



1

2

3

4

**Document Number: DSP1011**

**Date: 2016-05-17**

**Version: 1.0.3**

## 5 **Physical Asset Profile**

6 **Supersedes: 1.0.2**

7 **Document Class: Normative**

8 **Document Status: Published**

9 **Document Language: en-US**

## 10 Copyright Notice

11 Copyright © 2016 Distributed Management Task Force, Inc. (DMTF). All rights reserved.

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

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26 withdrawn or modified after publication, and shall be indemnified and held harmless by any party  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

29 For information about patents held by third-parties which have notified the DMTF that, in their opinion,  
30 such patent may relate to or impact implementations of DMTF standards, visit  
31 <http://www.dmtf.org/about/policies/disclosures.php>.

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

# CONTENTS

34	Foreword .....	7
35	Introduction.....	8
36	1 Scope .....	9
37	2 Normative references.....	9
38	2.1 Approved references.....	9
39	2.2 Other references.....	9
40	3 Terms and definitions.....	9
41	4 Symbols and abbreviated terms.....	11
42	5 Synopsis.....	12
43	6 Description .....	12
44	7 Implementation.....	14
45	7.1 Physical element.....	14
46	7.2 Finding the Scoping Instance of the CIM_System or CIM_ComputerSystem class .....	14
47	7.3 Modeling the physical aspects of logical representation of devices .....	15
48	7.4 Support for the Physical Element's FRU information .....	15
49	7.5 Compatibility of Physical Packages .....	16
50	7.6 Modeling System Chassis .....	16
51	7.7 Modeling configuration capacity .....	16
52	7.8 Modeling physical hierarchy .....	17
53	7.9 Modeling a physical memory .....	17
54	8 Methods.....	17
55	8.1 Profile conventions for operations .....	17
56	8.2 CIM_Card.....	17
57	8.3 CIM_Chassis.....	18
58	8.4 CIM_Chip .....	18
59	8.5 CIM_ComputerSystemPackage .....	18
60	8.6 CIM_ConfigurationCapacity .....	18
61	8.7 CIM_ConnectedTo.....	18
62	8.8 CIM_Container.....	19
63	8.9 CIM_ElementCapabilities .....	19
64	8.10 CIM_ElementCapacity .....	19
65	8.11 CIM_ElementInConnector .....	20
66	8.12 CIM_PhysicalAssetCapabilities .....	20
67	8.13 CIM_PhysicalComponent .....	20
68	8.14 CIM_PhysicalConnector .....	20
69	8.15 CIM_PhysicalFrame.....	20
70	8.16 CIM_PhysicalMemory .....	20
71	8.17 CIM_PhysicalPackage .....	20
72	8.18 CIM_Rack .....	20
73	8.19 CIM_Realizes.....	21
74	8.20 CIM_Slot .....	21
75	8.21 CIM_SystemPackaging.....	21
76	9 Use cases.....	22
77	9.1 System chassis FRU information.....	22
78	9.2 Fan package FRU information.....	22
79	9.3 Finding the Scoping Instance for a fan package .....	23
80	9.4 Physical topology and finding the Scoping Instance .....	24
81	9.5 Physical topology.....	26
82	9.6 Physical memory.....	27
83	9.7 Representing configuration capacity.....	28
84	9.8 Representing physical connector.....	30
85	9.9 Determining the part number of a failing component.....	30

86	9.10	Obtaining the physical inventory for all devices within a system .....	30
87	9.11	Obtaining the physical inventory for a System Chassis .....	30
88	9.12	Determining whether the slot is empty .....	31
89	9.13	Retrieving the fan capacity for the chassis .....	31
90	9.14	Retrieving the maximum capacity of the type of fan package within the chassis .....	31
91	10	CIM Elements .....	31
92	10.1	CIM_Card .....	32
93	10.2	CIM_Chassis .....	33
94	10.3	CIM_Chip .....	33
95	10.4	CIM_ComputerSystemPackage .....	34
96	10.5	CIM_ConfigurationCapacity .....	34
97	10.6	CIM_ConnectedTo .....	34
98	10.7	CIM_Container .....	35
99	10.8	CIM_ElementCapabilities .....	35
100	10.9	CIM_ElementCapacity .....	35
101	10.10	CIM_ElementInConnector .....	36
102	10.11	CIM_PhysicalAssetCapabilities .....	36
103	10.12	CIM_PhysicalComponent .....	36
104	10.13	CIM_PhysicalConnector .....	37
105	10.14	CIM_PhysicalFrame .....	37
106	10.15	CIM_PhysicalMemory .....	38
107	10.16	CIM_PhysicalPackage .....	38
108	10.17	CIM_Rack .....	39
109	10.18	CIM_Realizes .....	39
110	10.19	CIM_RegisteredProfile .....	40
111	10.20	CIM_Slot .....	40
112	10.21	CIM_SystemPackaging .....	41
113		ANNEX A (informative) Change log .....	42

114

## 115 Figures

116	Figure 1 – Physical Asset Profile: Profile class diagram .....	13
117	Figure 2 – System chassis object diagram .....	22
118	Figure 3 – CIM_PhysicalPackage object diagram .....	23
119	Figure 4 – Scoping Instance: Logical device object diagram .....	24
120	Figure 5 – Scoping Instance: Physical topology object diagram .....	25
121	Figure 6 – Physical Asset Profile: Topology object diagram .....	26
122	Figure 7 – Physical memory topology object diagram .....	27
123	Figure 8 – Configuration capacity object diagram .....	28
124	Figure 9 – Additional configuration capacity object diagram .....	29
125	Figure 10 – Network port connector object diagram .....	30

