



1
2
3
4

Document Number: DSP1108

Date: 2013-08-22

Version: 1.0.0

5 **Physical Computer System View Profile**

6 **Document Type: Specification**
7 **Document Status: DMTF Standard**
8 **Document Language: en-US**
9

10 Copyright Notice

11 Copyright © 2013 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26 withdrawn or modified after publication, and shall be indemnified and held harmless by any party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

33

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.....	23
45	8.1 CIM_PhysicalComputerSystemView.RequestStateChange()	23
46	8.2 CIM_PhysicalComputerSystemView.ClearLog()	24
47	8.3 CIM_PhysicalComputerSystemView.InstallSoftwareFromURI()	24
48	8.4 CIM_PhysicalComputerSystemView.SetOneTimeBootSource()	25
49	8.5 Profile conventions for operations	26
50	8.6 CIM_PhysicalComputerSystemView	26
51	8.7 CIM_ElementView	26
52	9 Use cases.....	27
53	9.1 Miscellaneous object diagrams.....	27
54	9.2 Small footprint managed computer system with additional CIM Schema implemented.....	27
55	9.3 Large managed computer system with additional CIM Profiles implemented.....	28
56	9.4 Managed system exposing multiple temperature sensors	29
57	9.5 Determine the power state of the physical computer system	31
58	9.6 Change the power state of the physical computer system	31
59	9.7 Get properties of a specific record log of the physical computer system	31
60	9.8 Browse the records of a log of the physical computer system	31
61	9.9 Monitor temperature sensor readings of the physical computer system	32
62	9.10 Configure a source of the physical computer system for next reboot only.....	32
63	9.11 Update the BIOS firmware of the physical computer system	32
64	10 CIM Elements.....	33
65	10.1 CIM_PhysicalComputerSystemView	33
66	10.2 CIM_RegisteredProfile.....	35
67	10.3 CIM_ElementView	36
68	10.4 CIM_ElementConformsToProfile	36
69	ANNEX A (informative) Change log.....	37

70

Figures

72	Figure 1 – Physical Computer System View Profile: Class diagram	11
73	Figure 2 – Simple PhysicalComputerSystemView implementation	27
74	Figure 3 – Small footprint PhysicalComputerSystemView implementation	28
75	Figure 4 – Rich PhysicalComputerSystemView implementation	29
76	Figure 5 – PhysicalComputerSystemView implementation of multiple temperature sensors	30

77

78

79 **Tables**

80	Table 1 – Referenced profiles	10
81	Table 2 – CIM_PhysicalComputerSystemView property model correspondence	13
82	Table 3 – Property origins for processor	18
83	Table 4 – Property origins for memory	18
84	Table 5 – Property origins for current BIOS or EFI firmware	19
85	Table 6 – Property origins for current management firmware	19
86	Table 7 – Property origins for the operating system	20
87	Table 8 – Property origins for the power allocation limit	20
88	Table 9 – Property origins for numeric sensors	21
89	Table 10 – Property origins for record logs	22
90	Table 11 – Property origin for boot sources	22
91	Table 12 – CIM_PhysicalComputerSystemView.RequestStateChange() method: Return code values ...	23
92	Table 13 – CIM_PhysicalComputerSystemView.RequestStateChange() method: Parameters	24
93	Table 14 – CIM_PhysicalComputerSystemView.ClearLog() method: Return code values	24
94	Table 15 – CIM_PhysicalComputerSystemView.ClearLog() method: Parameters	24
95	Table 16 – CIM_PhysicalComputerSystemView.InstallSoftwareFromURI() method: Return code	
96	values	25
97	Table 17 – CIM_PhysicalComputerSystemView.InstallSoftwareFromURI() method: Parameters	25
98	Table 18 – CIM_PhysicalComputerSystemView.SetOneTimeBootSource() method: Return code	
99	values	26
100	Table 19 – CIM_PhysicalComputerSystemView.SetOneTimeBootSource() method: Parameters	26
101	Table 20 – Operations: CIM_PhysicalComputerSystemView	26
102	Table 21 – CIM Elements: Physical Computer System View Profile	33
103	Table 22 – Class: CIM_PhysicalComputerSystemView	33
104	Table 23 – Class: CIM_RegisteredProfile	35
105	Table 24 – Class: CIM_ElementView	36
106	Table 25 – Class: CIM_ElementConformsToProfile	36
107		

108

Foreword

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

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

113 **Acknowledgments**

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

115 Editors:

- 116 • Steve Lee – Microsoft Corporation

117 Contributors:

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

123

124

Introduction

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

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

130 Document conventions

131 Typographical conventions

132 The following typographical conventions are used in this document:

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

138 ABNF usage conventions

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

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

143

Physical Computer System View Profile

144 1 Scope

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

150 2 Normative references

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

155 DMTF DSP0004, *CIM Infrastructure Specification 2.6*,
156 http://www.dmtf.org/standards/published_documents/DSP0004_2.6.pdf

157 DMTF DSP0200, *CIM Operations over HTTP 1.3*,
158 http://www.dmtf.org/standards/published_documents/DSP0200_1.3.pdf

159 DMTF DSP0223, *Generic Operations 1.0*,
160 http://www.dmtf.org/standards/published_documents/DSP0223_1.0.pdf

161 DMTF DSP1001, *Management Profile Specification Usage Guide 1.0*,
162 http://www.dmtf.org/standards/published_documents/DSP1001_1.0.pdf

