



1  
2  
3  
4

**Document Number: DSP1108**

**Date: 2012-11-12**

**Version: 1.0.0i**

## 5 **Physical Computer System View Profile**

### **Information for Work-in-Progress version:**

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

**It expires on: 2013-05-15**

**Provide any comments through the DMTF Feedback Portal:**

<http://www.dmtf.org/standards/feedback>

6 **Document Type: Specification**  
7 **Document Status: Work in Progress**  
8 **Document Language: en-US**  
9

## 10 Copyright Notice

11 Copyright © 2012 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26 withdrawn or modified after publication, and shall be indemnified and held harmless by any party  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..... 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.ModifyPersistentBootConfigOrder( ) ..... 24

49 8.5 CIM\_PhysicalComputerSystemView.SetOneTimeBootSource( ) ..... 25

50 8.6 Profile conventions for operations ..... 25

51 8.7 CIM\_PhysicalComputerSystemView ..... 26

52 8.8 CIM\_ElementView ..... 26

53 9 Use cases..... 27

54 9.1 Miscellaneous object diagrams..... 27

55 9.2 Small footprint managed computer system with additional CIM Schema implemented..... 27

56 9.3 Large managed computer system with additional CIM Profiles implemented..... 28

57 9.4 Managed system exposing multiple temperature sensors ..... 29

58 9.5 Determine the power state of the physical computer system ..... 31

59 9.6 Change the power state of the physical computer system ..... 31

60 9.7 Get properties of a specific record log of the physical computer system ..... 31

61 9.8 Browse the records of a log of the physical computer system ..... 31

62 9.9 Monitor temperate sensor readings of the physical computer system ..... 32

63 9.10 Modify persisted boot order of the physical computer system..... 32

64 9.11 Configure a source of the physical computer system for next reboot only ..... 32

65 9.12 Update the BIOS firmware of the physical computer system ..... 32

66 10 CIM Elements..... 33

67 10.1 CIM\_PhysicalComputerSystemView ..... 33

68 10.2 CIM\_RegisteredProfile..... 35

69 10.3 CIM\_ElementView ..... 36

70 ANNEX A (informative) Change log..... 37

71

## 72 Figures

73 Figure 1 – Physical Computer System View Profile: Class diagram ..... 11

74 Figure 2 – Simple PhysicalComputerSystemView implementation ..... 27

75 Figure 3 – Small footprint PhysicalComputerSystemView implementation ..... 28

76 Figure 4 – Rich PhysicalComputerSystemView implementation ..... 29

77 Figure 5 – PhysicalComputerSystemView implementation of multiple temperature sensors ..... 30

78

79

80

81 **Tables**

82	Table 1 – Referenced profiles .....	10
83	Table 2 – CIM_PhysicalComputerSystemView property model correspondence .....	13
84	Table 3 – Property origins for processor .....	17
85	Table 4 – Property origins for memory .....	17
86	Table 5 – Property origins for current BIOS or EFI firmware .....	17
87	Table 6 – Property origins for current management firmware .....	18
88	Table 7 – Property origins for the operating system .....	18
89	Table 8 – Property origins for the power allocation limit .....	19
90	Table 9 – Property origins for numeric sensors .....	20
91	Table 10 – Property origins for record logs .....	21
92	Table 11 – Property origin for boot sources .....	21
93	Table 12 – CIM_PhysicalComputerSystemView.RequestStateChange( ) method: Return code values ...	22
94	Table 13 – CIM_PhysicalComputerSystemView.RequestStateChange( ) method: Parameters .....	23
95	Table 14 – CIM_PhysicalComputerSystemView.ClearLog( ) method: Return code values .....	23
96	Table 15 – CIM_PhysicalComputerSystemView.ClearLog( ) method: Parameters .....	23
97	Table 16 – CIM_PhysicalComputerSystemView.InstallSoftwareFromURI( ) method: Return code values	24
98	Table 17 – CIM_PhysicalComputerSystemView.InstallSoftwareFromURI( ) method: Parameters.....	24
99	Table 18 – CIM_PhysicalComputerSystemView.ModifyPersistentBootConfigOrder( ) method: Return code	
100	values .....	24
101	Table 19 – CIM_PhysicalComputerSystemView.ModifyPersistentBootConfigOrder( ) method: Parameters	25
102	Table 20 – CIM_PhysicalComputerSystemView.SetOneTimeBootSource( ) method: Return code values	25
103	Table 21 – CIM_PhysicalComputerSystemView.SetOneTimeBootSource( ) method: Parameters .....	25
104	Table 22 – Operations: CIM_PhysicalComputerSystemView.....	26
105	Table 23 – CIM Elements: Physical Computer System View Profile .....	33
106	Table 24 – Class: CIM_PhysicalComputerSystemView .....	33
107	Table 25 – Class: CIM_RegisteredProfile .....	35
108	Table 26 – Class: CIM_ElementView .....	36
109	Table 27 – Class: CIM_ElementView .....	36
110		

111

## Foreword

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

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

## 116 Acknowledgments

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

118 Editors:

- 119 • Steve Lee – Microsoft Corporation

120 Contributors:

- 121 • Jeff Hilland – Hewlett-Packard Company
- 122 • John Leung – Intel Corporation
- 123 • Hemal Shah – Broadcom Corporation
- 124 • Satheesh Thomas – AMI
- 125 • Perry Vincent – Intel Corporation

126

127

## Introduction

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

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

### 133 Document conventions

#### 134 Typographical conventions

135 The following typographical conventions are used in this document:

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

#### 141 ABNF usage conventions

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

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

146

# Physical Computer System View Profile

## 147 1 Scope

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

## 153 2 Normative references

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

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

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

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

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

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

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

170 DMTF DSP1010, *Record Log Profile 1.0*,  
171 [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)

172 DMTF DSP1010, *Record Log Profile 2.0*,  
173 [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)

174 DMTF DSP1011, *Physical Asset Profile 1.0*,  
175 [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)

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

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

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

182 DMTF DSP1025, *Software Update Profile 1.0*,  
183 [http://dmtf.org/sites/default/files/standards/documents/DSP1025\\_1.0.pdf](http://dmtf.org/sites/default/files/standards/documents/DSP1025_1.0.pdf)