126

127 **Tables**

128 Table 1 – Referenced profiles ..... 12

129 Table 2 – Operations: CIM\_ComputerSystemPackage ..... 18

130 Table 3 – Operations: CIM\_ConnectedTo ..... 18

131 Table 4 – Operations: CIM\_Container ..... 19

132 Table 5 – Operations: CIM\_ElementCapabilities ..... 19

133 Table 6 – Operations: CIM\_ElementCapacity ..... 19

134 Table 7 – Operations: CIM\_ElementInConnector ..... 20

135 Table 8 – Operations: CIM\_Realizes ..... 21

136 Table 9 – Operations: CIM\_SystemPackaging ..... 21

137 Table 10 – CIM Elements: Physical Asset Profile ..... 31

138 Table 11 – Class: CIM\_Card ..... 32

139 Table 12 – Class: CIM\_Chassis ..... 33

140 Table 13 – Class: CIM\_Chip ..... 33

141 Table 14 – Class: CIM\_ComputerSystemPackage ..... 34

142 Table 15 – Class: CIM\_ConfigurationCapacity ..... 34

143 Table 16 – Class: CIM\_ConnectedTo ..... 34

144 Table 17 – Class: CIM\_Container ..... 35

145 Table 18 – Class: CIM\_ElementCapabilities ..... 35

146 Table 19 – Class: CIM\_ElementCapacity ..... 35

147 Table 20 – Class: CIM\_ElementInConnector ..... 36

148 Table 21 – Class: CIM\_PhysicalAssetCapabilities ..... 36

149 Table 22 – Class: CIM\_PhysicalComponent ..... 36

150 Table 23 – Class: CIM\_PhysicalConnector ..... 37

151 Table 24 – Class: CIM\_PhysicalFrame ..... 37

152 Table 25 – Class: CIM\_PhysicalMemory ..... 38

153 Table 26 – Class: CIM\_PhysicalPackage ..... 38

154 Table 27 – Class: CIM\_Rack ..... 39

155 Table 28 – Class: CIM\_Realizes ..... 39

156 Table 29 – Class: CIM\_RegisteredProfile ..... 40

157 Table 30 – Class: CIM\_Slot ..... 40

158 Table 31 – Class: CIM\_SystemPackaging ..... 41



160

## Foreword

161 The *Physical Asset Profile* (DSP1011) was prepared by the Physical Platform Profiles Working Group and  
162 Server Management Working Group.

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

## 165 Acknowledgments

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

### 167 Editors:

- 168 • Jon Hass – Dell Inc.
- 169 • Jeff Hilland – Hewlett Packard Enterprise
- 170 • Khachatur Papanyan – Dell Inc.
- 171 • Hemal Shah – Broadcom Corporation

### 172 Contributors:

- 173 • Jon Hass – Dell Inc.
- 174 • Jeff Hilland – Hewlett Packard Enterprise
- 175 • Arvind Kumar – Intel
- 176 • John Leung – Intel
- 177 • Jeff Lynch – IBM
- 178 • Aaron Merkin – IBM
- 179 • Khachatur Papanyan – Dell Inc.
- 180 • Hemal Shah – Broadcom Corporation
- 181 • Christina Shaw – Hewlett Packard Enterprise
- 182 • Perry Vincent – Intel

183

## Introduction

184 This document describes the physical aspects of the logical elements that the implementation is  
185 instantiating. Physical aspects include asset, inventory, and other descriptive physical information. Also  
186 included are descriptions of association classes that describe the relationship of physical elements and  
187 DMTF profile registration information. The information in this specification should be sufficient for a  
188 provider or consumer of this data to identify unambiguously the classes, properties, methods, and values  
189 that must be instantiated and manipulated to represent and manage classes representing physical  
190 elements of systems and subsystems modeled using the DMTF CIM core and extended model  
191 definitions.

192 The target audience for this specification is implementers who are writing CIM-based providers or  
193 consumers of management interfaces representing the component described in this document.



194

# Physical Asset Profile

## 195 1 Scope

196 The *Physical Asset Profile* extends the management capability of the referencing profiles by adding the  
197 capability to describe the physical aspects of logical elements that the implementation is instantiating. The  
198 profile also describes the relationship between the physical elements and the profile's registration for the  
199 schema implementation and version information.

## 200 2 Normative references

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

### 205 2.1 Approved references

206 DMTF DSP0004, *CIM Infrastructure Specification 2.3*,  
207 [http://www.dmtf.org/standards/published\\_documents/DSP0004\\_2.3.pdf](http://www.dmtf.org/standards/published_documents/DSP0004_2.3.pdf)

208 DMTF DSP0200, *CIM Operations over HTTP 1.2*,  
209 <http://www.dmtf.org/sites/default/files/standards/documents/DSP200.html>

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

212 DMTF DSP1013, *Fan Profile 1.0*,  
213 [http://www.dmtf.org/standards/published\\_documents/DSP1013\\_1.0.pdf](http://www.dmtf.org/standards/published_documents/DSP1013_1.0.pdf)

214 DMTF DSP1033, *Profile Registration Profile 1.0*,  
215 [http://www.dmtf.org/standards/published\\_documents/DSP1033\\_1.0.pdf](http://www.dmtf.org/standards/published_documents/DSP1033_1.0.pdf)

### 216 2.2 Other references

217 ISO/IEC Directives, Part 2, *Rules for the structure and drafting of International Standards*,  
218 <http://isotc.iso.org/livelink/livelink.exe?func=ll&objId=4230456&objAction=browse&sort=subtype>

## 219 3 Terms and definitions

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

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

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

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

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

235 **3.1**  
236 **can**  
237 used for statements of possibility and capability, whether material, physical, or causal

238 **3.2**  
239 **cannot**  
240 used for statements of possibility and capability, whether material, physical, or causal

241 **3.3**  
242 **conditional**  
243 indicates requirements to be followed strictly in order to conform to the document when the specified  
244 conditions are met

245 **3.4**  
246 **mandatory**  
247 indicates requirements to be followed strictly in order to conform to the document and from which no  
248 deviation is permitted

249 **3.5**  
250 **may**  
251 indicates a course of action permissible within the limits of the document

252 **3.6**  
253 **need not**  
254 indicates a course of action permissible within the limits of the document

255 **3.7**  
256 **optional**  
257 indicates a course of action permissible within the limits of the document

258 **3.8**  
259 **referencing profile**  
260 indicates a profile that owns the definition of this class and can include a reference to this profile in its  
261 "Referenced Profiles" table

262 **3.9**  
263 **shall**  
264 indicates requirements to be followed strictly in order to conform to the document and from which no  
265 deviation is permitted

266 **3.10**  
267 **shall not**  
268 indicates requirements to be followed in order to conform to the document and from which no deviation is  
269 permitted

- 270 **3.11**  
271 **should**  
272 indicates that among several possibilities, one is recommended as particularly suitable, without  
273 mentioning or excluding others, or that a certain course of action is preferred but not necessarily required
- 274 **3.12**  
275 **should not**  
276 indicates that a certain possibility or course of action is deprecated but not prohibited
- 277 **3.13**  
278 **unspecified**  
279 indicates that this profile does not define any constraints for the referenced CIM element or operation
- 280 **3.14**  
281 **Delimited Substring**  
282 a substring element of the VendorCompatibilityStrings property of a Physical Element or an instance of  
283 CIM\_ConfigurationCapacity. The substring starts at the beginning of the string (representing an element  
284 in the array of the VendorCompatibilityStrings property) and terminates at the end of the string, or at a  
285 character that precedes a colon (:).
- 286 **3.15**  
287 **Physical Element**  
288 an instance of a CIM\_PhysicalElement subclass (such as CIM\_PhysicalConnector, CIM\_Slot,  
289 CIM\_PhysicalComponent, CIM\_Chip, CIM\_PhysicalMemory, CIM\_PhysicalPackage,  
290 CIM\_PhysicalFrame, CIM\_Chassis, CIM\_Rack, and CIM\_Card) that represents a physical element
- 291 **3.16**  
292 **Physical Package**  
293 an instance of a CIM\_PhysicalPackage or a CIM\_PhysicalPackage subclass (such as  
294 CIM\_PhysicalFrame, CIM\_Chassis, CIM\_Rack, and CIM\_Card) or CIM\_PhysicalComponent or  
295 CIM\_PhysicalComponent subclass (such as CIM\_Chip or CIM\_PhysicalMemory) that represents a  
296 package
- 297 **3.17**  
298 **System Chassis**  
299 an instance of the CIM\_PhysicalElement or CIM\_Chassis that is associated to an instance of  
300 CIM\_System or CIM\_ComputerSystem through the CIM\_SystemPackaging or  
301 CIM\_ComputerSystemPackage association, representing the physical package of the managed system.