163 DMTF DSP1009, *Sensors Profile 1.0*,
164 http://dmtof.org/sites/default/files/standards/documents/DSP1009_1.0.pdf

165 DMTF DSP1009, *Sensors Profile 1.1*,
166 http://dmtof.org/sites/default/files/standards/documents/DSP1009_1.1.pdf

167 DMTF DSP1010, *Record Log Profile 1.0*,
168 http://www.dmtf.org/sites/default/files/standards/documents/DSP1010_1.0.pdf

169 DMTF DSP1010, *Record Log Profile 2.0*,
170 http://www.dmtf.org/sites/default/files/standards/documents/DSP1010_2.0.pdf

171 DMTF DSP1011, *Physical Asset Profile 1.0*,
172 http://www.dmtf.org/sites/default/files/standards/documents/DSP1011_1.0.pdf

173 DMTF DSP1012, *Boot Control Profile 1.0*,
174 http://dmtof.org/sites/default/files/standards/documents/DSP1012_1.0.pdf

175 DMTF DSP1022, *CPU Profile 1.0*,
176 http://dmtof.org/sites/default/files/standards/documents/DSP1022_1.0.pdf

177 DMTF DSP1023, *Software Inventory Profile 1.0*,
178 http://dmtof.org/sites/default/files/standards/documents/DSP1023_1.0.pdf

- 179 DMTF DSP1025, *Software Update Profile 1.0*,
180 http://dmtf.org/sites/default/files/standards/documents/DSP1025_1.0.pdf
- 181 DMTF DSP1026, *System Memory Profile 1.0*,
182 http://dmtf.org/sites/default/files/standards/documents/DSP1026_1.0.pdf
- 183 DMTF DSP1029, *OS Status Profile 1.0*,
184 http://dmtf.org/sites/default/files/standards/documents/DSP1029_1.0.pdf
- 185 DMTF DSP1029, *OS Status Profile 1.1*,
186 http://dmtf.org/sites/default/files/standards/documents/DSP1029_1.1.pdf
- 187 DMTF DSP1033, *Profile Registration Profile 1.0*,
188 http://www.dmtf.org/standards/published_documents/DSP1033_1.0.pdf
- 189 DMTF DSP1052, *Computer System Profile 1.0*,
190 http://www.dmtf.org/sites/default/files/standards/documents/DSP1052_1.0.pdf
- 191 DMTF DSP1085, *Power Utilization Management Profile 1.0*,
192 http://dmtf.org/sites/default/files/standards/documents/DSP1085_1.0.pdf
- 193 IETF RFC5234, *ABNF: Augmented BNF for Syntax Specifications, January 2008*,
194 <http://tools.ietf.org/html/rfc5234>
- 195 ISO/IEC Directives, Part 2, *Rules for the structure and drafting of International Standards*,
196 <http://isotc.iso.org/livelink/livelink.exe?func=ll&objId=4230456&objAction=browse&sort=subtype>
- 197

198 3 Terms and definitions

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

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

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

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

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

214 3.1

215 **conditional**

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

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

232 **4 Symbols and abbreviated terms**

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

- 235 **4.1**
236 **BIOS**
237 Basic Input Output System
- 238 **4.2**
239 **EFI**
240 Extensible Firmware Interface
- 241 **4.3**
242 **FRU**
243 Field Replaceable Unit
- 244 **4.4**
245 **SKU**
246 Stock Keeping Unit
247