- 184 DMTF DSP1026, *System Memory Profile 1.0*,  
185 [http://dmtof.org/sites/default/files/standards/documents/DSP1026\\_1.0.pdf](http://dmtof.org/sites/default/files/standards/documents/DSP1026_1.0.pdf)
- 186 DMTF DSP1029, *OS Status Profile 1.0*,  
187 [http://dmtof.org/sites/default/files/standards/documents/DSP1029\\_1.0.pdf](http://dmtof.org/sites/default/files/standards/documents/DSP1029_1.0.pdf)
- 188 DMTF DSP1029, *OS Status Profile 1.1*,  
189 [http://dmtof.org/sites/default/files/standards/documents/DSP1029\\_1.1.pdf](http://dmtof.org/sites/default/files/standards/documents/DSP1029_1.1.pdf)
- 190 DMTF DSP1033, *Profile Registration Profile 1.0*,  
191 [http://www.dmtf.org/standards/published\\_documents/DSP1033\\_1.0.pdf](http://www.dmtf.org/standards/published_documents/DSP1033_1.0.pdf)
- 192 DMTF DSP1052, *Computer System Profile 1.0*,  
193 [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)
- 194 DMTF DSP1052, *Computer System Profile 1.1*,  
195 [http://www.dmtf.org/sites/default/files/standards/documents/DSP1052\\_1.1.pdf](http://www.dmtf.org/sites/default/files/standards/documents/DSP1052_1.1.pdf)
- 196 DMTF DSP1085, *Power Utilization Management Profile 1.0*,  
197 [http://dmtof.org/sites/default/files/standards/documents/DSP1085\\_1.0.pdf](http://dmtof.org/sites/default/files/standards/documents/DSP1085_1.0.pdf)
- 198 IETF RFC5234, *ABNF: Augmented BNF for Syntax Specifications, January 2008*,  
199 <http://tools.ietf.org/html/rfc5234>
- 200 ISO/IEC Directives, Part 2, *Rules for the structure and drafting of International Standards*,  
201 <http://isotc.iso.org/livelink/livelink.exe?func=ll&objId=4230456&objAction=browse&sort=subtype>
- 202

### 203 3 Terms and definitions

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

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

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

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

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

#### 219 3.1

##### 220 **can**

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

#### 222 3.2

##### 223 **cannot**

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



- 225 **3.3**  
226 **conditional**  
227 indicates requirements to be followed strictly to conform to the document when the specified conditions  
228 are met
- 229 **3.4**  
230 **mandatory**  
231 indicates requirements to be followed strictly to conform to the document and from which no deviation is  
232 permitted
- 233 **3.5**  
234 **may**  
235 indicates a course of action permissible within the limits of the document
- 236 **3.6**  
237 **need not**  
238 indicates a course of action permissible within the limits of the document
- 239 **3.7**  
240 **optional**  
241 indicates a course of action permissible within the limits of the document
- 242 **3.8**  
243 **referencing profile**  
244 indicates a profile that owns the definition of this class and can include a reference to this profile in its  
245 "Referenced Profiles" table
- 246 **3.9**  
247 **shall**  
248 indicates requirements to be followed strictly to conform to the document and from which no deviation is  
249 permitted
- 250 **3.10**  
251 **shall not**  
252 indicates requirements to be followed strictly to conform to the document and from which no deviation is  
253 permitted
- 254 **3.11**  
255 **should**  
256 indicates that among several possibilities, one is recommended as particularly suitable, without  
257 mentioning or excluding others, or that a certain course of action is preferred but not necessarily required
- 258 **3.12**  
259 **should not**  
260 indicates that a certain possibility or course of action is deprecated but not prohibited
- 261 **3.13**  
262 **unspecified**  
263 indicates that this profile does not define any constraints for the referenced CIM element or operation

## 264 **4 Symbols and abbreviated terms**

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

- 267 **4.1**  
 268 **BIOS**  
 269 Basic Input Output System
- 270 **4.2**  
 271 **EFI**  
 272 Extensible Firmware Interface
- 273 **4.3**  
 274 **FRU**  
 275 Field Replaceable Unit
- 276 **4.4**  
 277 **SKU**  
 278 Stock Keeping Unit

## 279 5 Synopsis

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

281 **Version:** 1.0.0

282 **Organization:** DMTF

283 **CIM schema version:** 2.33

284 **Central class:** CIM\_PhysicalComputerSystemView

285 **Scoping class:** CIM\_ComputerSystem

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

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

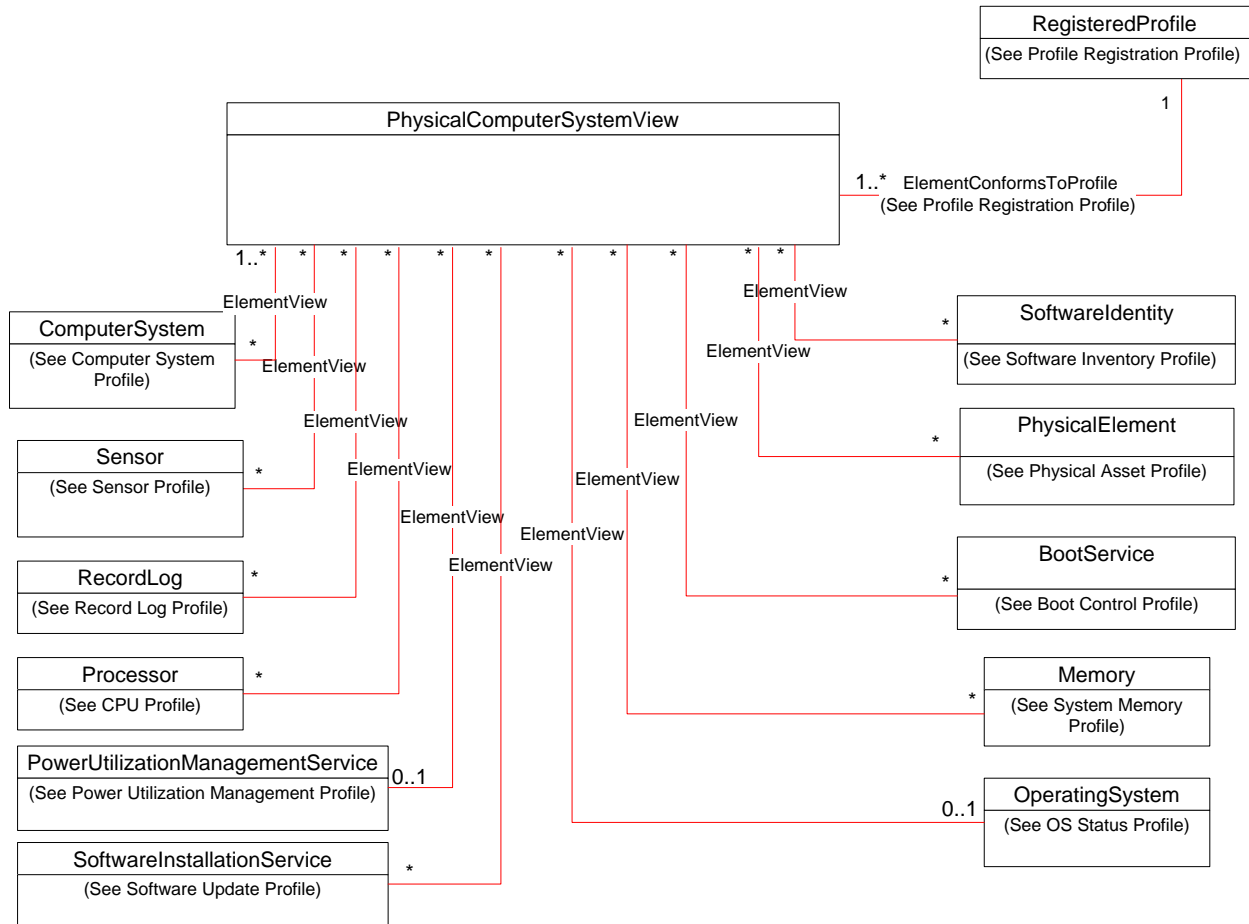
291 **Table 1 – Referenced profiles**

Profile Name	Organization	Version	Requirement	Description
Profile Registration	DMTF	1.0	Mandatory	None
Computer System	DMTF	1.1	Optional	None
Power Utilization Management	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	1.1	Optional	None
CPU	DMTF	1.0	Optional	None
Software Update	DMTF	1.0	Optional	None

