM_BootSourceSetting property name (origin)
StructuredBootString[]	StructuredBootString
OneTimeBootSource	n/a

402 **7.1.2.9.1 CIM\_PhysicalComputerSystemView.StructuredBootString**

403 This property represents the boot sources that are available to be used for the next one-time boot of the  
 404 physical computer system.

405 **7.1.2.9.2 CIM\_PhysicalComputerSystemView.OneTimeBootSource**

406 This property represents the boot source that is used for the next one-time boot of the physical computer  
 407 system. The value of this property is an index into the StructuredBootString property. A value of NULL  
 408 shall represent that the one-time boot source is not configured.

409 **7.1.2.10 Representing system identity information**

410 **7.1.2.10.1 CIM\_PhysicalComputerSystemView.Dedicated**

411 When implemented according to Table 2, the Dedicated property shall indicate the purposes to which the  
 412 physical computer system is dedicated, if any, and what functionality is provided. See the specialized  
 413 profiles of [DSP1052](#) for requirements (for example, DSP1004 and DSP1058).

414 **7.1.2.10.2 CIM\_PhysicalComputerSystemView.OtherDedicatedDescriptions**

415 When implemented according to Table 2, the OtherDedicatedDescriptions property shall contain strings  
 416 describing how or why the physical computer system is dedicated when the Dedicated property includes  
 417 “Other” (value=2). See the specialized profiles of [DSP1052](#) for requirements (for example, DSP1004 and  
 418 DSP1058).

419 **7.1.2.11 Representing system FRU information**420 **7.1.2.11.1 CIM\_PhysicalComputerSystemView.FRUInfoSupported**

421 When implemented according to Table 2, the FRUInfoSupported property shall correspond to the value of  
 422 the CIM\_PhysicalAssetCapabilities.FRUInfoSupported property value associated to the CIM\_Chassis  
 423 instance.

424 **8 Methods**

425 This clause details the requirements for supporting intrinsic operations and extrinsic methods for the CIM  
 426 elements defined by this profile.

427 **8.1 CIM\_PhysicalComputerSystemView.RequestStateChange( )**

428 When the ImplementedFeatures property of the CIM\_RegisteredProfile instance contains  
 429 “DMTF:ComputerSystemView”, RequestStateChange( ) shall be implemented and the implemented  
 430 method shall not return a value of 1 (Method is unsupported).

431 Invocation of the RequestStateChange( ) method changes the physical computer system’s state to the  
 432 value specified in the RequestedState parameter. A return code value of zero shall indicate that the  
 433 requested state change was initiated successfully.

434 Detailed requirements of the RequestStateChange( ) method are specified in Table 12 and Table 13.

435 No standard messages are defined.

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

438 See CIM\_ComputerSystem.RequestStateChange( ) in [DSP1052](#) for additional details.

439 **Table 12 – CIM\_PhysicalComputerSystemView.RequestStateChange( ) method: Return code**  
 440 **values**

Value	Description
0	Request was successfully executed.
2	Error occurred.
4096	Job started: REF returned to started CIM_ConcreteJob.

441

442 **Table 13 – CIM\_PhysicalComputerSystemView.RequestStateChange() method: Parameters**

Qualifiers	Name	Type	Description/Values
IN, REQ	RequestedState	uint16	Valid state values (mapped to CIM_ComputerSystem.RequestStateChange( )) 2 (Enabled) – On 3 (Disabled) – Off-Soft 9 (Quiesce) – Sleep 11 (Reset) – Power-Cycle (Off-Soft)
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
OUT	Job	CIM_ConcreteJob REF	Returned if job started

443 **8.2 CIM\_PhysicalComputerSystemView.ClearLog( )**

444 When the ImplementedFeatures property of the CIM\_RegisteredProfile instance contains  
445 “DMTF:RecordLogView”, ClearLog( ) may be implemented.

446 Invocation of the ClearLog( ) method deletes all the entries of the specified record log of the physical  
447 computer system identified by the LogInstanceID parameter. A return code value of zero shall indicate  
448 that the clearing of the log entries was initiated successfully.

449 Detailed requirements of the ClearLog( ) method are specified in Table 14 and Table 15.

450 No standard messages are defined.

451 **Table 14 – CIM\_PhysicalComputerSystemView.ClearLog() method: Return code values**

Value	Description
0	Request was successfully executed.
1	Method is unsupported.
2	Error occurred.