## 302 **4 Symbols and abbreviated terms**

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

- 305 **4.1**  
306 **CIM**  
307 Common Information Model
- 308 **4.2**  
309 **FRU**  
310 Field Replaceable Unit

## 311 5 Synopsis

312 **Profile Name:** Physical Asset

313 **Version:** 1.0.3

314 **Organization:** DMTF

315 **CIM Schema version:** 2.22

316 **Central Class:** CIM\_PhysicalElement

317 **Scoping Class:** CIM\_ManagedSystemElement

318 The *Physical Asset Profile* extends the management capability of the referencing profiles by adding the  
319 capability to describe the physical aspects of the logical elements that the implementation is instantiating.  
320 Physical aspects include asset, inventory, and other descriptive physical information.

321 **Table 1 – Referenced profiles**

Profile Name	Organization	Version	Relationship	Behavior
Profile Registration	DMTF	1.0	Mandatory	

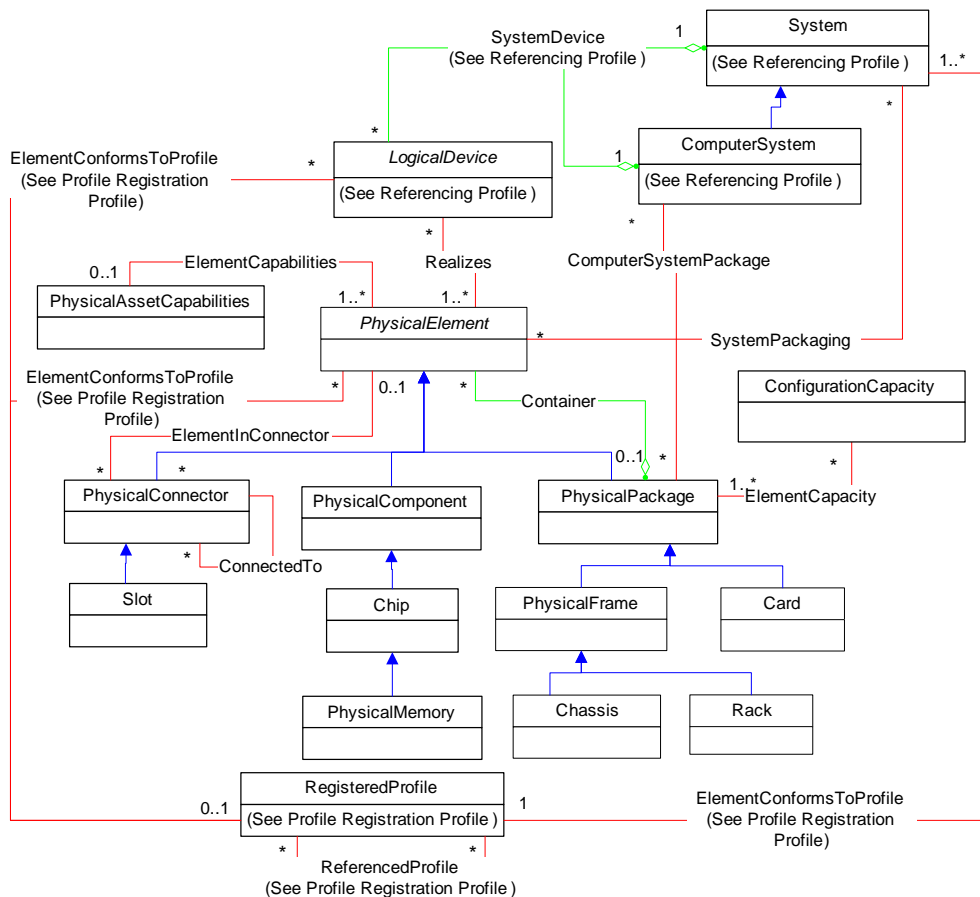
322 The Central Instance for the *Physical Asset Profile* shall be the instance of the CIM\_PhysicalElement  
323 subclass.

324 The Scoping Instance for the *Physical Asset Profile* shall be the instance of the  
325 CIM\_ManagedSystemElement. Note that this may include the subclass of CIM\_System, the  
326 CIM\_ComputerSystem class. The Scoping Instance is determined using the algorithm described in clause  
327 7.2.

## 328 6 Description

329 The *Physical Asset Profile* describes the necessary elements needed to provide the descriptive and asset  
330 information about the physical components in a managed domain and their topology. The profile does not  
331 cover the geographic location of the physical assets.

332 Figure 1 shows the CIM classes that are used in this profile. (For simplicity, the prefix CIM\_ has been  
333 removed from the names of the classes.) A Physical Element (see clause 3.15) describes the physical  
334 properties, including the FRU information, of a managed element. The capabilities of the Physical  
335 Elements are described by the properties of the CIM\_PhysicalAssetCapabilities class. The Physical  
336 Elements could be associated to the logical representation of the managed element through the  
337 CIM\_Realizes association. The enclosures or chassis of the managed systems are represented by a  
338 CIM\_PhysicalElement or CIM\_Chassis instance that is associated to the  
339 CIM\_System/CIM\_ComputerSystem instance through the  
340 CIM\_SystemPackaging/CIM\_ComputerSystemPackage association and are referred to as a System  
341 Chassis (see clause 3.17). Configuration capacity of the System Chassis is also represented within this  
342 profile by CIM\_ConfigurationCapacity instances.



343

344

**Figure 1 – Physical Asset Profile: Profile class diagram**

345 Physical Elements can be also arranged in a topology. The CIM\_Container, CIM\_ConnectedTo, and  
 346 CIM\_ElementInConnector associations are used to associate the Physical Elements and create the  
 347 physical topology of the managed elements.

348 Figure 1 also represents the ecosystem of *Physical Asset Profile* classes, illustrating their relationship  
 349 with classes of referencing profiles. The referencing profiles can identify the subclass of  
 350 CIM\_PhysicalElement to be used for representing the physical aspects of the managed element. For  
 351 example, the referencing profiles that contain a CIM\_LogicalDevice subclass can restrict the associated  
 352 subclass of CIM\_PhysicalPackage to CIM\_PhysicalMemory for instantiation of the *Physical Asset Profile*.  
 353 Such restrictions will be described in the referencing profiles.