## 292 6 Description

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

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



297

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

## 299 7 Implementation

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

### 302 7.1 Representing a physical computer system view

303 An instance of CIM\_PhysicalComputerSystemView shall represent the managed computer system. The  
 304 properties of the instance of CIM\_PhysicalComputerSystemView shall reflect the current state and  
 305 configuration of a managed computer system conforming to the referenced profiles. When the  
 306 ImplementedFeatures property of the CIM\_RegisteredProfile instance contains the value listed in Table 2  
 307 below, then the corresponding CIM\_PhysicalComputerSystemView property shall conform to  
 308 requirements for the corresponding property listed in the table as specified by the referenced profile.

**309 7.1.1 Representing information from multiple instances of the same class.**

310 A subset of properties CIM\_PhysicalComputerSystemView may represent information from multiple  
311 instances of same class. Indexed Arrays are used in CIM\_PhysicalComputerSystemView for such  
312 properties. The value in the array correspondence column in

313 Table , identifies the class whose multiple instances are represented by the corresponding array property  
314 in CIM\_PhysicalComputerSystemView.

**315 7.1.1.1 Properties of each instance**

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

**318 7.1.1.2 Property correspondence**

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

**321 7.1.1.3 Matching property values to normalized instances**

322 For all properties with array correspondence with a class as in Table 2, the value at each index shall  
323 match the value of the corresponding property of the corresponding instance of the 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 7.4
	Tag	CIM_Chassis.Tag	<a href="#">DSP1011 1.0</a> Clause 10.2
	Manufacturer	CIM_Chassis.Manufacturer	<a href="#">DSP1011 1.0</a> Clause 7.4
	Model	CIM_Chassis.Model	<a href="#">DSP1011 1.0</a> Clause 7.4
	SKU	CIM_Chassis.SKU	<a href="#">DSP1011 1.0</a> Clause 7.4
	SerialNumber	CIM_Chassis.SerialNumber	<a href="#">DSP1011 1.0</a> Clause 7.4
	Version	CIM_Chassis.Version	<a href="#">DSP1011 1.0</a> Clause 10.2
	PartNumber	CIM_Chassis.PartNumber	<a href="#">DSP1011 1.0</a> Clause 7.4
DMTF:CPUView	NumberOfProcessors	Number of CIM_Processor instances associated to associated CIM_ComputerSystem	<a href="#">DSP1022 1.0</a> Clause 7.1
	NumberOfProcessor Cores	CIM_ProcessorCapabilitiesNumberOfProcessorCores	<a href="#">DSP1022 1.0</a> Clause 7.2.1
	NumberOfProcessor Threads	CIM_ProcessorCapabilitiesNumberOfHardwareThreads	<a href="#">DSP1022 1.0</a> Clause 7.2.1
	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 7.2
	MemoryConsumable Blocks	CIM_Memory.ConsumableBlocks	<a href="#">DSP1026 1.0</a> Clause 7.2
DMTF:Software InventoryView	CurrentBIOSMajor Version	CIM_SoftwareIdentity.MajorVersion representing the BIOS/EFI	<a href="#">DSP1023 1.0</a> Clause 7.3
	CurrentBIOSMinor Version	CIM_SoftwareIdentity.MinorVersion representing the BIOS/EFI	<a href="#">DSP1023 1.0</a> Clause 7.3
	CurrentBIOSRevision Number	CIM_SoftwareIdentity.Revision Number representing the BIOS/EFI	<a href="#">DSP1023 1.0</a> Clause 7.3
	CurrentBIOSBuild Number	CIM_SoftwareIdentity.BuildNumber representing the BIOS/EFI	<a href="#">DSP1023 1.0</a> Clause 7.3

Implemented Feature Value	CIM_Physical ComputerSystem View Property	Origin Class/Property or Model Correspondence	Referenced Profile
	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 7.3
	CurrentManagementFirmwareMinorVersion	CIM_SoftwareIdentity.MinorVersion representing the management firmware	<a href="#">DSP1023 1.0</a> Clause 7.3
	CurrentManagementFirmwareRevisionNumber	CIM_SoftwareIdentity.RevisionNumber representing the management firmware	<a href="#">DSP1023 1.0</a> Clause 7.3
	CurrentManagementFirmwareBuildNumber	CIM_SoftwareIdentity.BuildNumber representing the management firmware	<a href="#">DSP1023 1.0</a> Clause 7.3
	CurrentManagementFirmwareElementName	CIM_SoftwareIdentity.ElementName representing the management firmware	<a href="#">DSP1023 1.0</a> Clause 7.3
	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 7.4.1
	OSVersion	CIM_OperatingSystem.Version	<a href="#">DSP1029 1.1</a> Clause 7.4.6
DMTF:Computer SystemView	OtherIdentifyingInfo	CIM_ComputerSystem.OtherIdentifyingInfo	<a href="#">DSP1052 1.0</a> Clause 7.1.1
	IdentifyingDescriptions	CIM_ComputerSystem.IdentifyingDescriptions	<a href="#">DSP1052 1.0</a> Clause 7.1.1
	Dedicated	CIM_ComputerSystem.Dedicated	<a href="#">DSP1052 1.1</a>
	OtherDedicatedDescriptions	CIM_ComputerSystem.OtherDedicatedDescriptions	<a href="#">DSP1052 1.1</a>
	EnabledState	CIM_ComputerSystem.EnabledState	<a href="#">DSP1052 1.0</a> Clause 7.7.1.1
	RequestedState	CIM_ComputerSystem.RequestedState	<a href="#">DSP1052 1.0</a> Clause 7.7.1.1
	OperationalStatus	CIM_ComputerSystem.OperationalStatus	<a href="#">DSP1052 1.0</a> Clause 10.1
	HealthState	CIM_ComputerSystem.HealthState	<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
DMTF:Power Utilization Management View	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 7.2.1
	PowerAllocationUnits	CIM_PowerAllocationSettingData.AllocationUnits	<a href="#">DSP1085 1.0</a> Clause 7.2.1
DMTF: Temperature SensorView	NumericSensor ElementName	CIM_NumericSensor.ElementName	<a href="#">DSP1009 1.0</a> Clause 7.19
	NumericSensor EnabledState	CIM_NumericSensor.EnabledState	<a href="#">DSP1009 1.0</a> Clause 7.16
	NumericSensor Context	CIM_NumericSensor.Sensor Context	<a href="#">DSP1009 1.1</a> Clause 7.20
	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 7.4
	NumericSensor PrimaryStatus	CIM_NumericSensor.PrimaryStatus	<a href="#">DSP1009 1.0</a> Clause 7.4
	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 7.17
	NumericSensorUpper ThresholdNonCritical	CIM_NumericSensor.Upper ThresholdNonCritical	<a href="#">DSP1009 1.0</a> Clause 7.6
	NumericSensorUpper ThresholdFatal	CIM_NumericSensor.Upper ThresholdFatal	<a href="#">DSP1009 1.0</a> Clause 7.10
	NumericSensorUpper ThresholdCritical	CIM_NumericSensor.Upper ThresholdCritical	<a href="#">DSP1009 1.0</a> Clause 7.8
	NumericSensorLower ThresholdNonCritical	CIM_NumericSensor.Lower ThresholdNonCritical	<a href="#">DSP1009 1.0</a> Clause 7.6
NumericSensorLower ThresholdFatal	CIM_NumericSensor.Lower ThresholdFatal	<a href="#">DSP1009 1.0</a> Clause 7.10	