248 5 Synopsis

249 **Profile name:** Physical Computer System View

250 **Version:** 1.0.0s

251 **Organization:** DMTF

252 **CIM schema version:** 2.36

253 **Central class:** CIM_PhysicalComputerSystemView

254 **Scoping class:** CIM_ComputerSystem

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

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

260

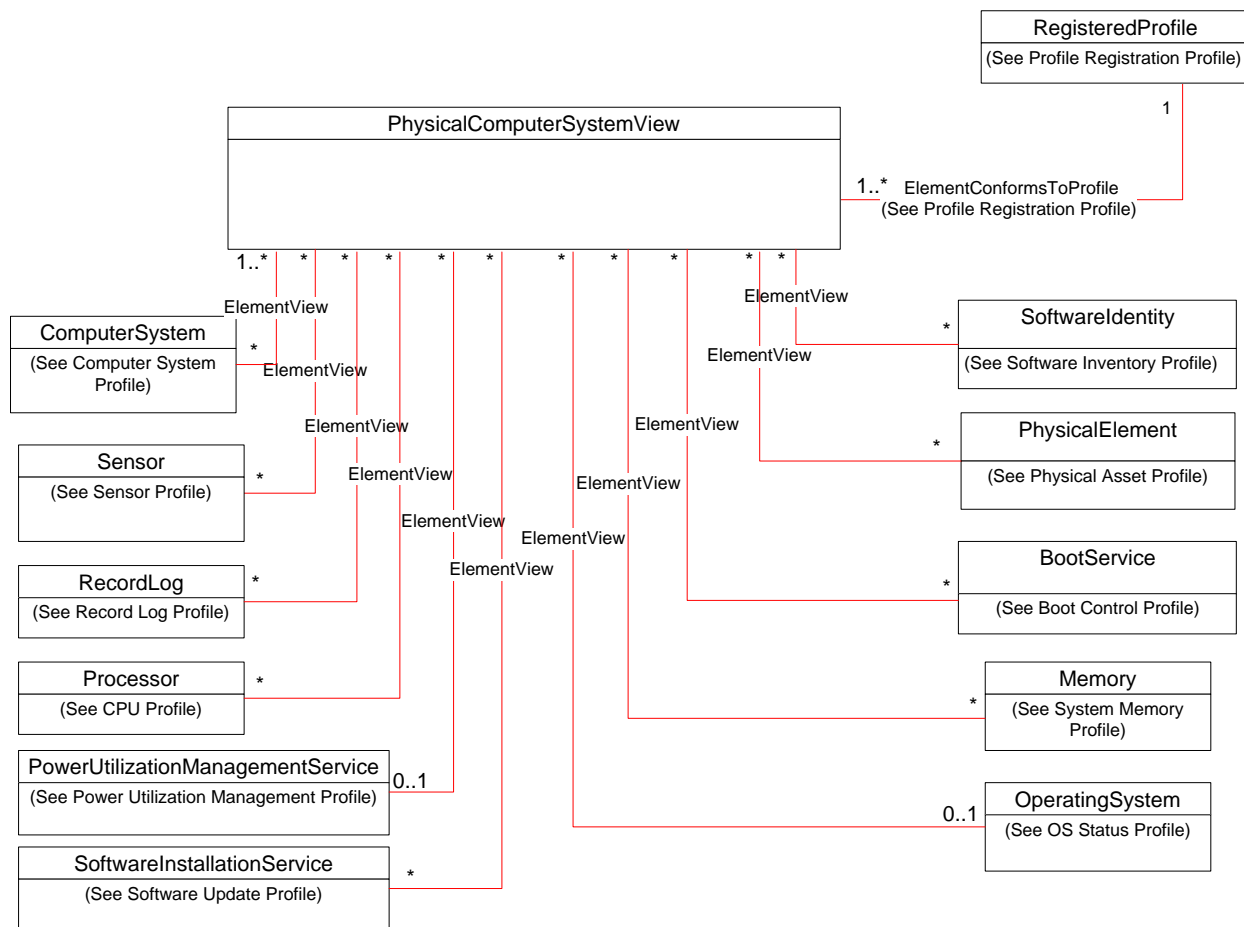
Table 1 – Referenced profiles

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

261 **6 Description**

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

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



266

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

268 **7 Implementation**

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

271 **7.1 Representing a physical computer system view**

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

277 CIM_PhysicalComputerSystemView property shall conform to requirements for the corresponding
278 property listed in the table as specified by the referenced profile.

279 **7.1.1 Representing information from multiple instances of the same class**

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

284 **7.1.1.1 Properties of each instance**

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

287 **7.1.1.2 Property correspondence**

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

290 **7.1.1.3 Matching property values to normalized instances**

291 For all properties with array correspondence with a class as in Table 2, the value at each index shall
292 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	DSP1011 1.0 Clause 10.2
	Tag	CIM_Chassis.Tag	DSP1011 1.0 Clause 10.2
	Manufacturer	CIM_Chassis.Manufacturer	DSP1011 1.0 Clause 10.2
	Model	CIM_Chassis.Model	DSP1011 1.0 Clause 10.2
	SKU	CIM_Chassis.SKU	DSP1011 1.0 Clause 10.2
	SerialNumber	CIM_Chassis.SerialNumber	DSP1011 1.0 Clause 10.2
	Version	CIM_Chassis.Version	DSP1011 1.0 Clause 10.2
	PartNumber	CIM_Chassis.PartNumber	DSP1011 1.0 Clause 10.2
DMTF:CPUView	NumberOfProcessors	Number of CIM_Processor instances associated to associated CIM_ComputerSystem	DSP1022 1.0 Clause 10.11
	NumberOfProcessor Cores	CIM_ProcessorCapabilities.NumberOfProcessorCores	DSP1022 1.0 Clause 10.12
	NumberOfProcessor Threads	CIM_ProcessorCapabilities.NumberOfHardwareThreads	DSP1022 1.0 Clause 10.12
	ProcessorFamily	CIM_Processor.Family	DSP1022 1.0 Clause 10.11
	ProcessorCurrent ClockSpeed	CIM_Processor.CurrentClockSpeed	DSP1022 1.0 Clause 10.11
	ProcessorMaxClock Speed	CIM_Processor.MaxClockSpeed	DSP1022 1.0 Clause 10.11
DMTF:System MemoryView	MemoryBlockSize	CIM_Memory.BlockSize	DSP1026 1.0 Clause 10.3
	MemoryNumberOf Blocks	CIM_Memory.NumberOfBlocks	DSP1026 1.0 Clause 10.3
	MemoryConsumable Blocks	CIM_Memory.ConsumableBlocks	DSP1026 1.0 Clause 10.3

Implemented Feature Value	CIM_Physical ComputerSystem View Property	Origin Class/Property or Model Correspondence	Referenced Profile
DMTF:Software InventoryView	CurrentBIOSMajorVersion	CIM_SoftwareIdentity.MajorVersion representing the BIOS/EFI	DSP1023 1.0 Clause 10.1
	CurrentBIOSMinorVersion	CIM_SoftwareIdentity.MinorVersion representing the BIOS/EFI	DSP1023 1.0 Clause 10.1
	CurrentBIOSRevisionNumber	CIM_SoftwareIdentity.RevisionNumber representing the BIOS/EFI	DSP1023 1.0 Clause 10.1
	CurrentBIOSBuildNumber	CIM_SoftwareIdentity.BuildNumber representing the BIOS/EFI	DSP1023 1.0 Clause 10.1
	CurrentBIOSVersionString	CIM_SoftwareIdentity.VersionString representing the BIOS/EFI	DSP1023 1.0 Clause 10.1
	CurrentManagementFirmwareMajorVersion	CIM_SoftwareIdentity.MajorVersion representing the management firmware	DSP1023 1.0 Clause 10.1
	CurrentManagementFirmwareMinorVersion	CIM_SoftwareIdentity.MinorVersion representing the management firmware	DSP1023 1.0 Clause 10.1
	CurrentManagementFirmwareRevisionNumber	CIM_SoftwareIdentity.RevisionNumber representing the management firmware	DSP1023 1.0 Clause 10.1
	CurrentManagementFirmwareBuildNumber	CIM_SoftwareIdentity.BuildNumber representing the management firmware	DSP1023 1.0 Clause 10.1
	CurrentManagementFirmwareElementName	CIM_SoftwareIdentity.ElementName representing the management firmware	DSP1023 1.0 Clause 10.1
	CurrentManagementFirmwareVersionString	CIM_SoftwareIdentity.VersionString representing the management firmware	DSP1023 1.0 Clause 10.1
DMTF:OSView	OSType	CIM_OperatingSystem.OSType	DSP1029 1.0 Clause 10.1
	OSEnabledState	CIM_OperatingSystem.EnabledState	DSP1029 1.0 Clause 10.1
	OSVersion	CIM_OperatingSystem.Version	DSP1029 1.1 Clause 10.1