452 **Table 15 – CIM\_PhysicalComputerSystemView.ClearLog() method: Parameters**

Qualifiers	Name	Type	Description/Values
IN, REQ	LogInstanceID	String	Identifier of record log that is requested to be cleared

453 **8.3 CIM\_PhysicalComputerSystemView.InstallSoftwareFromURI( )**

454 When the ImplementedFeatures property of the CIM\_RegisteredProfile instance contains  
455 “DMTF:SoftwareInventoryView”, InstallSoftwareFromURI( ) may be implemented.

456 Invocation of the InstallSoftwareFromURI( ) method starts a job to install software from the designated  
457 URI to the physical computer system. A return code value of zero shall indicate that the installation of  
458 software was initiated successfully. Based on the payloads, implementations shall determine whether the  
459 installation is intended for BIOS or Management Firmware.

460 Detailed requirements of the InstallSoftwareFromURI() method are specified in Table 16 and Table 17.

461 No standard messages are defined.

462 See CIM\_SoftwareInstallationService.InstallFromURI() in [DSP1025](#) for additional details.

463 **Table 16 – CIM\_PhysicalComputerSystemView.InstallSoftwareFromURI() method: Return code**  
464 **values**

Value	Description
0	Job completed with no error.
1	Method is unsupported.
2	Error occurred.
4096	Job started: REF returned to started CIM_ConcreteJob.

465 **Table 17 – CIM\_PhysicalComputerSystemView.InstallSoftwareFromURI() method: Parameters**

Qualifiers	Name	Type	Description/Values
IN, REQ	URI	string	A URI for the software.
IN	InstallOptions[]	uint16	Options to control the install process. See CIM_SoftwareInstallationService.InstallFromURI() in <a href="#">DSP1025</a> for additional details.
IN	Classifications[]	uint16	Identify the classification of software to install. See CIM_SoftwareIdentity.Classifications in <a href="#">DSP1023</a> for additional details.
IN	InstallOptionsValues[]	string	InstallOptionsValues is an array of strings providing additional information to InstallOptions for the method to install the software. See CIM_SoftwareInstallationService.InstallFromURI() in <a href="#">DSP1025</a> for additional details.
OUT	Job	CIM_ConcreteJob REF	Returned if job started

#### 466 **8.4 CIM\_PhysicalComputerSystemView.SetOneTimeBootSource( )**

467 When the ImplementedFeatures property of the CIM\_RegisteredProfile instance contains  
468 “DMTF:BootControlView”, SetOneTimeBootSource( ) may be implemented.

469 Invocation of the SetOneTimeBootSource() method sets the boot source for the next boot only. A return  
470 code value of zero shall indicate that the new one time boot source was set successfully.

471 If the StructuredBootString parameter contains a value not contained in the StructuredBootString property  
472 of the CIM\_PhysicalComputerSystemView instance, then the method shall return 2 (Error Occurred).

473 Detailed requirements of the SetOneTimeBootSource() method are specified in Table 18 and Table 19.

474 No standard messages are defined.



475 **Table 18 – CIM\_PhysicalComputerSystemView.SetOneTimeBootSource() method: Return code**  
 476 **values**

Value	Description
0	Completed with no error.
1	Method is unsupported.
2	Error occurred.
4096	Job started: REF returned to started CIM_ConcreteJob.

477 **Table 19 – CIM\_PhysicalComputerSystemView.SetOneTimeBootSource() method: Parameters**

Qualifiers	Name	Type	Description/Values
IN, REQ	StructuredBootString	string	A StructuredBootString value
OUT	Job	CIM_ConcreteJob REF	Returned if job started

478 **8.5 Profile conventions for operations**

479 For each profile class (including associations), the implementation requirements for operations, including  
 480 those in the following default list, are specified in class-specific subclauses of this clause.

481 The default list of operations is as follows:

- 482 • GetInstance
- 483 • EnumerateInstances
- 484 • EnumerateInstanceNames
- 485 • Associators
- 486 • AssociatorNames
- 487 • References
- 488 • ReferenceNames

489 **8.6 CIM\_PhysicalComputerSystemView**

490 Table 20 lists operations that either have special requirements beyond those from [DSP0200](#) or shall not  
 491 be supported.

492 **Table 20 – Operations: CIM\_PhysicalComputerSystemView**

Operation	Requirement	Description
InvokeMethod	Conditional	If “DMTF:ComputerSystemView”, “DMTF:RecordLogView”, “DMTF:BootControlView”, or “DMTF:SoftwareUpdateView” is an implemented feature, this operation shall be supported. See 8.