Implemented Feature Value	CIM_PhysicalComputerSystemView Property	Origin Class/Property or Model Correspondence	Referenced Profile
	NumericSensorLowerThresholdCritical	CIM_NumericSensor.LowerThresholdCritical	<a href="#">DSP1009 1.0</a> Clause 7.8
DMTF:RecordLogView	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 7.8
	LogState	CIM_RecordLog.LogState	<a href="#">DSP1010 1.0</a> Clause 7.5.2
DMTF:BootControlView	OneTimeBootSource	CIM_BootSourceSetting.StructuredBootString	DSP1108 Clause 7.1.2.9.1
	PersistentBootConfigOrder	CIM_BootSourceSetting.StructuredBootString	DSP1108 Clause 7.1.2.9.2

325

## 326 7.1.2 Additional requirements

327 This subclause details additional requirements on some properties of  
328 CIM\_PhysicalComputerSystemView.

### 329 7.1.2.1 CIM\_PhysicalComputerSystemView.InstanceID

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

### 333 7.1.2.2 Representing system processor information

334 The intent of this set of properties is to model the central processing unit.

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

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

340 The properties of the central processing unit of the physical system shall be represented as properties  
341 defined in Table 3 from [DSP1022](#). When one or more instances of CIM\_Processor are instantiated and  
342 represented in the view class, each CIM\_Processor instance should be associated with the  
343 CIM\_PhysicalComputerSystemView instance via an instance of CIM\_ElementView where the Antecedent  
344 property is a reference to the corresponding CIM\_Processor instance and the Dependent property is a  
345 reference to the CIM\_PhysicalComputerSystemView instance.



346

**Table 3 – Property origins for processor**

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

### 347 7.1.2.3 Representing system memory information

348 The properties of the memory of the physical system shall be represented as properties defined in Table  
 349 4 from [DSP1026](#). When an instance of CIM\_Memory is instantiated and represented in the view class,  
 350 the CIM\_Memory instance should be associated with the CIM\_PhysicalComputerSystemView instance  
 351 via an instance of CIM\_ElementView where the Antecedent property is a reference to the CIM\_Memory  
 352 instance and the Dependent property is a reference to the CIM\_PhysicalComputerSystemView instance.

353

**Table 4 – Property origins for memory**

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

### 354 7.1.2.4 Representing system software inventory

355 The properties of the current BIOS or EFI firmware of the physical system shall be represented as  
 356 properties defined in Table 5 from [DSP1023](#). The current BIOS or EFI firmware property values shall  
 357 correspond to the instance of CIM\_SoftwareIdentity referenced by the instance of  
 358 CIM\_ElementSoftwareIdentity where the ElementSoftwareStatus property has a value of 2 (Current).  
 359 When an instance of CIM\_SoftwareIdentity representing the current BIOS or EFI firmware is instantiated  
 360 and represented in the view class, the CIM\_SoftwareIdentity instance should be associated with the  
 361 CIM\_PhysicalComputerSystemView instance via an instance of CIM\_ElementView where the Antecedent  
 362 is a reference to the CIM\_SoftwareIdentity instance and the Dependent property is a reference to the  
 363 CIM\_PhysicalComputerSystemView instance.

364

**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

365 The properties of the current management firmware of the physical system shall be represented as  
 366 properties defined in Table 6 from [DSP1023](#). The current management firmware property values shall  
 367 correspond to the instance of CIM\_SoftwareIdentity referenced by the instance of

368 CIM\_ElementSoftwareIdentity where the ElementSoftwareStatus property has a value of 2 (Current).  
 369 When an instance of CIM\_SoftwareIdentity representing the current management firmware is instantiated  
 370 and represented in the view class, the CIM\_SoftwareIdentity instance should be associated with the  
 371 CIM\_PhysicalComputerSystemView instance via an instance of CIM\_ElementView where the Antecedent  
 372 is a reference to the CIM\_SoftwareIdentity instance and the Dependent property is a reference to the  
 373 CIM\_PhysicalComputerSystemView instance.

374 **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

#### 375 7.1.2.5 Representing operating system information

376 This section describes the requirements for representing the running operating system for the  
 377 CIM\_PhysicalComputerSystemView class.

378 The properties of the operating system of the physical computer system shall be represented as  
 379 properties as defined in Table 7 from [DSP1029](#). When an instance of CIM\_OperatingSystem is  
 380 instantiated and represented in the view class, the CIM\_OperatingSystem instance should be associated  
 381 with the CIM\_PhysicalComputerSystemView instance via an instance of CIM\_ElementView where the  
 382 Antecedent property is a reference to the CIM\_OperatingSystem instance and the Dependent property is  
 383 a reference to the CIM\_PhysicalComputerSystemView instance.

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

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

#### 385 7.1.2.6 Representing power utilization information

386 The PowerAllocationLimit property shall correspond to the Limit property of the  
 387 CIM\_PowerAllocationSettingData class defined in [DSP1085](#) with the added “PowerAllocation” prefix to  
 388 logically group properties related to power utilization and avoid naming collision.

389 The power allocation limit of the physical computer system shall be represented as the property as  
 390 defined in Table 8 from [DSP1085](#). When an instance of CIM\_PowerAllocationSettingData is instantiated  
 391 and represented in the view class, the CIM\_PowerUtilizationManagementService instance should be  
 392 associated with the CIM\_PhysicalComputerSystemView instance via an instance of CIM\_ElementView  
 393 where the Antecedent property is a reference to the CIM\_PowerUtilizationManagementService instance  
 394 and the Dependent property is a reference to the CIM\_PhysicalComputerSystemView instance.