Implemented Feature Value	CIM_Physical ComputerSystem View Property	Origin Class/Property or Model Correspondence	Referenced Profile
DMTF:Computer SystemView	OtherIdentifyingInfo	CIM_ComputerSystem.Other IdentifyingInfo	DSP1052 1.0 Clause 10.1
	IdentifyingDescriptions	CIM_ComputerSystem.Identifying Descriptions	DSP1052 1.0 Clause 10.1
	Dedicated	CIM_ComputerSystem.Dedicated	DSP1108 1.0 Clause 7.1.2.10.1
	OtherDedicated Descriptions	CIM_ComputerSystem.Other DedicatedDescriptions	DSP1108 1.0 Clause 7.1.2.10.2
	EnabledState	CIM_ComputerSystem.Enabled State	DSP1052 1.0 Clause 10.1
	RequestedState	CIM_ComputerSystem.Requested State	DSP1052 1.0 Clause 10.1
	OperationalStatus	CIM_ComputerSystem.Operational Status	DSP1052 1.0 Clause 10.1
DMTF:Power Utilization ManagementView	PowerUtilizationModes Supported	CIM_PowerUtilizationManagement Capabilities.PowerUtilizationModes Supported	DSP1085 1.0 Clause 10.2
	PowerUtilizationMode	CIM_PowerUtilizationManagement Service.PowerUtilizationMode	DSP1085 1.0 Clause 10.1
	PowerAllocationLimit	CIM_PowerAllocationSettingData.Limit	DSP1085 1.0 Clause 10.6

Implemented Feature Value	CIM_Physical ComputerSystem View Property	Origin Class/Property or Model Correspondence	Referenced Profile
DMTF:Numeric SensorView	NumericSensor ElementName	CIM_NumericSensor.ElementName	DSP1009 1.0 Clause 10.2
	NumericSensor EnabledState	CIM_NumericSensor.EnabledState	DSP1009 1.0 Clause 10.2
	NumericSensor Context	CIM_NumericSensor.Sensor Context	DSP1009 1.1 Clause 10.2
	NumericSensorHealth State	CIM_NumericSensor.HealthState	DSP1009 1.0 Clause 10.2
	NumericSensor CurrentState	CIM_NumericSensor.CurrentState	DSP1009 1.0 Clause 10.2
	NumericSensor PrimaryStatus	CIM_NumericSensor.PrimaryStatus	DSP1009 1.0 Clause 10.2
	NumericSensorBase Units	CIM_NumericSensor.BaseUnits	DSP1009 1.0 Clause 10.2
	NumericSensorUnit Modifier	CIM_NumericSensor.UnitModifier	DSP1009 1.0 Clause 10.2
	NumericSensorRate Units	CIM_NumericSensor.RateUnits	DSP1009 1.0 Clause 10.2
	NumericSensor CurrentReading	CIM_NumericSensor.Current Reading	DSP1009 1.0 Clause 10.2
	NumericSensorSensor Type	CIM_NumericSensor.SensorType	DSP1009 1.0 Clause 10.2
	NumericSensorOther SensorType Description	CIM_NumericSensor.OtherSensor TypeDescription	DSP1009 1.0 Clause 10.2
	NumericSensorUpper ThresholdNonCritical	CIM_NumericSensor.Upper ThresholdNonCritical	DSP1009 1.0 Clause 10.2
	NumericSensorUpper ThresholdFatal	CIM_NumericSensor.Upper ThresholdFatal	DSP1009 1.0 Clause 10.2
	NumericSensorUpper ThresholdCritical	CIM_NumericSensor.Upper ThresholdCritical	DSP1009 1.0 Clause 10.2
	NumericSensorLower ThresholdNonCritical	CIM_NumericSensor.Lower ThresholdNonCritical	DSP1009 1.0 Clause 10.2
	NumericSensorLower ThresholdFatal	CIM_NumericSensor.Lower ThresholdFatal	DSP1009 1.0 Clause 10.2
NumericSensorLower ThresholdCritical	CIM_NumericSensor.Lower ThresholdCritical	DSP1009 1.0 Clause 10.2	