493 **8.7 CIM\_ElementView**

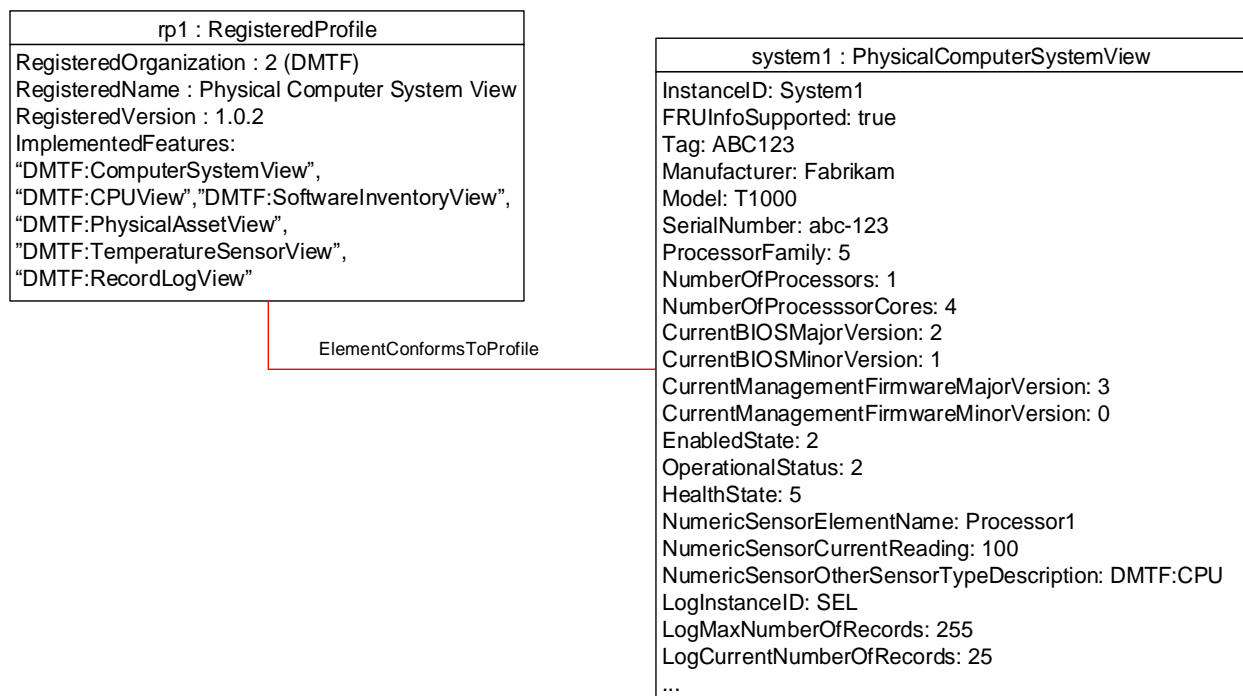
494 All operations in the default list in 8.5 shall be implemented as defined in [DSP0200](#).

## 495 9 Use cases

496 This clause contains object diagrams and use cases for the *Physical Computer System View Profile*.

### 497 9.1 Miscellaneous object diagrams

498 The object diagram in Figure 2 shows one possible method for advertising profile conformance. The  
 499 instances of CIM\_RegisteredProfile are used to identify the version of the *Physical Computer System  
 500 View Profile* with which an instance of CIM\_PhysicalComputerSystemView and its associated instances  
 501 are conformant. An instance of CIM\_RegisteredProfile exists for each profile that is instrumented in the  
 502 system. An instance of CIM\_RegisteredProfile identifies the “DMTF Physical Computer System View  
 503 Profile version 1.0.0”. This diagram represents a simple managed computer system that only implements  
 504 the Profile Registration Profile and the Physical Computer System View Profile. The implementation of  
 505 Physical Computer System View Profile indicates that properties and methods related to Computer  
 506 System, CPU, Software Inventory, Physical Asset, Sensors, and Record Log are implemented as  
 507 advertised by using the ImplementedFeatures property of the instance of CIM\_RegisteredProfile.