395

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

CIM_PhysicalComputerSystemView property name	CIM_PowerAllocationSettingData property name (origin)
PowerAllocationLimit	Limit
PowerAllocationUnits	AllocationUnits

396 **7.1.2.7 Representing system numeric sensors**

397 The properties of a numeric sensor of the physical computer system shall be represented as elements of  
 398 a group of indexed array properties as defined in Table 9 from [DSP1009](#). For these properties, the array  
 399 elements with same index shall present a view of the same numeric sensor. When an instance of  
 400 CIM\_NumericSensor is instantiated and represented in the view class, the CIM\_NumericSensor instance  
 401 should be associated with the CIM\_PhysicalComputerSystemView instance via an instance of  
 402 CIM\_ElementView where the Antecedent property is a reference to the CIM\_NumericSensor instance  
 403 and the Dependent property is a reference to the CIM\_PhysicalComputerSystemView instance.

404

**Table 9 – Property origins for numeric sensors**

<b>CIM_PhysicalComputerSystemView property name</b>	<b>CIM_NumericSensor property name (origin)</b>
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

#### 405 7.1.2.8 Representing system record logs

406 The properties of a record log of the physical computer system shall be represented as elements of a  
407 group of indexed array properties as defined in Table 10 from [DSP1010](#). For these properties, the array  
408 elements with same index shall present a view of the same record log. When an instance of  
409 CIM\_RecordLog is instantiated and represented in the view class, the CIM\_RecordLog instance should  
410 be associated with the CIM\_PhysicalComputerSystemView instance via an instance of CIM\_ElementView  
411 where the Antecedent property is a reference to the CIM\_RecordLog instance and the Dependent  
412 property is a reference to the CIM\_PhysicalComputerSystemView instance.

413

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

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

414 **7.1.2.9 Representing system boot configuration**

415 The enabled boot sources of the persistent default boot configuration of the physical computer system  
 416 shall be represented as elements of an ordered array property as defined in Table 11 from [DSP1012](#).

417 When an instance of CIM\_BootSourceSetting is instantiated and represented in the view class, the  
 418 CIM\_BootService instance should be associated with the CIM\_PhysicalComputerSystemView instance  
 419 via an instance of CIM\_ElementView where the Antecedent property is a reference to the  
 420 CIM\_BootService instance and the Dependent property is a reference to the  
 421 CIM\_PhysicalComputerSystemView instance.

422

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

CIM_PhysicalComputerSystemView property name	CIM_BootSourceSetting property name (origin)
PersistentBootConfigOrder[]	StructuredBootString
OneTimeBootSource	StructuredBootString

423 **7.1.2.9.1 CIM\_PhysicalComputerSystemView.OneTimeBootSource**

424 This property represents the boot source that is used for the next one-time boot of the physical computer  
 425 system. The value of this property is a StructuredBootString value. A value of empty string shall represent  
 426 that the one-time boot source is not configured.

427 **7.1.2.9.2 CIM\_PhysicalComputerSystemView.PersistentBootConfigOrder**

428 This property is an ordered array of elements representing the boot order of enabled boot sources of the  
 429 persistent default boot configuration on the physical computer system. The value of each element in this  
 430 array is a StructuredBootString value. An empty array shall represent that no persistent boot order is  
 431 specified. If the array is not empty, then each boot source specified in the array should support fail-  
 432 through. An unsuccessful boot attempt should result in continuing through the ordered array of boot  
 433 sources for which to attempt to boot.

434 **7.1.2.10 Representing system identity information**

435 **7.1.2.10.1 CIM\_PhysicalComputerSystemView.Dedicated**

436 The Dedicated property shall indicate the purpose(s) to which the physical computer system is dedicated,  
 437 if any, and what functionality is provided. See the specialized profiles of [DSP1052](#) for requirements (for  
 438 example DSP1004 and DSP1058).

### 439 7.1.2.10.2 CIM\_PhysicalComputerSystemView.OtherDedicatedDescriptions

440 The OtherDedicatedDescriptions property shall contain strings describing how or why the system is  
 441 dedicated when the Dedicated property includes “Other” (value=2). See the specialized profiles of  
 442 [DSP1052](#) for requirements (for example DSP1004 and DSP1058).

### 443 7.1.2.11 Representing system FRU information

#### 444 7.1.2.11.1 CIM\_PhysicalComputerSystemView.FRUInfoSupported

445 The FRUInfoSupported property shall correspond to the value of the  
 446 CIM\_PhysicalAssetCapabilities.FRUInfoSupported property value associated to the CIM\_Chassis  
 447 instance.

## 448 8 Methods

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

### 451 8.1 CIM\_PhysicalComputerSystemView.RequestStateChange( )

452 When the ImplementedFeatures property of the CIM\_RegisteredProfile instance contains  
 453 “DMTF:ComputerSystemView”, then RequestStateChange( ) shall be implemented.

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

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

458 No standard messages are defined.

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

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

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

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

464

465

466 **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 4 (Shut Down) 6 (Offline) 7 (Test) 8 (Defer) 9 (Quiesce) 10 (Reboot) 11 (Reset) – Power-Cycle
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

467 **8.2 CIM\_PhysicalComputerSystemView.ClearLog( )**

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

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

472 No standard messages are defined.

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

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

474 **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

475

476 **8.3 CIM\_PhysicalComputerSystemView.InstallSoftwareFromURI( )**

477 Invocation of the InstallSoftwareFromURI() method starts a job to install software from the designated  
478 URI to the physical computer system. A return code value of zero shall indicate that the installation of  
479 software was successfully initiated. Implementations shall determine based on the payload whether the  
480 installation is intended for BIOS or Management Firmware.

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

482 No standard messages are defined.

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

484 **Table 16 – CIM\_PhysicalComputerSystemView.InstallSoftwareFromURI( ) method: Return code**  
485 **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.

486 **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	InstallOptionValues	string	InstallOptionValues 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

#### 487 **8.4 CIM\_PhysicalComputerSystemView.ModifyPersistentBootConfigOrder( )**

488 Invocation of the ModifyPersistentBootConfigOrder( ) method modifies the persistent boot configuration  
489 order. A return code value of zero shall indicate that the new persistent boot configuration order was  
490 modified successfully.

491 Detailed requirements of the ModifyPersistentBootConfigOrder( ) method are specified in Table 18 and  
492 Table 19.

493 No standard messages are defined.

494 **Table 18 – CIM\_PhysicalComputerSystemView.ModifyPersistentBootConfigOrder( ) method:**  
495 **Return code values**

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



496 **Table 19 – CIM\_PhysicalComputerSystemView.ModifyPersistentBootConfigOrder() method:**  
 497 **Parameters**

Qualifiers	Name	Type	Description/Values
IN, REQ	StructuredBootString	string array	An array of StructuredBootString values
OUT	Job	CIM_ConcreteJob REF	Returned if job started

## 498 **8.5 CIM\_PhysicalComputerSystemView.SetOneTimeBootSource( )**

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

501 Detailed requirements of the SetOneTimeBootSource() method are specified in Table 20 and Table 21.

502 No standard messages are defined.

503 **Table 20 – CIM\_PhysicalComputerSystemView.SetOneTimeBootSource() method: Return code**  
 504 **values**

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

505 **Table 21 – 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

## 506 **8.6 Profile conventions for operations**

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

509 The default list of operations is as follows:

- 510 • GetInstance
- 511 • EnumerateInstances
- 512 • EnumerateInstanceNames
- 513 • Associators
- 514 • AssociatorNames
- 515 • References
- 516 • ReferenceNames

517 **8.7 CIM\_PhysicalComputerSystemView**

518 Table 22 lists operations that either have special requirements beyond those from [DSP0200](#) or shall not  
519 be supported.