354 The *Physical Asset Profile* is advertised through the CIM\_RegisteredProfile instance.

355 The *Physical Asset Profile* can be instantiated to represent a combination of the following scenarios:

- 356 • the physical aspects of a managed system, such as the FRU information for the chassis (see  
 357 clause 7.6)
- 358 • the physical aspects of a specific managed element, such as the FRU information of a fan (see  
 359 clause 7.3)

- 360 • the physical hierarchy of a managed system, such as the relationship between chassis, slots, and  
361 packages (see clause 7.8)
- 362 • the configuration capacity of a managed element, such as the minimum and maximum number of  
363 certain types of packages that the managed system can handle (see clause 7.7)

## 364 7 Implementation

365 This clause details the requirements related to the arrangement of instances and their properties for  
366 implementations of this profile.

### 367 7.1 Physical element

368 The implementation shall instantiate at least one instance of the subclass of CIM\_PhysicalElement  
369 (Physical Element). Referencing profiles may state the subclass of CIM\_PhysicalElement that is to be  
370 instantiated as part of the *Physical Asset Profile*.

371 At least one instance of CIM\_Realizes, CIM\_ComputerSystemPackage, or CIM\_SystemPackaging  
372 association class shall reference an instance of a subclass of CIM\_PhysicalElement (Physical Element).

373 Every Physical Element shall be referenced by at least one of the following properties:  
374 CIM\_ComputerSystemPackage.Antecedent, CIM\_SystemPackaging.Antecedent,  
375 CIM\_Realizes.Antecedent, CIM\_Container.PartComponent, or CIM\_ElementInConnector.Dependent.

### 376 7.2 Finding the Scoping Instance of the CIM\_System or CIM\_ComputerSystem 377 class

378 The following algorithm shall be used for locating the Scoping Instance of the CIM\_System or  
379 CIM\_ComputerSystem class starting from any selected Physical Element.

- 380 I. If the selected instance is of a Physical Package, proceed as follows:
  - 381 A. If the Physical Package is associated to the CIM\_LogicalDevice through the CIM\_Realizes  
382 association or to the CIM\_System/CIM\_ComputerSystem through the  
383 CIM\_SystemPackaging/CIM\_ComputerSystemPackage association, the Scoping Instance of  
384 the *Physical Asset Profile* shall be either of the following:
    - 385 a. the Scoping Instance of the CIM\_LogicalDevice instance that is associated to the  
386 Physical Package through the instance of CIM\_Realizes
    - 387 b. the Scoping Instance of CIM\_System/CIM\_ComputerSystem instance that is  
388 associated to the Physical Package through the instance of CIM\_SystemPackaging or  
389 CIM\_ComputerSystemPackage
  - 390 B. If the Physical Package is not associated to the CIM\_LogicalDevice through the CIM\_Realizes  
391 association or to the CIM\_System or CIM\_ComputerSystem through the  
392 CIM\_SystemPackaging or CIM\_ComputerSystemPackage association, proceed as follows:
    - 393 1. If the Physical Package is the Dependent or PartComponent reference in  
394 CIM\_ElementInConnector or CIM\_Container associations, respectively, choose one of the  
395 following paths:
      - 396 a. If the Antecedent or GroupComponent reference of the association is a Physical  
397 Package, select the Antecedent or GroupComponent referenced instance, and go to  
398 I.A.
      - 399 b. Else if the Antecedent or GroupComponent reference of the association is a Physical  
400 Element:

- 401 (1) If the Physical Element is associated to the CIM\_LogicalDevice through the  
 402 CIM\_Realizes association, the Scoping Instance of the *Physical Asset Profile*  
 403 shall be the Scoping Instance of the CIM\_LogicalDevice instance.
- 404 (2) If the Physical Element instance is not associated to the CIM\_LogicalDevice  
 405 through the CIM\_Realizes association:
- 406 (a) If the Physical Element is the PartComponent reference in the  
 407 CIM\_Container association:
- 408 1) If a Physical Package is the GroupComponent reference for the  
 409 CIM\_Container association, select the GroupComponent referenced  
 410 instance, and go to I.A.
- 411 2) If a Physical Element is the GroupComponent or Antecedent reference,  
 412 go to I.B.1.b(1).
- 413 (b) If the Physical Element is not the PartComponent or Dependent reference in  
 414 a CIM\_Container association, the Scoping Instance shall be the Central  
 415 Instance; thus, the Central Instance is associated to the  
 416 CIM\_RegisteredProfile instance.
- 417 2. Else the Scoping Instance shall be the Central Instance, thus, the Central Instance is  
 418 associated to the CIM\_RegisteredProfile instance.
- 419 II. If the instance is not a Physical Package, go to I.B.1.b(1).

### 420 7.3 Modeling the physical aspects of logical representation of devices

421 The implementation may implement the physical aspects of a managed device through instantiation of a  
 422 Physical Element.

423 When the physical aspects of the logical device are implemented, the CIM\_LogicalDevice subclass  
 424 instance, which represents the logical device, shall be associated with the Physical Element, which  
 425 represents the physical aspects of the logical device, through the CIM\_Realizes association.

### 426 7.4 Support for the Physical Element's FRU information

427 The Physical Element's support of FRU information shall be advertised by a  
 428 CIM\_PhysicalAssetCapabilities instance associated with the Physical Element. At most, one instance of  
 429 CIM\_PhysicalAssetCapabilities shall be associated with the Physical Element through the  
 430 CIM\_ElementCapabilities association.

431 When no CIM\_PhysicalAssetCapabilities instance is associated to the Physical Element, the Physical  
 432 Element's FRU information may not be supported.

433 When a CIM\_PhysicalAssetCapabilities instance is associated to the Physical Element and the  
 434 CIM\_PhysicalAssetCapabilities.FRUInfoSupported has a value of TRUE, the Physical Element's FRU  
 435 information shall be supported.

436 When FRU information is supported, the implementation shall populate the properties of the Physical  
 437 Element below with non-null, non-blank values. At least one of these properties shall be non-null, non-  
 438 blank without any whitespace characters (e.g., pattern "[^WSP]+"). Any property stated below shall not be  
 439 populated with a string containing only whitespace characters. If the SKU property is non-null, it shall be  
 440 used to convey the FRU number. Some combination of the properties below should be used for  
 441 replacement part information.

- 442 • Manufacturer
- 443 • Model
- 444 • PartNumber

- 445       • SerialNumber
- 446       • SKU

## 447   **7.5   Compatibility of Physical Packages**

448   When the Physical Package is instantiated, the implementation may represent the compatibility of the  
449   Physical Package. In that case, the conditions and requirements in this clause shall apply.

450   The compatibility between the physical packages, which are represented by Physical Packages, and  
451   slots, which are represented by CIM\_Slot instances, shall be advertised through the  
452   VendorCompatibilityStrings property.

453   The VendorCompatibilityStrings property of a Physical Package and an instance of CIM\_Slot shall be an  
454   array of strings, each uniquely identifying the specific type of package and matching a “:” character-free,  
455   non-zero length string, delimited by “:” character (pattern “[^:]+(:[^\:]+)”).