Implemented Feature Value	CIM_Physical ComputerSystem View Property	Origin Class/Property or Model Correspondence	Referenced Profile
DMTF:Record LogView	LogInstanceID	CIM_RecordLog.InstanceID	DSP1010 1.0 Clause 10.5
	LogMaxNumberOfRecords	CIM_RecordLog.MaxNumberOfRecords	DSP1010 1.0 Clause 10.5
	LogCurrentNumberOfRecords	CIM_RecordLog.CurrentNumberOfRecords	DSP1010 1.0 Clause 10.5
	LogOverWritePolicy	CIM_RecordLog.OverwritePolicy	DSP1010 1.0 Clause 10.5
	LogState	CIM_RecordLog.LogState	DSP1010 1.0 Clause 10.5
DMTF:Boot ControlView	StructuredBootString	CIM_BootSourceSetting.StructuredBootString	DSP1012 1.0 Clause 10.6
	OneTimeBootSource	n/a	DSP1108 1.0 Clause 7.1.2.9.2

294 **7.1.2 Additional requirements**

295 This subclause details additional requirements for some properties of
296 CIM_PhysicalComputerSystemView.

297 **7.1.2.1 CIM_PhysicalComputerSystemView.InstanceID**

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

301 **7.1.2.2 Representing system processor information**

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

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

307 The NumberOfProcessorThreads property shall correspond to the NumberOfHardwareThreads property
308 in the CIM_ProcessorCapabilities class defined in [DSP1022](#).

309 The properties of the central processing unit of the physical computer system shall be represented as
310 properties defined in Table 3 from [DSP1022](#). When one or more instances of CIM_Processor are
311 instantiated and represented in the view class, each CIM_Processor instance should be associated with
312 the CIM_PhysicalComputerSystemView instance through an instance of CIM_ElementView where the
313 Antecedent property is a reference to the corresponding CIM_Processor instance and the Dependent
314 property is a reference to the CIM_PhysicalComputerSystemView instance.

315

Table 3 – Property origins for processor

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

316

7.1.2.3 Representing system memory information

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

323

Table 4 – Property origins for memory

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

324

7.1.2.4 Representing system software inventory

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

332 When an instance of CIM_SoftwareIdentity representing the current BIOS or EFI firmware is instantiated
 333 and represented in the view class, the CIM_SoftwareIdentity instance should be associated with the
 334 CIM_PhysicalComputerSystemView instance through an instance of CIM_ElementView where the
 335 Antecedent is a reference to the CIM_SoftwareIdentity instance and the Dependent property is a
 336 reference to the CIM_PhysicalComputerSystemView instance.

337

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

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

342 When an instance of CIM_SoftwareIdentity representing the current management firmware is instantiated
 343 and represented in the view class, the CIM_SoftwareIdentity instance should be associated with the
 344 CIM_PhysicalComputerSystemView instance through an instance of CIM_ElementView where the
 345 Antecedent is a reference to the CIM_SoftwareIdentity instance and the Dependent property is a
 346 reference to the CIM_PhysicalComputerSystemView instance.

347

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

348 **7.1.2.5 Representing operating system information**

349 This subclause describes the requirements for representing the running operating system for the
 350 CIM_PhysicalComputerSystemView class.

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

357

Table 7 – Property origins for the operating system

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

358 7.1.2.6 Representing power utilization information

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

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

370

Table 8 – Property origins for the power allocation limit

CIM_PhysicalComputerSystemView property name	CIM_PowerAllocationSettingData property name (origin)
PowerAllocationLimit	Limit

371 7.1.2.7 Representing system numeric sensors

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

380

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

381 **7.1.2.8 Representing system record logs**

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

386 Instances of CIM_RecordLog that contain information about the underlying computer system should be
 387 represented in the view class.

388 When instantiated, the CIM_RecordLog instance shall be associated with the
 389 CIM_PhysicalComputerSystemView instance through an instance of CIM_ElementView where the
 390 Antecedent property is a reference to the CIM_RecordLog instance and the Dependent property is a
 391 reference to the CIM_PhysicalComputerSystemView instance.

392

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

393 7.1.2.9 Representing system boot configuration

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

397 When an instance of CIM_BootSourceSetting is instantiated and represented in the view class, the
 398 CIM_BootService instance should be associated with the CIM_PhysicalComputerSystemView instance
 399 through an instance of CIM_ElementView where the Antecedent property is a reference to the
 400 CIM_BootService instance and the Dependent property is a reference to the
 401 CIM_PhysicalComputerSystemView instance.

402

Table 11 – Property origin for boot sources

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

403 7.1.2.9.1 CIM_PhysicalComputerSystemView.StructuredBootString

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

406 7.1.2.9.2 CIM_PhysicalComputerSystemView.OneTimeBootSource

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

410 7.1.2.10 Representing system identity information

411 7.1.2.10.1 CIM_PhysicalComputerSystemView.Dedicated

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

415 **7.1.2.10.2 CIM_PhysicalComputerSystemView.OtherDedicatedDescriptions**

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

420 **7.1.2.11 Representing system FRU information**

421 **7.1.2.11.1 CIM_PhysicalComputerSystemView.FRUInfoSupported**

422 When implemented according to Table 2, the FRUInfoSupported property shall correspond to the value of
 423 the CIM_PhysicalAssetCapabilities.FRUInfoSupported property value associated to the CIM_Chassis
 424 instance.

425 **8 Methods**

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

428 **8.1 CIM_PhysicalComputerSystemView.RequestStateChange()**

429 When the ImplementedFeatures property of the CIM_RegisteredProfile instance contains
 430 “DMTF:ComputerSystemView”, RequestStateChange() shall be implemented.