520 **Table 22 – 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.

521 **8.8 CIM\_ElementView**

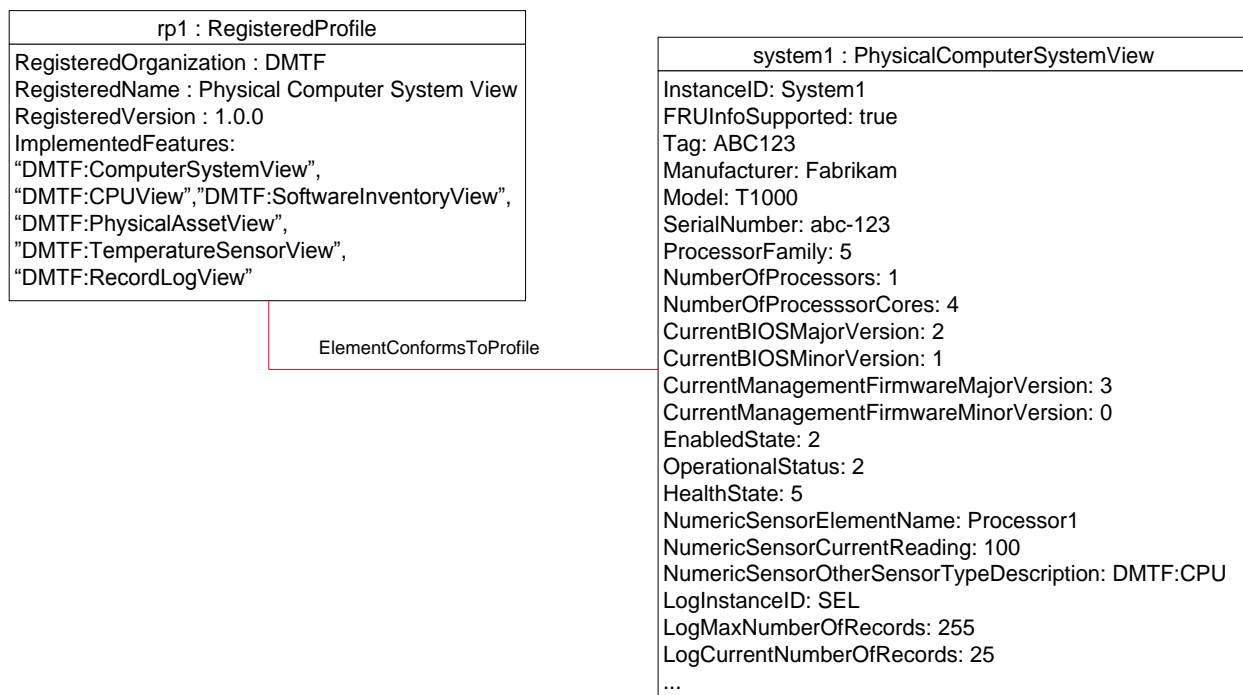
522 All operations in the default list in 8.2 shall be implemented as defined in [DSP0200](#).

523 **9 Use cases**

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

525 **9.1 Miscellaneous object diagrams**

526 The object diagram in Figure 2 shows one possible method for advertising profile conformance. The  
 527 instances of CIM\_RegisteredProfile are used to identify the version of the *Physical Computer System*  
 528 *View Profile* with which an instance of CIM\_PhysicalComputerSystemView and its associated instances  
 529 are conformant. An instance of CIM\_RegisteredProfile exists for each profile that is instrumented in the  
 530 system. An instance of CIM\_RegisteredProfile identifies the “DMTF Physical Computer System View  
 531 Profile version 1.0.0”. This diagram represents a simple managed computer system that only implements  
 532 the Profile Registration Profile and the Physical Computer System View Profile. The implementation of  
 533 Physical Computer System View Profile indicates that properties and methods related to Computer  
 534 System, CPU, Software Inventory, Physical Asset, Sensors, and Record Log are implemented as  
 535 advertised using the ImplementedFeatures property of the instance of CIM\_RegisteredProfile.

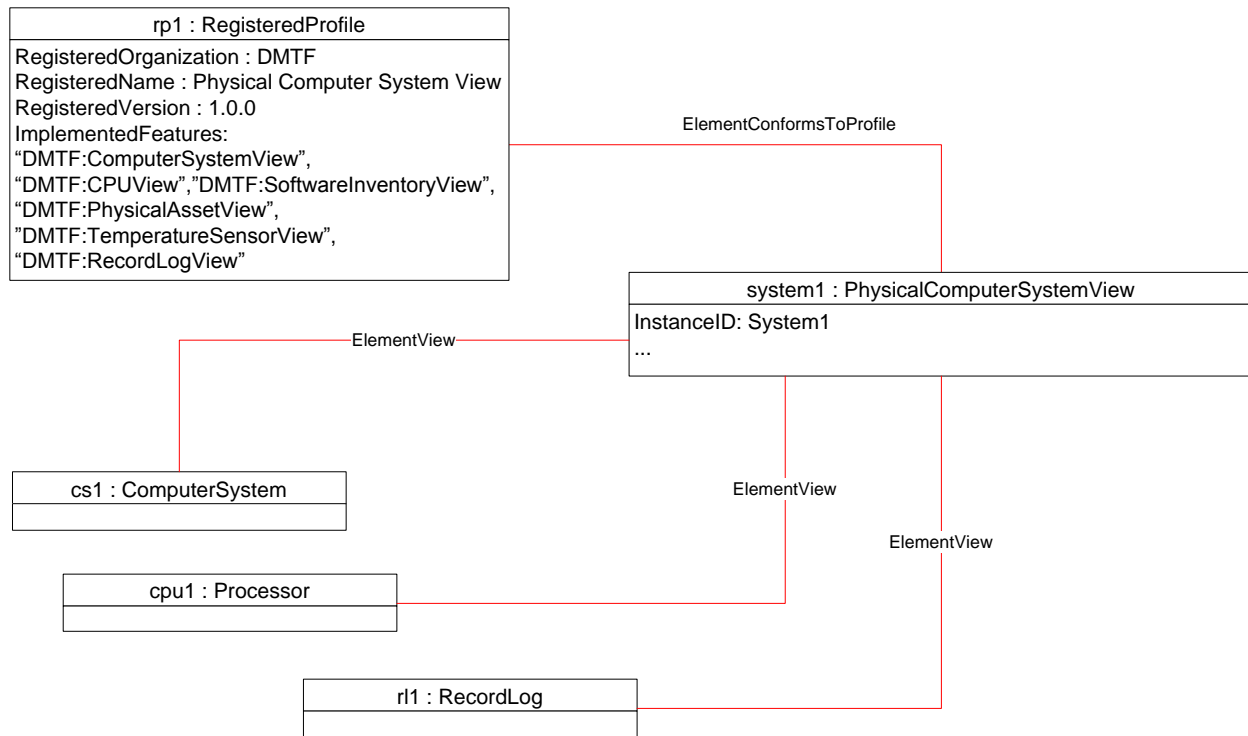


536

537 **Figure 2 – Simple PhysicalComputerSystemView implementation**

538 **9.2 Small footprint managed computer system with additional CIM Schema**  
 539 **implemented**