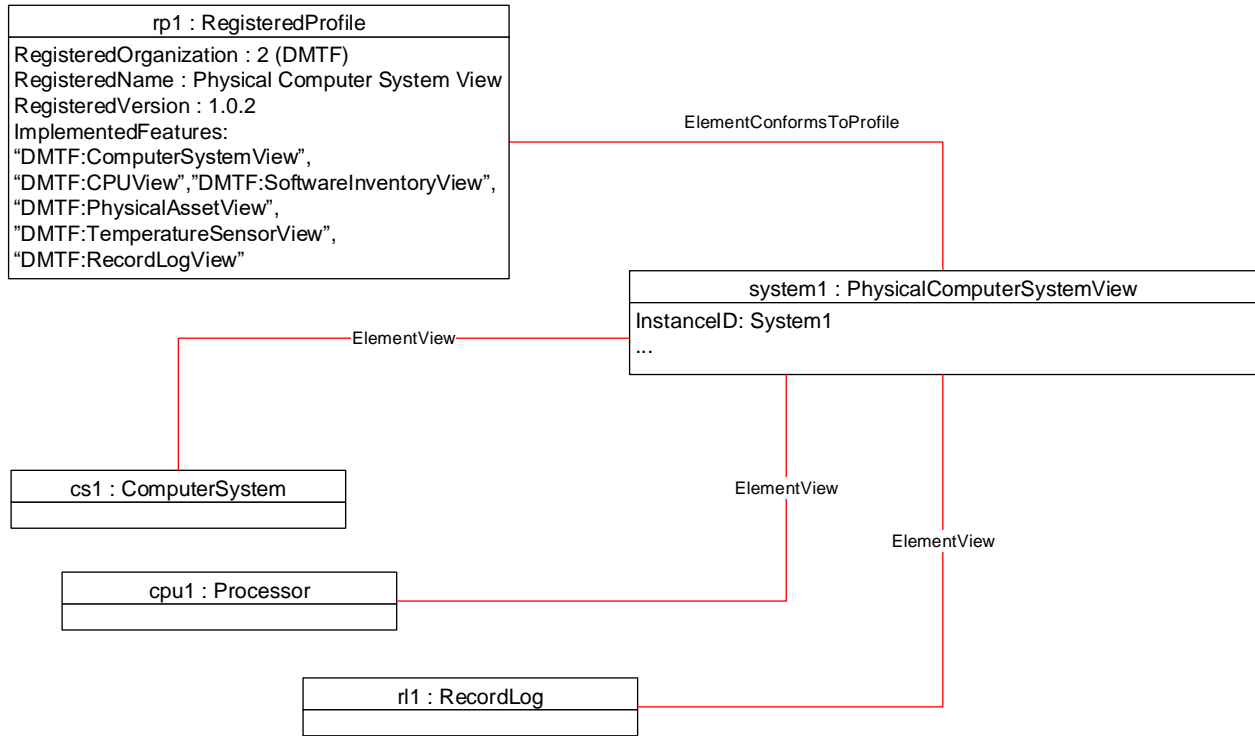


508

509 **Figure 2 – Simple PhysicalComputerSystemView implementation**

### 510 9.2 Small footprint managed computer system with additional CIM Schema 511 implemented

512 The object diagram in Figure 3 shows a small footprint managed computer system that implements the  
 513 same features as the managed computer system in Figure 2, but also implements CIM Schema that is  
 514 associated to specific properties. This managed computer system does not claim conformance to  
 515 additional profiles, but exposes additional properties/methods through the CIM classes associated to the  
 516 instance of CIM\_PhysicalComputerSystemView.