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.
1	Method is unsupported.
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() shall 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() shall 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 DSP1025 for additional details.
IN	Classifications[]	uint16	Identify the classification of software to install. See CIM_SoftwareIdentity.Classifications in DSP1023 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 DSP1025 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() shall 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

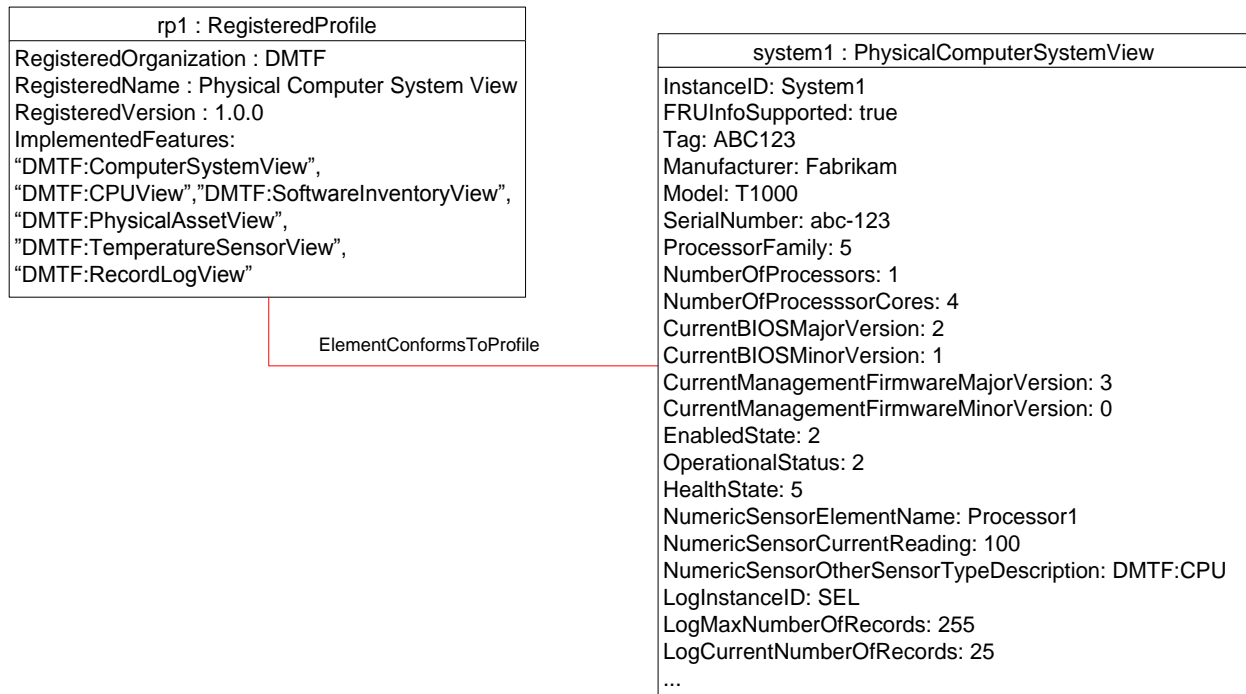
494 All operations in the default list in 8.5 shall be implemented as defined in [DSP0200](#).

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.

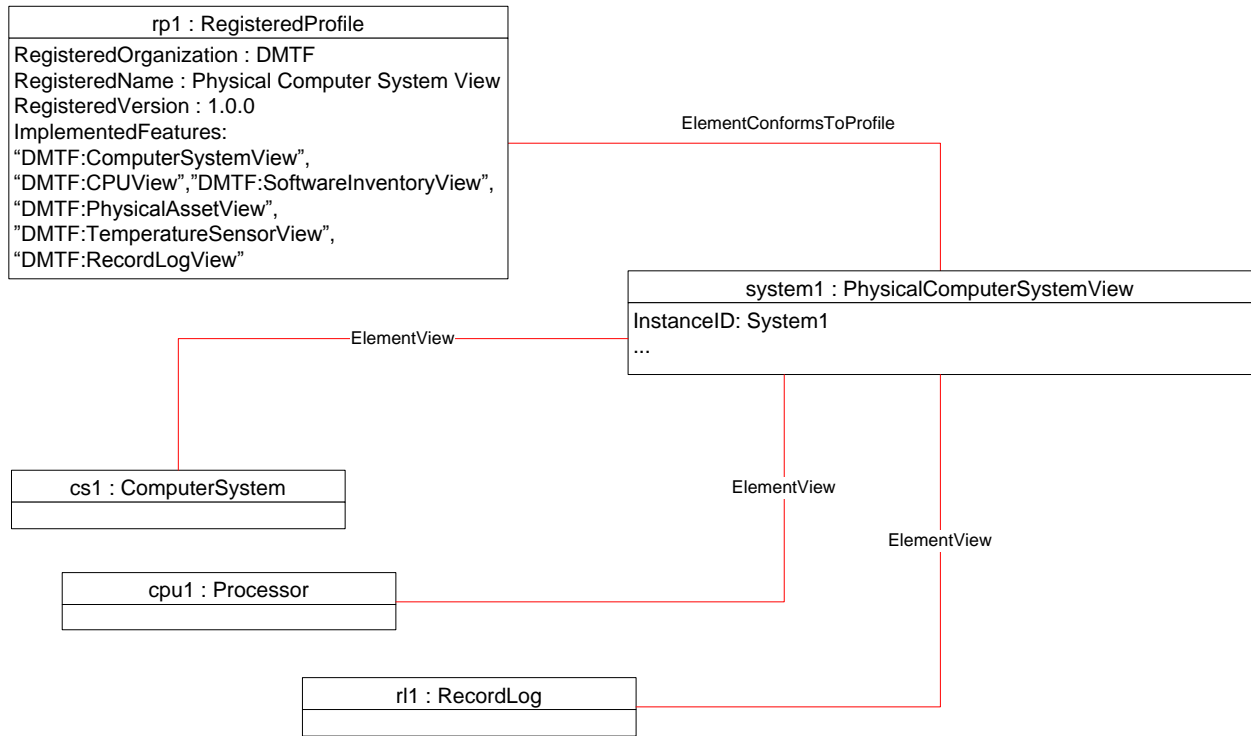


508

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.