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



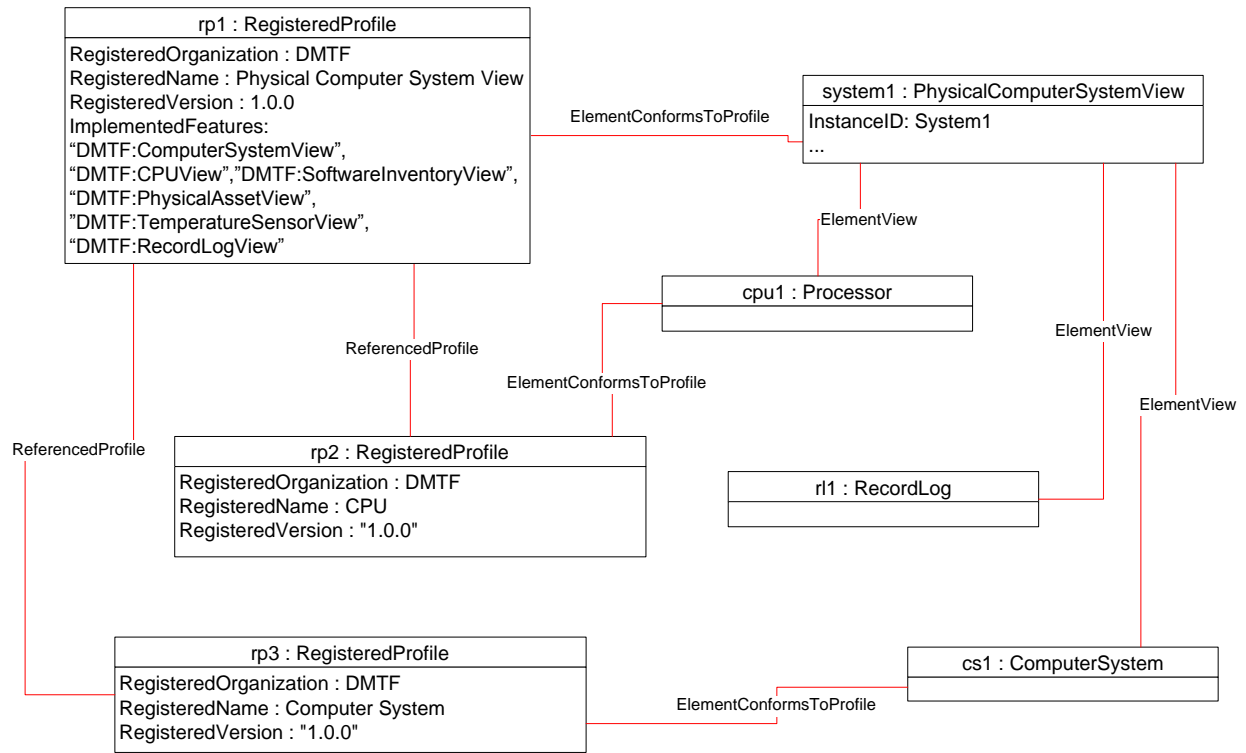
545  
546

547

**Figure 3 – Small footprint PhysicalComputerSystemView implementation**

### 548 **9.3 Large managed computer system with additional CIM Profiles implemented**

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



552

553

**Figure 4 – Rich PhysicalComputerSystemView implementation**

554

### 9.4 Managed system exposing multiple temperature sensors

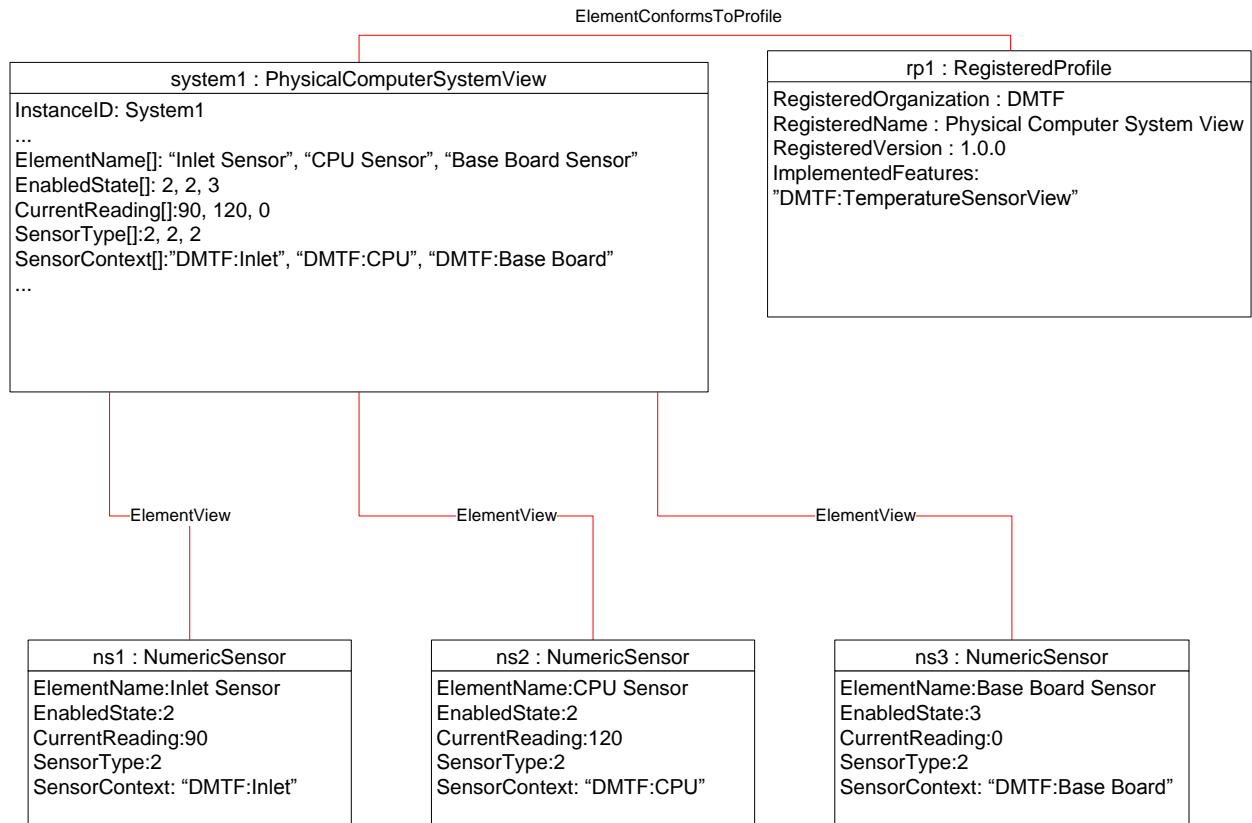
555

556

557

558

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.