517

518

Figure 3 – Small footprint PhysicalComputerSystemView implementation

519

### 9.3 Large managed computer system with additional CIM Profiles implemented

520

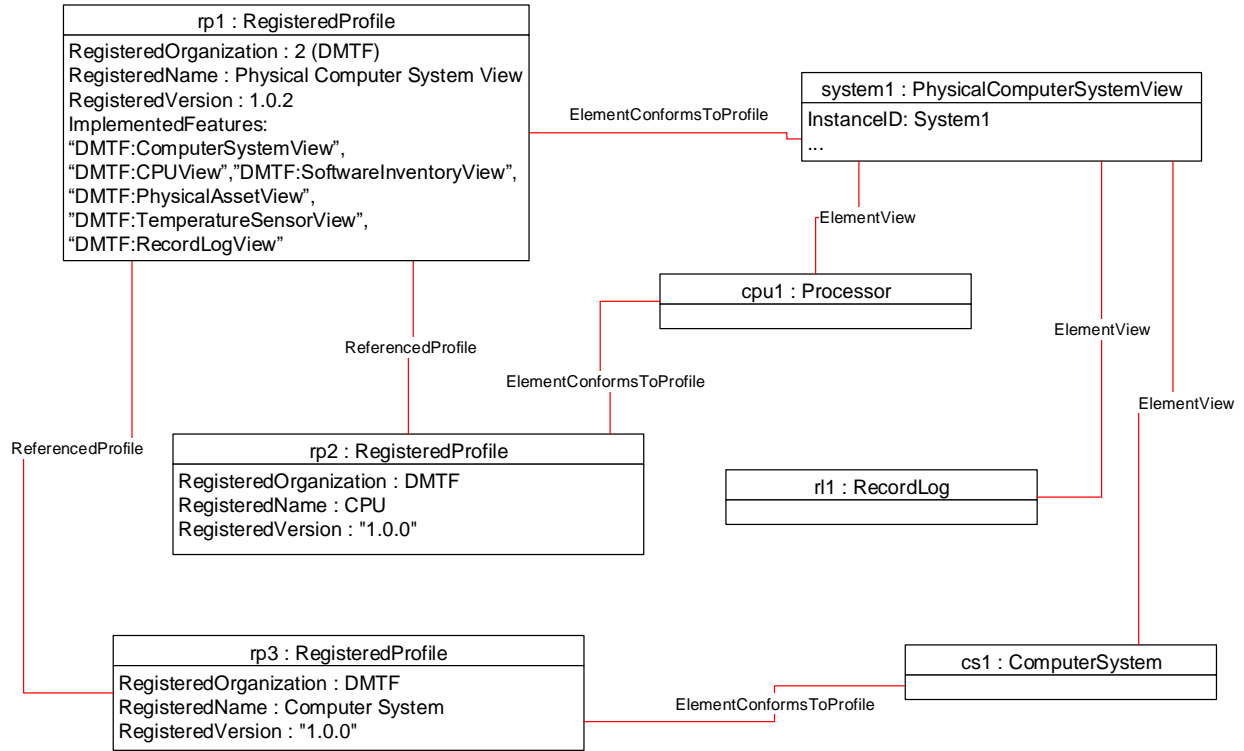
The object diagram in Figure 4 shows a large managed computer system that implements the same

521

features as the managed computer system in Figure 3, but also implements CIM Schema that is

522

conformant to CIM Profiles providing richer management capabilities.