456   Only if the physical package represented by the Physical Package can be inserted into the slot  
457   represented by the instance of CIM\_Slot, the VendorCompatibilityStrings property of Physical Package  
458   shall contain an element with a Delimited Substring equal to a string of one of the elements from the  
459   VendorCompatibilityStrings property of an instance of CIM\_Slot.

## 460   **7.6   Modeling System Chassis**

461   The implementation may instantiate a System Chassis. When a System Chassis is instantiated, the  
462   System Chassis shall be associated with the instance of CIM\_System through the instance of  
463   CIM\_SystemPackaging, or with the instance of CIM\_ComputerSystem through the instance of  
464   CIM\_ComputerSystemPackage.

## 465   **7.7   Modeling configuration capacity**

466   The implementation may advertise the configuration capacity of the physical packages within the chassis,  
467   including the chassis itself. The configuration capacity shall be represented through the  
468   CIM\_ConfigurationCapacity instances.

469   When a System Chassis is present, the instrumentation shall associate all the instances of  
470   CIM\_ConfigurationCapacity to the System Chassis through the instances of CIM\_ElementCapacity.  
471   Additionally, when the configuration capacity is for a particular physical package represented by a  
472   Physical Package, the instrumentation may associate the Physical Package with the  
473   CIM\_ConfigurationCapacity through an instance of CIM\_ElementCapacity.

474   When instances of CIM Slot are instantiated, for each unique value of the  
475   CIM\_Slot.VendorCompatibilityStrings, an instance of CIM\_ConfigurationCapacity with an equal value for  
476   the CIM\_ConfigurationCapacity.VendorCompatibilityStrings property shall exist. Additional instances of  
477   CIM\_ConfigurationCapacity may exist.

478   When CIM\_Slot instances are not instantiated or the CIM\_Slot.VendorCompatibilityStrings property is not  
479   instrumented, the CIM\_ConfigurationCapacity.VendorCompatibilityStrings array property shall contain an  
480   element with a Delimited Substring that is equal to a string of one of the elements from the  
481   VendorCompatibilityStrings array property of a Physical Package that can be part of the configuration.



## 482 7.8 Modeling physical hierarchy

483 The physical hierarchy is represented by relationship and containment of Physical Elements. The  
484 implementation may represent the physical hierarchy as follows:

- 485 • When a physical element resides within a package, the Physical Element shall be associated  
486 with the Physical Package through the CIM\_Container association.
- 487 • When a package is plugged or connected to a slot or connector, the Physical Package shall be  
488 associated with the CIM\_PhysicalConnector or CIM\_Slot instance through the  
489 CIM\_ElementInConnector association.
- 490 • When physical connectors or slots are connected, the CIM\_PhysicalConnector or CIM\_Slot  
491 instances shall be associated through the CIM\_ConnectedTo association.

## 492 7.9 Modeling a physical memory

493 The implementation may implement the physical aspects of a memory inside the system through  
494 instantiation of the CIM\_PhysicalMemory class.

495 When a physical memory is modeled as an instance of CIM\_PhysicalMemory, the  
496 CIM\_PhysicalMemory.Speed property represents the speed of the physical memory in nanoseconds. The  
497 following requirements apply for CIM\_PhysicalMemory.Speed:

- 498 • If the speed of the physical memory is less than one nanosecond or unknown, then the  
499 CIM\_PhysicalMemory.Speed property shall be set to 0.
- 500 • If the speed of the physical memory is variable, then the CIM\_PhysicalMemory.Speed property  
501 shall be set to  $2^{32}-1$  (nanoseconds).

## 502 8 Methods

503 This clause details the requirements for supporting intrinsic operations for the CIM elements defined by  
504 this profile. The *Physical Asset Profile* does not define any extrinsic methods.

### 505 8.1 Profile conventions for operations

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

508 The default list of operations is as follows:

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

### 516 8.2 CIM\_Card

517 All operations in the default list in 8.1 shall be implemented as defined in [DSP0200](#).

518 NOTE Related profiles may define additional requirements on operations for the profile class.

519 **8.3 CIM\_Chassis**520 All operations in the default list in 8.1 shall be implemented as defined in [DSP0200](#).

521 NOTE Related profiles may define additional requirements on operations for the profile class.

522 **8.4 CIM\_Chip**523 All operations in the default list in 8.1 shall be implemented as defined in [DSP0200](#).

524 NOTE Related profiles may define additional requirements on operations for the profile class.

525 **8.5 CIM\_ComputerSystemPackage**526 Table 2 lists implementation requirements for operations. If implemented, these operations shall be  
527 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 2, all operations in  
528 the default list in 8.1 shall be implemented as defined in [DSP0200](#).

529 NOTE Related profiles may define additional requirements on operations for the profile class.

530

**Table 2 – Operations: CIM\_ComputerSystemPackage**

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

531 **8.6 CIM\_ConfigurationCapacity**532 All operations in the default list in 8.1 shall be implemented as defined in [DSP0200](#).

533 NOTE Related profiles may define additional requirements on operations for the profile class.

534 **8.7 CIM\_ConnectedTo**535 Table 3 lists implementation requirements for operations. If implemented, these operations shall be  
536 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 3, all operations in  
537 the default list in 8.1 shall be implemented as defined in [DSP0200](#).

538 NOTE Related profiles may define additional requirements on operations for the profile class.

539

**Table 3 – Operations: CIM\_ConnectedTo**

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

540 **8.8 CIM\_Container**

541 Table 4 lists implementation requirements for operations. If implemented, these operations shall be  
 542 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 4, all operations in  
 543 the default list in 8.1 shall be implemented as defined in [DSP0200](#).

544 NOTE Related profiles may define additional requirements on operations for the profile class.

545 **Table 4 – Operations: CIM\_Container**

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

546 **8.9 CIM\_ElementCapabilities**

547 Table 5 lists implementation requirements for operations. If implemented, these operations shall be  
 548 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 5, all operations in  
 549 the default list in 8.1 shall be implemented as defined in [DSP0200](#).

550 NOTE Related profiles may define additional requirements on operations for the profile class.

551 **Table 5 – Operations: CIM\_ElementCapabilities**

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

552 **8.10 CIM\_ElementCapacity**

553 Table 6 lists implementation requirements for operations. If implemented, these operations shall be  
 554 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 6, all operations in  
 555 the default list in 8.1 shall be implemented as defined in [DSP0200](#).

556 NOTE Related profiles may define additional requirements on operations for the profile class.

557 **Table 6 – Operations: CIM\_ElementCapacity**

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

## 558 8.11 CIM\_ElementInConnector

559 Table 7 lists implementation requirements for operations. If implemented, these operations shall be  
 560 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 7, all operations in  
 561 the default list in 8.1 shall be implemented as defined in [DSP0200](#).