559

560

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

## 561 **9.5 Determine the power state of the physical computer system**

562 To determine the power state:

- 563 1) Verify that the managed computer system implements the DMTF:ComputerSystemView feature  
564 by reading ImplementedFeatures property of the CIM\_RegisteredProfile instance.
- 565 2) Read the EnabledState property of the instance of CIM\_PhysicalComputerSystemView.
- 566 3) See Table 12 for the mapping of power states to the EnabledState property.

## 567 **9.6 Change the power state of the physical computer system**

568 To change the power state:

- 569 1) Verify that the managed computer system implements the DMTF:ComputerSystemView feature  
570 by reading the ImplementedFeatures property of the CIM\_RegisteredProfile instance.
- 571 2) Invoke the RequestStateChange() method of the instance of CIM\_PhysicalComputerSystemView  
572 providing the new requested state.
- 573 3) See Table 12 for the mapping of power states to the EnabledState property.

## 574 **9.7 Get properties of a specific record log of the physical computer system**

575 To get the properties of a specific record log:

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

## 581 **9.8 Browse the records of a log of the physical computer system**

582 To browse log entries:

- 583 1) Verify that the managed computer system implements the DMTF:RecordLogView feature by  
584 reading the ImplementedFeatures property of the CIM\_RegisteredProfile instance.
- 585 2) Verify that an instance of CIM\_RegisteredProfile for Record Log Profile exists.
- 586 3) Traverse the association to the CIM\_RecordLog instance with the LogInstanceID using the  
587 CIM\_ElementView association class.
- 588 4) Enumerate instances of CIM\_LogEntry that are associated through the CIM\_LogManagesRecord  
589 association to the given instance of CIM\_RecordLog.  
590

## 591 **9.9 Monitor temperate sensor readings of the physical computer system**

592 To monitor CPU temperature sensor readings:

- 593 1) Verify that the managed computer system implements the DMTF:TemperatureSensorView  
594 feature by reading the ImplementedFeatures property of the CIM\_RegisteredProfile instance.
- 595 2) Identify the index to a specific sensor that contains "DMTF:CPU" as the value for  
596 NumericSensorContext.
- 597 3) Use this index to read the ordered arrays NumericSensorBaseUnits, NumericSensorUnitModifier,  
598 NumericSensorRateUnits, and NumericSensorCurrentReading to compute the reading value for  
599 that specific sensor.

## 600 **9.10 Modify persisted boot order of the physical computer system**

601 To modify the persisted boot order:

- 602 1) Verify that the managed computer system implements the DMTF:BootControlView feature by  
603 reading the ImplementedFeatures property of the CIM\_RegisteredProfile instance.
- 604 2) Invoke the ModifyPersistentBootOrder() method of the instance of  
605 CIM\_PhysicalComputerSystemView to reflect the desired boot order based on index values of  
606 StructedBootString array.
- 607 3) To exclude boot sources from the boot order exclude indexes for those boot sources from the  
608 StructuredBootStringIndex parameter array value.

## 609 **9.11 Configure a source of the physical computer system for next reboot only**

610 To configure the boot source for next reboot only:

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

## 616 **9.12 Update the BIOS firmware of the physical computer system**

617 To update the BIOS firmware:

- 618 1) Verify that the managed computer system implements the DMTF:SoftwareUpdateView feature by  
619 reading the ImplementedFeatures property of the CIM\_RegisteredProfile instance.
- 620 2) Invoke the InstallSoftwareFromURI( ) method of the instance of  
621 CIM\_PhysicalComputerSystemView providing the classification value of 6 (Firmware/BIOS),  
622 location of the software and required options.  
623



624 **10 CIM Elements**

625 Table 23 shows the instances of CIM Elements for this profile. Instances of the CIM Elements shall be  
 626 implemented as described in Table 23.

627 **Table 23 – 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		

628 **10.1 CIM\_PhysicalComputerSystemView**

629 CIM\_PhysicalComputerSystemView describes a physical computer system and associated management  
 630 information in a managed computer system.

631 **Table 24 – 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	Conditional	See Table 2.
RequestedState	Conditional	See Table 2.
OperationalStatus	Conditional	See Table 2.
HealthState	Conditional	See Table 2.
Dedicated	Conditional	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.
PowerUtilizationModesSupported	Conditional	See Table 2.

Elements	Requirement	Description
PowerUtilizationMode	Conditional	See Table 2.
PowerAllocationLimit	Conditional	See Table 2.
NumericSensorElementName	Conditional	See Table 2.
NumericSensorEnabledState	Conditional	See Table 2.
NumericSensorHealthState	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.
RecordLogInstanceID	Conditional	See Table 2.
RecordLogMaxNumberOfRecords	Conditional	See Table 2.
RecordLogCurrentNumberOfRecords	Conditional	See Table 2.
RecordLogOverWritePolicy	Conditional	See Table 2.
RecordLogState	Conditional	See Table 2.
StructuredBootString	Conditional	See Table 2.
PersistentBootConfigOrder	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.
MemoryNumberOfBlocks	Conditional	See Table 2.
MemoryConsumableBlocks	Conditional	See Table 2.

Elements	Requirement	Description
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()	Optional	See 8.1.
ClearLog()	Optional	See 8.2.
InstallSoftwareFromURI()	Optional	See 8.3.
ModifyPersistentBootConfigOrder()	Optional	See 8.4.
SetOneTimeBootSource()	Optional	See 8.5.

632 **10.2 CIM\_RegisteredProfile**

633 CIM\_RegisteredProfile identifies the *Physical Computer System View Profile* in order for a client to  
 634 determine whether an instance of CIM\_PhysicalComputerSystemView is conformant with this profile. The  
 635 CIM\_RegisteredProfile class is defined by the [Profile Registration Profile](#). With the exception of the  
 636 mandatory values specified for the properties in Table 25 , the behavior of the CIM\_RegisteredProfile  
 637 instance is in accordance with the [Profile Registration Profile](#).

638 **Table 25 – 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	This property shall contain "DMTF:ComputerSystemView". This property may contain one or more of these values "DMTF:RecordLogView", "DMTF:TemperatureSensorView", "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.

### 639 10.3 CIM\_ElementView

640 CIM\_ElementView associates additional CIM Schema to the CIM\_PhysicalComputerSystemView  
641 instance

642 **Table 26 – 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.

### 643 10.4 CIM\_ElementConformsToProfile

644 CIM\_ElementConformsToProfile associates the instance of CIM\_RegisteredProfile to the  
645 CIM\_PhysicalComputerSystemView instance

646 **Table 27 – Class: CIM\_ElementView**

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.

647  
648  
649  
650

## **ANNEX A (informative)**

### **Change log**

<b>Version</b>	<b>Date</b>	<b>Description</b>
1.0.0a	2012-07-05	DMTF Work In Progress
1.0.0i	2012-11-12	DMTF Work In Progress Update

651  
652