523

524

Figure 4 – Rich PhysicalComputerSystemView implementation

525

### 9.4 Managed system exposing multiple temperature sensors

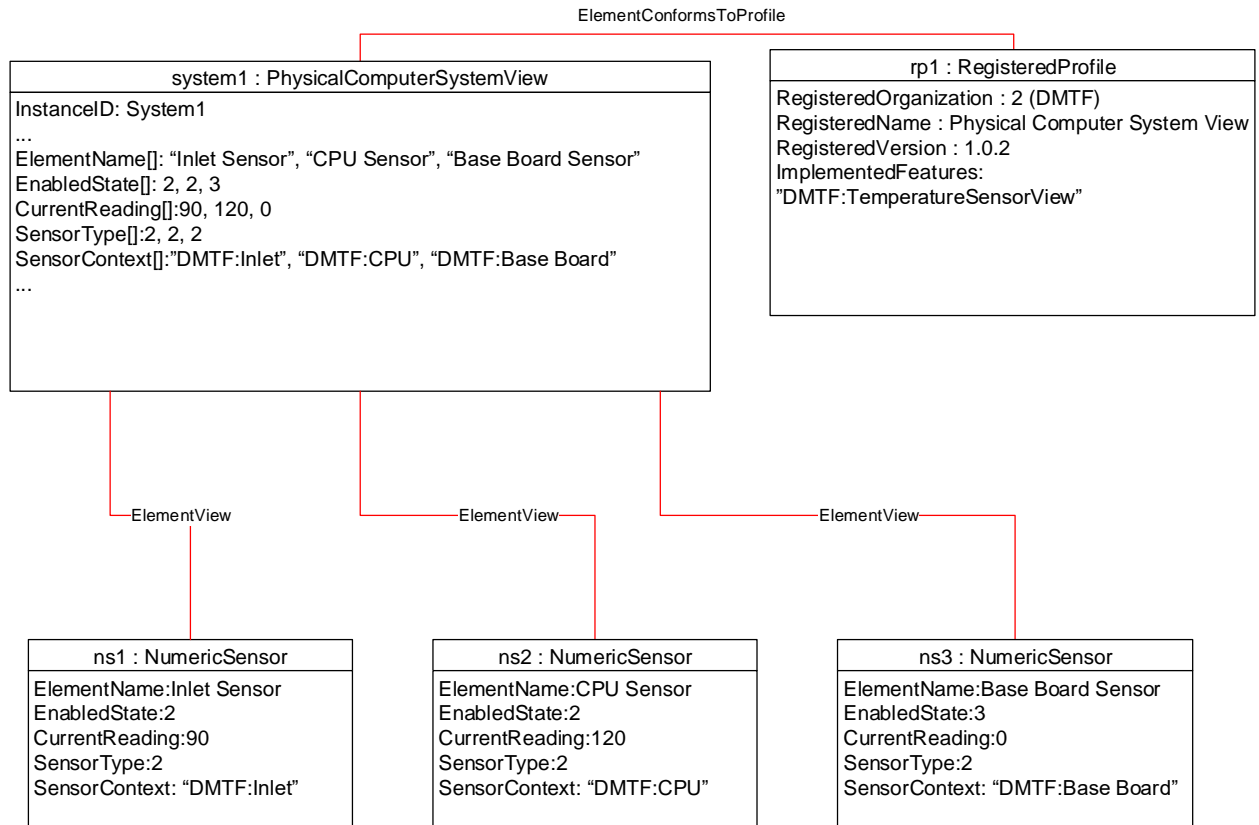
526

527

528

529

The object diagram in Figure 5 shows a managed computer system that implements multiple temperature sensors for Inlet, CPU, and Base Board. Although the instances of CIM\_NumericSensor and corresponding association are not required, they illustrate how to represent multiple numeric sensors as an indexed array in the CIM\_PhysicalComputerSystemView instance.



530

531

**Figure 5 – PhysicalComputerSystemView implementation of multiple temperature sensors**

**532 9.5 Determine the enabled state of the physical computer system**

533 To determine the enabled state:

- 534 1) Verify that the managed computer system implements the DMTF:ComputerSystemView feature  
535 by reading the ImplementedFeatures property of the CIM\_RegisteredProfile instance.
- 536 2) Read the EnabledState property of the instance of CIM\_PhysicalComputerSystemView.

**537 9.6 Change the enabled state of the physical computer system**

538 To change the enabled state:

- 539 1) Verify that the managed computer system implements the DMTF:ComputerSystemView feature  
540 by reading the ImplementedFeatures property of the CIM\_RegisteredProfile instance.
- 541 2) Invoke the RequestStateChange( ) method of the instance of CIM\_PhysicalComputerSystemView  
542 providing the new requested state.
- 543 3) See Table 13 for the supported list of requested enabled states.

**544 9.7 Get properties of a specific record log of the physical computer system**

545 To get the properties of a specific record log:

- 546 1) Verify that the managed computer system implements the DMTF:RecordLogView feature by  
547 reading the ImplementedFeatures property of the CIM\_RegisteredProfile instance.
- 548 2) Identify the index to the specific log that matches the LogInstanceID.
- 549 3) Use this index to read the ordered arrays MaxNumberOfRecords, CurrentNumberOfRecords,  
550 OverWritePolicy, and LogState for that specific record log.

**551 9.8 Browse the records of a log of the physical computer system**

552 To browse log entries:

- 553 1) Verify that the managed computer system implements the DMTF:RecordLogView feature by  
554 reading the ImplementedFeatures property of the CIM\_RegisteredProfile instance.
- 555 2) Verify that an instance of CIM\_RegisteredProfile for Record Log Profile exists.
- 556 3) Traverse the association to the CIM\_RecordLog instance with the LogInstanceID by using the  
557 CIM\_ElementView association class.
- 558 4) Enumerate instances of CIM\_LogEntry that are associated through the CIM\_LogManagesRecord  
559 association to the given instance of CIM\_RecordLog.  
560

**561 9.9 Monitor temperature sensor readings of the physical computer system**

562 To monitor CPU temperature sensor readings:

- 563 1) Verify that the managed computer system implements the DMTF:NumericSensorView feature by  
564 reading the ImplementedFeatures property of the CIM\_RegisteredProfile instance.
- 565 2) Identify the index to a specific sensor that contains "DMTF:CPU" as the value for  
566 NumericSensorContext.
- 567 3) Use this index to read the ordered arrays NumericSensorBaseUnits, NumericSensorUnitModifier,  
568 NumericSensorRateUnits, and NumericSensorCurrentReading to compute the reading value for  
569 that specific sensor.

**570 9.10 Configure a source of the physical computer system for next reboot only**

571 To configure the boot source for next reboot only:

- 572 1) Verify that the managed computer system implements the DMTF:BootControlView feature by  
573 reading the ImplementedFeatures property of the CIM\_RegisteredProfile instance.
- 574 2) Invoke the SetOneTimeBootSource( ) method of the instance of  
575 CIM\_PhysicalComputerSystemView to reflect the desired boot source based on an index value  
576 from the StructedBootString array.