562 NOTE Related profiles may define additional requirements on operations for the profile class.

563 **Table 7 – Operations: CIM\_ElementInConnector**

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

## 564 8.12 CIM\_PhysicalAssetCapabilities

565 All operations in the default list in 8.1 shall be implemented as defined in [DSP0200](#).

566 NOTE Related profiles may define additional requirements on operations for the profile class.

## 567 8.13 CIM\_PhysicalComponent

568 All operations in the default list in 8.1 shall be implemented as defined in [DSP0200](#).

569 NOTE Related profiles may define additional requirements on operations for the profile class.

## 570 8.14 CIM\_PhysicalConnector

571 All operations in the default list in 8.1 shall be implemented as defined in [DSP0200](#).

572 NOTE Related profiles may define additional requirements on operations for the profile class.

## 573 8.15 CIM\_PhysicalFrame

574 All operations in the default list in 8.1 shall be implemented as defined in [DSP0200](#).

575 NOTE Related profiles may define additional requirements on operations for the profile class.

## 576 8.16 CIM\_PhysicalMemory

577 All operations in the default list in 8.1 shall be implemented as defined in [DSP0200](#).

578 NOTE Related profiles may define additional requirements on operations for the profile class.

## 579 8.17 CIM\_PhysicalPackage

580 All operations in the default list in 8.1 shall be implemented as defined in [DSP0200](#).

581 NOTE Related profiles may define additional requirements on operations for the profile class.

## 582 8.18 CIM\_Rack

583 All operations in the default list in 8.1 shall be implemented as defined in [DSP0200](#).

584 NOTE Related profiles may define additional requirements on operations for the profile class.

585 **8.19 CIM\_Realizes**

586 Table 8 lists implementation requirements for operations. If implemented, these operations shall be  
 587 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 8, all operations in  
 588 the default list in 8.1 shall be implemented as defined in [DSP0200](#).

589 NOTE Related profiles may define additional requirements on operations for the profile class.

590 **Table 8 – Operations: CIM\_Realizes**

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

591 **8.20 CIM\_Slot**

592 All operations in the default list in 8.1 shall be implemented as defined in [DSP0200](#).

593 NOTE Related profiles may define additional requirements on operations for the profile class.

594 **8.21 CIM\_SystemPackaging**

595 Table 9 lists implementation requirements for operations. If implemented, these operations shall be  
 596 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 9, all operations in  
 597 the default list in 8.1 shall be implemented as defined in [DSP0200](#).

598 NOTE Related profiles may define additional requirements on operations for the profile class.

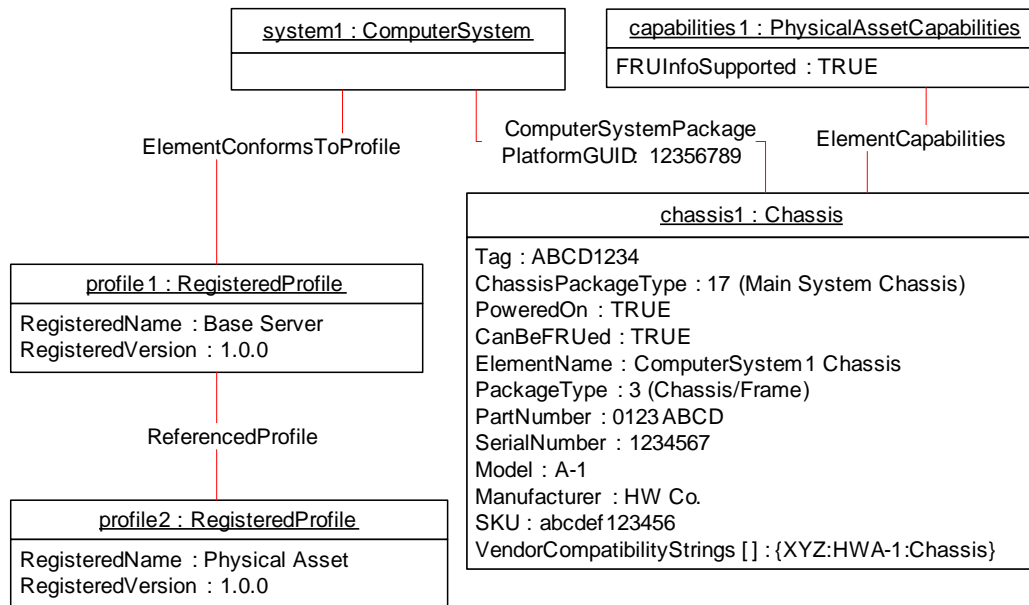
599 **Table 9 – Operations: CIM\_SystemPackaging**

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

## 600 9 Use cases

### 601 9.1 System chassis FRU information

602 Figure 2 represents a possible instantiation of the *Physical Asset Profile*. In this case, the physical  
 603 aspects of the instance of CIM\_ComputerSystem are represented by an instance of CIM\_Chassis  
 604 through a CIM\_ComputerSystemPackage association. The Tag property of Chassis1 represents the  
 605 asset tag of the chassis. The TRUE value of the FRUInfoSupported property of capabilities1 indicates  
 606 that chassis1 contains non-zero, non-blank properties describing FRU information such as PartNumber,  
 607 SerialNumber, Model, and Manufacturer. (See clause 7.4 for more details.) Profile2 advertises the  
 608 implemented *Physical Asset Profile* information.