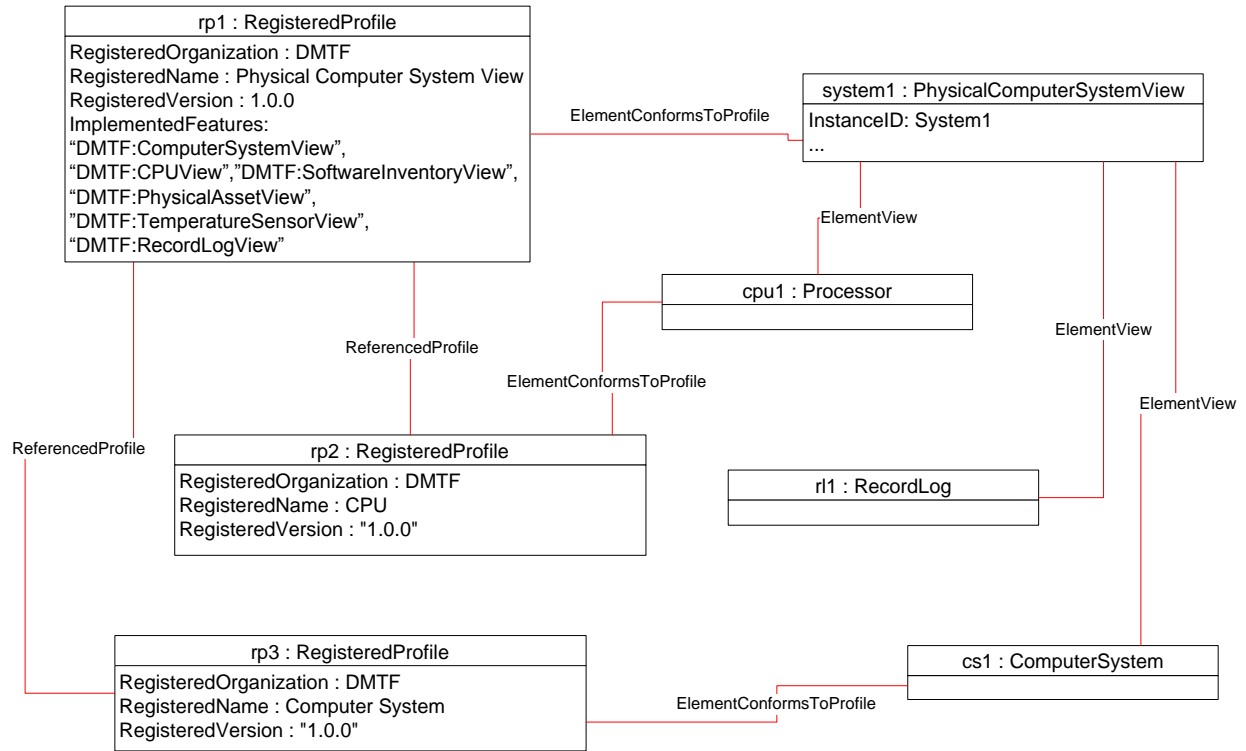


517
518

519 **Figure 3 – Small footprint PhysicalComputerSystemView implementation**

520 **9.3 Large managed computer system with additional CIM Profiles implemented**

521 The object diagram in Figure 4 shows a large managed computer system that implements the same
 522 features as the managed computer system in Figure 3, but also implements CIM Schema that is
 523 conformant to CIM Profiles providing richer management capabilities.