**577 9.11 Update the BIOS firmware of the physical computer system**

578 To update the BIOS firmware:

- 579 1) Verify that the managed computer system implements the DMTF:SoftwareUpdateView feature by  
580 reading the ImplementedFeatures property of the CIM\_RegisteredProfile instance.
- 581 2) Invoke the InstallSoftwareFromURI( ) method of the instance of  
582 CIM\_PhysicalComputerSystemView providing the classification value of 6 (Firmware/BIOS),  
583 location of the software, and required options.  
584

585 **10 CIM Elements**

586 Table 21 shows the instances of CIM Elements for this profile. Instances of the CIM Elements shall be  
 587 implemented as described in Table 21.

588 **Table 21 – CIM Elements: Physical Computer System View Profile**

Element Name	Requirement	Description
<b>Classes</b>		
CIM_PhysicalComputerSystemView	Mandatory	See 10.1.
CIM_RegisteredProfile	Mandatory	See 10.2.
CIM_ElementView	Optional	See 10.3
CIM_ElementConformsToProfile	Mandatory	See 10.4
<b>Indications</b>		
None defined in this profile		

589 **10.1 CIM\_PhysicalComputerSystemView**

590 CIM\_PhysicalComputerSystemView describes a physical computer system and associated management  
 591 information in a managed computer system.

592 **Table 22 – Class: CIM\_PhysicalComputerSystemView**

Elements	Requirement	Description
InstanceID	Mandatory	Key. See 7.1.2.1.
OtherIdentifyingInfo	Conditional	See Table 2.
IdentifyingDescriptions	Conditional	See Table 2.
EnabledState	Mandatory	See Table 2.
RequestedState	Conditional	See Table 2.
OperationalStatus	Mandatory	See Table 2.
HealthState	Mandatory	See Table 2.
Dedicated	Mandatory	See Table 2.
OtherDedicatedDescriptions	Conditional	See Table 2.
FRUInfoSupported	Conditional	See Table 2.
Tag	Conditional	See Table 2.
Manufacturer	Conditional	See Table 2.
Model	Conditional	See Table 2.
SKU	Conditional	See Table 2.
SerialNumber	Conditional	See Table 2.
Version	Conditional	See Table 2.
PartNumber	Conditional	See Table 2.



Elements	Requirement	Description
PowerUtilizationModesSupported	Conditional	See Table 2.
PowerUtilizationMode	Conditional	See Table 2.
PowerAllocationLimit	Conditional	See Table 2.
NumericSensorElementName	Conditional	See Table 2.
NumericSensorEnabledState	Conditional	See Table 2.
NumercSensorHealthState	Conditional	See Table 2.
NumericSensorCurrentState	Conditional	See Table 2.
NumericSensorPrimaryStatus	Conditional	See Table 2.
NumericSensorBaseUnits	Conditional	See Table 2.
NumericSensorUnitModifier	Conditional	See Table 2.
NumericSensorRateUnits	Conditional	See Table 2.
NumericSensorCurrentReading	Conditional	See Table 2.
NumericSensorSensorType	Conditional	See Table 2.
NumericSensorOtherSensorTypeDescription	Conditional	See Table 2.
NumericSensorContext	Conditional	See Table 2.
NumericSensorUpperThresholdNonCritical	Conditional	See Table 2.
NumericSensorUpperThresholdCritical	Conditional	See Table 2.
NumericSensorUpperThresholdFatal	Conditional	See Table 2.
NumericSensorLowerThresholdNonCritical	Conditional	See Table 2.
NumericSensorLowerThresholdCritical	Conditional	See Table 2.
NumericSensorLowerThresholdFatal	Conditional	See Table 2.
LogInstanceID	Conditional	See Table 2.
LogMaxNumberOfRecords	Conditional	See Table 2.
LogCurrentNumberOfRecords	Conditional	See Table 2.
LogOverWritePolicy	Conditional	See Table 2.
LogState	Conditional	See Table 2.
StructuredBootString	Conditional	See Table 2.
OneTimeBootSource	Conditional	See Table 2.
NumberOfProcessors	Conditional	See Table 2.
NumberOfProcessorCores	Conditional	See Table 2.
NumberOfProcessorThreads	Conditional	See Table 2.
ProcessorFamily	Conditional	See Table 2.
ProcessorCurrentClockSpeed	Conditional	See Table 2.
ProcessorMaxClockSpeed	Conditional	See Table 2.
MemoryBlockSize	Conditional	See Table 2.