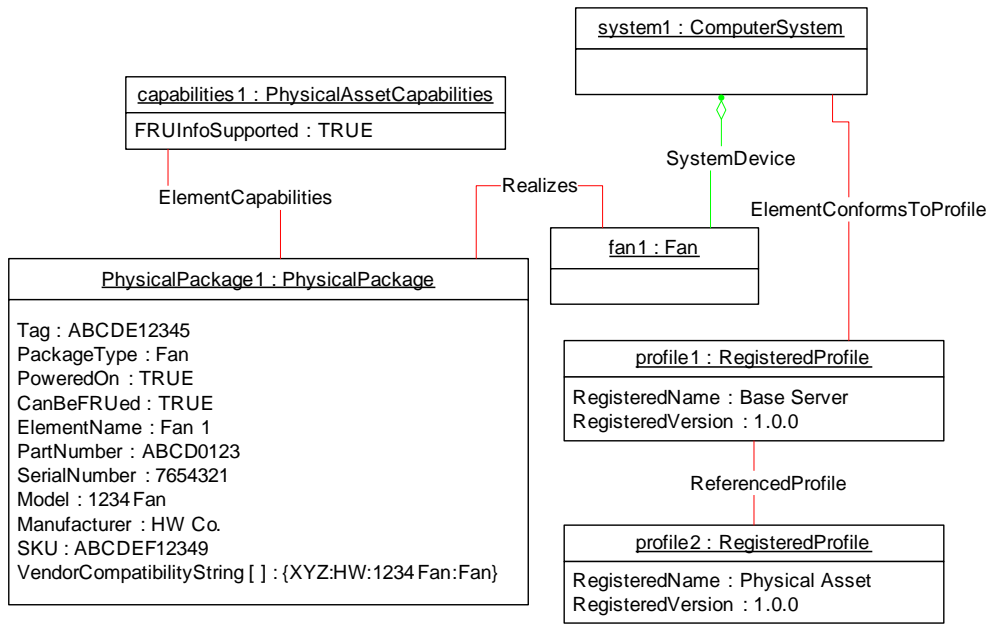
609

610

Figure 2 – System chassis object diagram

### 611 9.2 Fan package FRU information

612 Figure 3 represents another possible instantiation of the *Physical Asset Profile*. The instance of  
 613 CIM\_PhysicalPackage represents the physical properties of the given instance of CIM\_Fan through a  
 614 CIM\_Realizes association. The CIM\_PhysicalPackage.Tag property represents the asset tag of the fan1.  
 615 The TRUE value of the FRUInfoSupported property of capabilities1 indicates that physicalpackage1  
 616 contains non-zero, non-blank properties describing FRU information such as PartNumber, SerialNumber,  
 617 Model, Manufacturer, and SKU. (See clause 7.4 for more details.)



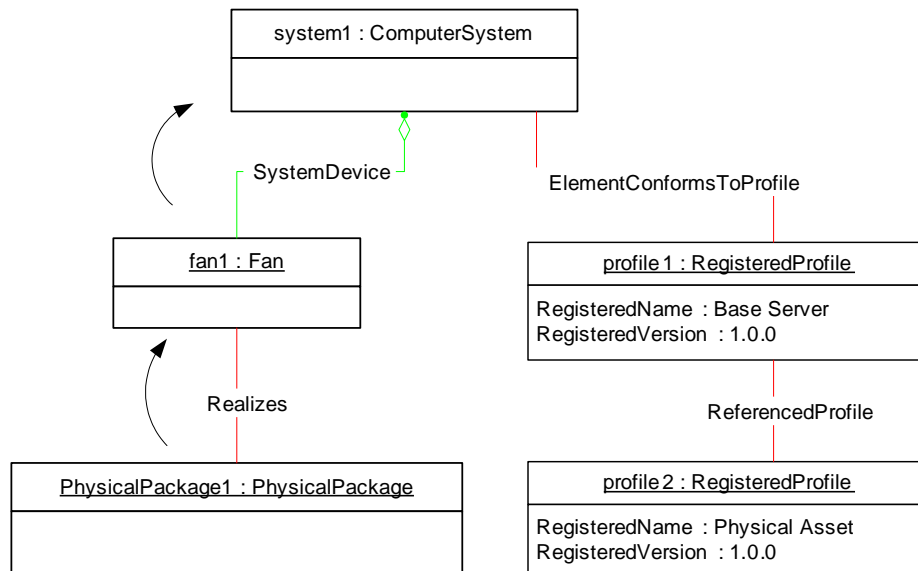
618

619

Figure 3 – CIM\_PhysicalPackage object diagram

620 **9.3 Finding the Scoping Instance for a fan package**

621 Figure 4 represents another possible instantiation of *Physical Asset Profile*. To find the Scoping Instance  
 622 of PhysicalPackage1, the client needs to select the fan1 associated through the CIM\_Realizes  
 623 association and then find the Scoping Instance for fan1. As defined in the [Fan Profile](#), the Scoping  
 624 Instance of fan1 is the CIM\_ComputerSystem instance associated to fan1 through the  
 625 CIM\_SystemDevice association: system1. Thus, system1 is the Scoping Instance of PhysicalPackage1.  
 626 By traversing through the CIM\_ElementConformsToProfile and subsequently the CIM\_ReferencedProfile  
 627 association, the client can find profile2, which advertises the *Physical Asset Profile* information.



628

629

**Figure 4 – Scoping Instance: Logical device object diagram**

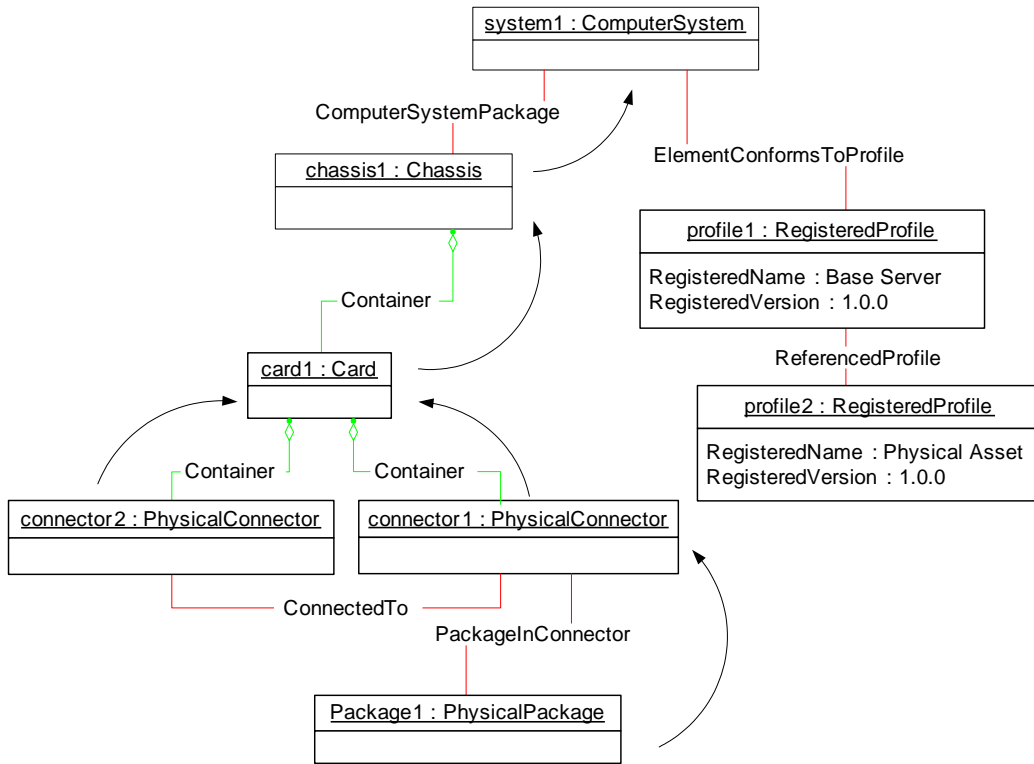
630

#### 9.4 Physical topology and finding the Scoping Instance

631 Figure 5 represents another possible instantiation of the *Physical Asset Profile*. To find the Scoping  
 632 Instance of package1, because package1 is referenced by the CIM\_ElementInConnector.Dependent  
 633 property, the client needs to select connector1, which is referenced by the  
 634 CIM\_ElementInConnector.Antecedent property. Then, because connector1 is referenced by the  
 635 CIM\_Container.PartComponent property, the client needs to select card1, which is referenced by the  
 636 CIM\_Container.GroupComponent. Then, because card1 is referenced by the  
 637 CIM\_Container.PartComponent property, the client needs to select chassis1, which is referenced by the  
 638 CIM\_Container.GroupComponent. Then, because chassis1 is associated to system1 through the  
 639 CIM\_ComputerSystemPackage association, system1 is the Scoping Instance of package1. The client can  
 640 traverse through the CIM\_ElementConformsToProfile and, subsequently, the CIM\_ReferencedProfile  
 641 association, to find profile2, which advertises the *Physical Asset Profile* information.