524

525

Figure 4 – Rich PhysicalComputerSystemView implementation

526

9.4 Managed system exposing multiple temperature sensors

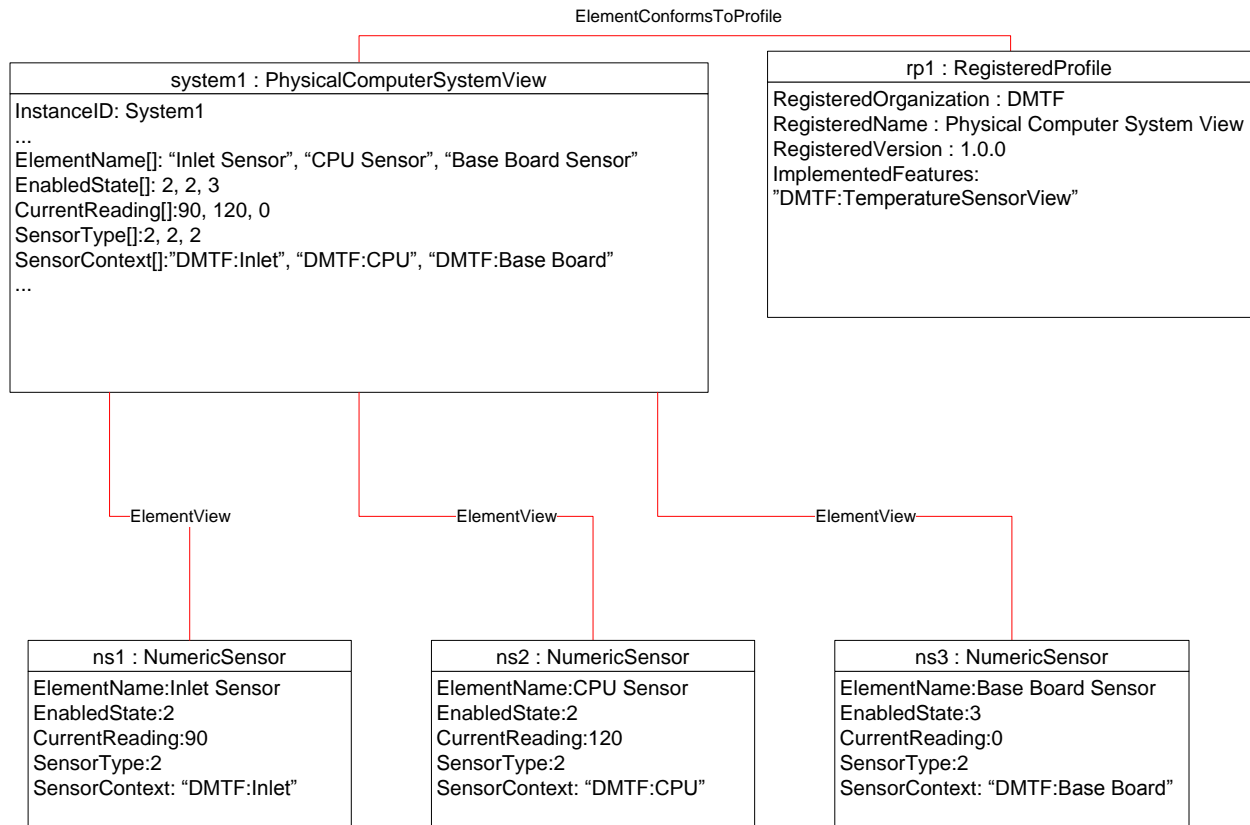
527

528

529

530

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.



531

532

Figure 5 – PhysicalComputerSystemView implementation of multiple temperature sensors

533 9.5 Determine the power state of the physical computer system

534 To determine the power state:

- 535 1) Verify that the managed computer system implements the DMTF:ComputerSystemView feature
536 by reading the ImplementedFeatures property of the CIM_RegisteredProfile instance.
- 537 2) Read the EnabledState property of the instance of CIM_PhysicalComputerSystemView.
- 538 3) See Table 12 for the mapping of power states to the EnabledState property.

539 9.6 Change the power state of the physical computer system

540 To change the power state:

- 541 1) Verify that the managed computer system implements the DMTF:ComputerSystemView feature
542 by reading the ImplementedFeatures property of the CIM_RegisteredProfile instance.
- 543 2) Invoke the RequestStateChange() method of the instance of CIM_PhysicalComputerSystemView
544 providing the new requested state.
- 545 3) See Table 12 for the mapping of power states to the EnabledState property.

546 9.7 Get properties of a specific record log of the physical computer system

547 To get the properties of a specific record log:

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

553 9.8 Browse the records of a log of the physical computer system

554 To browse log entries:

- 555 1) Verify that the managed computer system implements the DMTF:RecordLogView feature by
556 reading the ImplementedFeatures property of the CIM_RegisteredProfile instance.
- 557 2) Verify that an instance of CIM_RegisteredProfile for Record Log Profile exists.
- 558 3) Traverse the association to the CIM_RecordLog instance with the LogInstanceID by using the
559 CIM_ElementView association class.
- 560 4) Enumerate instances of CIM_LogEntry that are associated through the CIM_LogManagesRecord
561 association to the given instance of CIM_RecordLog.
562

563 9.9 Monitor temperature sensor readings of the physical computer system

564 To monitor CPU temperature sensor readings:

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

572 9.10 Configure a source of the physical computer system for next reboot only

573 To configure the boot source for next reboot only:

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

579 9.11 Update the BIOS firmware of the physical computer system

580 To update the BIOS firmware:

- 581 1) Verify that the managed computer system implements the DMTF:SoftwareUpdateView feature by
582 reading the ImplementedFeatures property of the CIM_RegisteredProfile instance.
- 583 2) Invoke the InstallSoftwareFromURI() method of the instance of
584 CIM_PhysicalComputerSystemView providing the classification value of 6 (Firmware/BIOS),
585 location of the software, and required options.
586

587 **10 CIM Elements**

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

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

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

591 **10.1 CIM_PhysicalComputerSystemView**

592 CIM_PhysicalComputerSystemView describes a physical computer system and associated management
 593 information in a managed computer system.

594 **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()	Conditional	See 8.1.
ClearLog()	Conditional	See 8.2.
InstallSoftwareFromURI()	Conditional	See 8.3.
SetOneTimeBootSource()	Conditional	See 8.4.

595 **10.2 CIM_RegisteredProfile**

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

601 **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 "1.0.0".
RegisteredOrganization	Mandatory	This property shall have a value of "DMTF".

Elements	Requirement	Description
ImplementedFeatures	Mandatory	<p>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".</p> <p>Presences of values in this property only indicate implemented properties/methods in CIM_PhysicalComputerSystemView and do not indicate conformance to additional CIM Profiles.</p>

602 10.3 CIM_ElementView

603 CIM_ElementView associates additional CIM Schema to the CIM_PhysicalComputerSystemView
604 instance.

605 **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.

606 10.4 CIM_ElementConformsToProfile

607 CIM_ElementConformsToProfile associates the instance of CIM_RegisteredProfile to the
608 CIM_PhysicalComputerSystemView instance.

609 **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.

610
611
612
613

ANNEX A (informative)

Change log

Version	Date	Description
1.0.0	2013-08-22	

614
615