Elements	Requirement	Description
MemoryNumberOfBlocks	Conditional	See Table 2.
MemoryConsumableBlocks	Conditional	See Table 2.
CurrentBIOSMajorVersion	Conditional	See Table 2.
CurrentBIOSMinorVersion	Conditional	See Table 2.
CurrentBIOSRevisionNumber	Conditional	See Table 2.
CurrentBIOSBuildNumber	Conditional	See Table 2.
CurrentBIOSVersionString	Conditional	See Table 2.
CurrentManagementFirmwareMajorVersion	Conditional	See Table 2.
CurrentManagementFirmwareMinorVersion	Conditional	See Table 2.
CurrentManagementFirmwareRevisionNumber	Conditional	See Table 2.
CurrentManagementFirmwareBuildNumber	Conditional	See Table 2.
CurrentManagementFirmwareElementName	Conditional	See Table 2.
CurrentManagementFirmwareVersionString	Conditional	See Table 2.
OSType	Conditional	See Table 2.
OSVersion	Conditional	See Table 2.
OSEnabledState	Conditional	See Table 2.
RequestStateChange( )	Mandatory	See 8.1.
ClearLog( )	Conditional	See 8.2.
InstallSoftwareFromURI( )	Conditional	See 8.3.
SetOneTimeBootSource( )	Conditional	See 8.4.

## 593 10.2 CIM\_RegisteredProfile

594 CIM\_RegisteredProfile identifies the *Physical Computer System View Profile* in order for a client to  
 595 determine whether an instance of CIM\_PhysicalComputerSystemView is conformant with this profile. The  
 596 CIM\_RegisteredProfile class is defined by the *Profile Registration Profile* ([DSP1033](#)). With the exception  
 597 of the mandatory values specified for the properties in Table 23, the behavior of the  
 598 CIM\_RegisteredProfile instance is in accordance with [DSP1033](#).

599 **Table 23 – Class: CIM\_RegisteredProfile**

Elements	Requirement	Description
RegisteredName	Mandatory	This property shall have a value of "Physical Computer System View".
RegisteredVersion	Mandatory	This property shall have a value of "1.0.2".
RegisteredOrganization	Mandatory	This property shall have a value of 2 (DMTF).

Elements	Requirement	Description
ImplementedFeatures	Mandatory	This property shall contain "DMTF:ComputerSystemView". This property may contain one or more of these values "DMTF:RecordLogView", "DMTF:NumericSensorView", "DMTF:CPUView", "DMTF:BootControlView", "DMTF:SoftwareInventoryView", "DMTF:PhysicalAssetView", "DMTF:SystemMemoryView", "DMTF:PowerUtilizationManagementView", "DMTF:OSView", "DMTF:SoftwareUpdateView". Presences of values in this property only indicate implemented properties/methods in CIM_PhysicalComputerSystemView and do not indicate conformance to additional CIM Profiles.

600 **10.3 CIM\_ElementView**

601 CIM\_ElementView associates additional CIM Schema to the CIM\_PhysicalComputerSystemView  
602 instance.

603 **Table 24 – Class: CIM\_ElementView**

Elements	Requirement	Description
Antecedent	Mandatory	This property shall be a reference to the CIM_ManagedElement that is an instance in the normalized representation of the managed resource.
Dependent	Mandatory	This property shall be a reference to the CIM_PhysicalComputerSystemView instance.

604 **10.4 CIM\_ElementConformsToProfile**

605 CIM\_ElementConformsToProfile associates the instance of CIM\_RegisteredProfile to the  
606 CIM\_PhysicalComputerSystemView instance.

607 **Table 25 – Class: CIM\_ElementConformsToProfile**

Elements	Requirement	Description
ConformantStandard	Mandatory	This property shall be a reference to the CIM_RegisteredProfile instance for the Physical Computer System View Profile.
ManagedElement	Mandatory	This property shall be a reference to the CIM_PhysicalComputerSystemView instance.

608  
609  
610  
611

## ANNEX A (informative)

### Change log

Version	Date	Description
1.0.0	2013-08-22	
1.0.1	2014-05-22	<p>This errata addresses these issues:</p> <ul style="list-style-type: none"> <li>• Added clarifying language that RequestStateChange() shall not return 1 (Method is unsupported) as it is mandatory to be implemented. In addition, removed return value 1 (Method is unsupported) from Table 12</li> <li>• Clarified language that all other methods may be implemented and not conditional as requirements are based on referenced profiles</li> <li>• Corrected language of use cases referring to power state to enabled state</li> <li>• Fixed incorrect table reference in 9.6</li> </ul>
1.0.2	2017-03-18	<p>This errata addresses these issues:</p> <ul style="list-style-type: none"> <li>• Updated RegisteredVersion to reflect errata version number in Section 10.2, and</li> <li>• Updated RegisteredOrganization description to reflect correct value for DMTF in Section 10.2.</li> <li>• Updated use cases to reflect the above fixes.</li> </ul>

612  
613