642 **NOTE** To enable finding the Scoping Instance of connector2, the implementation has instantiated an instance of  
 643 CIM\_Container that references card1 and connector2. Merely instantiating the instance of  
 644 CIM\_ConnectedTo referencing connector2 will not conform to the algorithm described in clause 7.2.





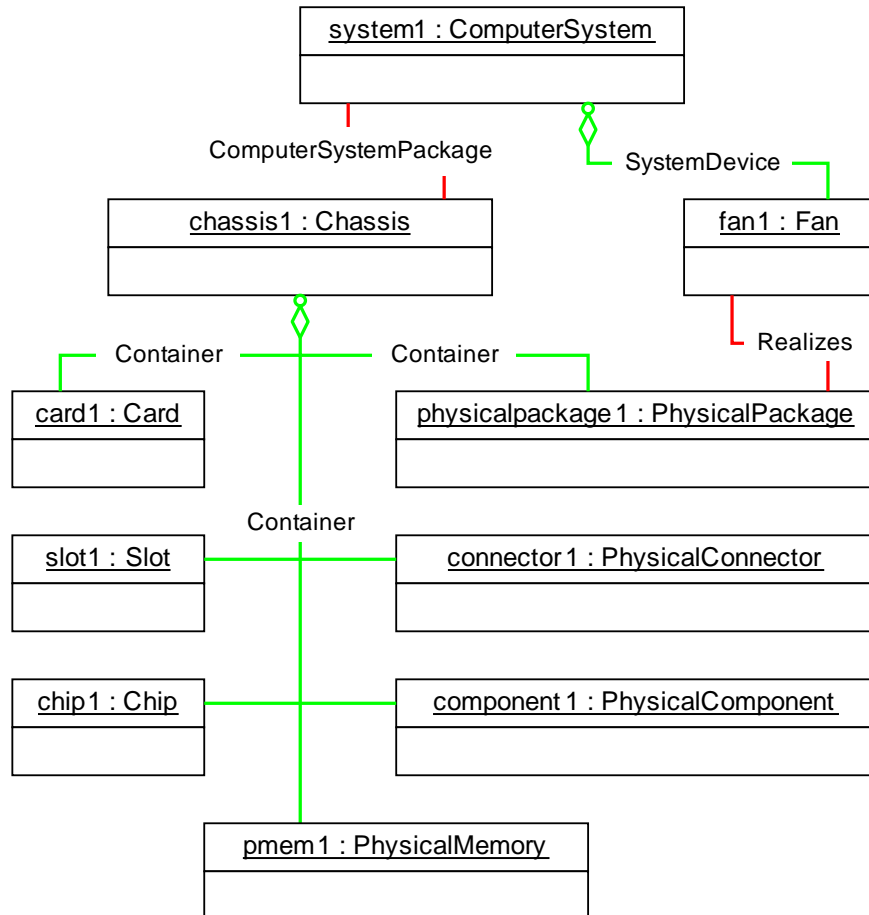
645

646

Figure 5 – Scoping Instance: Physical topology object diagram

647 **9.5 Physical topology**

648 Figure 6 represents another possible instantiation of the *Physical Asset Profile*. Chassis1 is a System  
 649 Chassis of system1. Physicalpackage1 is a Physical Package for fan1. The physical topology of chassis1  
 650 contains a single level because card1, slot1, chip1, pmem1, component1, connector1, and  
 651 physicalpackage1 are all directly associated to chassis1 through the instances of CIM\_Container.

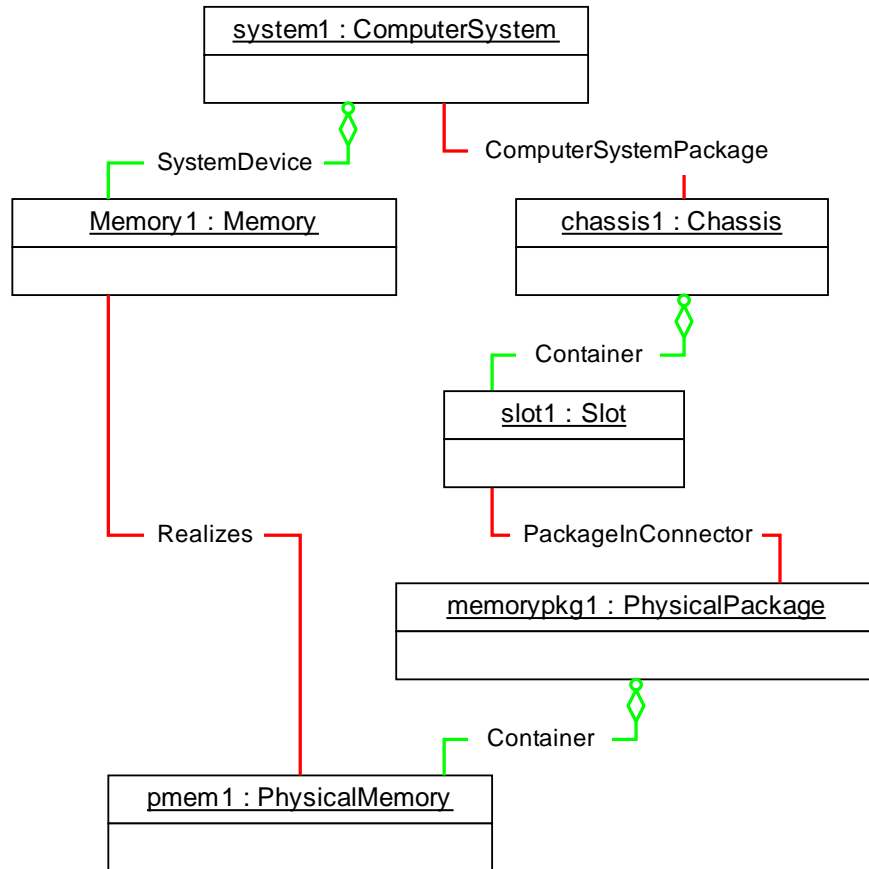


652

653 **Figure 6 – Physical Asset Profile: Topology object diagram**

654 **9.6 Physical memory**

655 Figure 7 represents another possible instantiation of the *Physical Asset Profile*. System1's system  
 656 memory is represented by Memory1. Memory1's physical aspects are represented by pmem1. chassis1 is  
 657 a System Chassis of system1. chassis1 contains slot1, into which the memory package, memorypkg1, is  
 658 plugged. memorypkg1 contains pmem1, the physical representation of the system memory, Memory1.



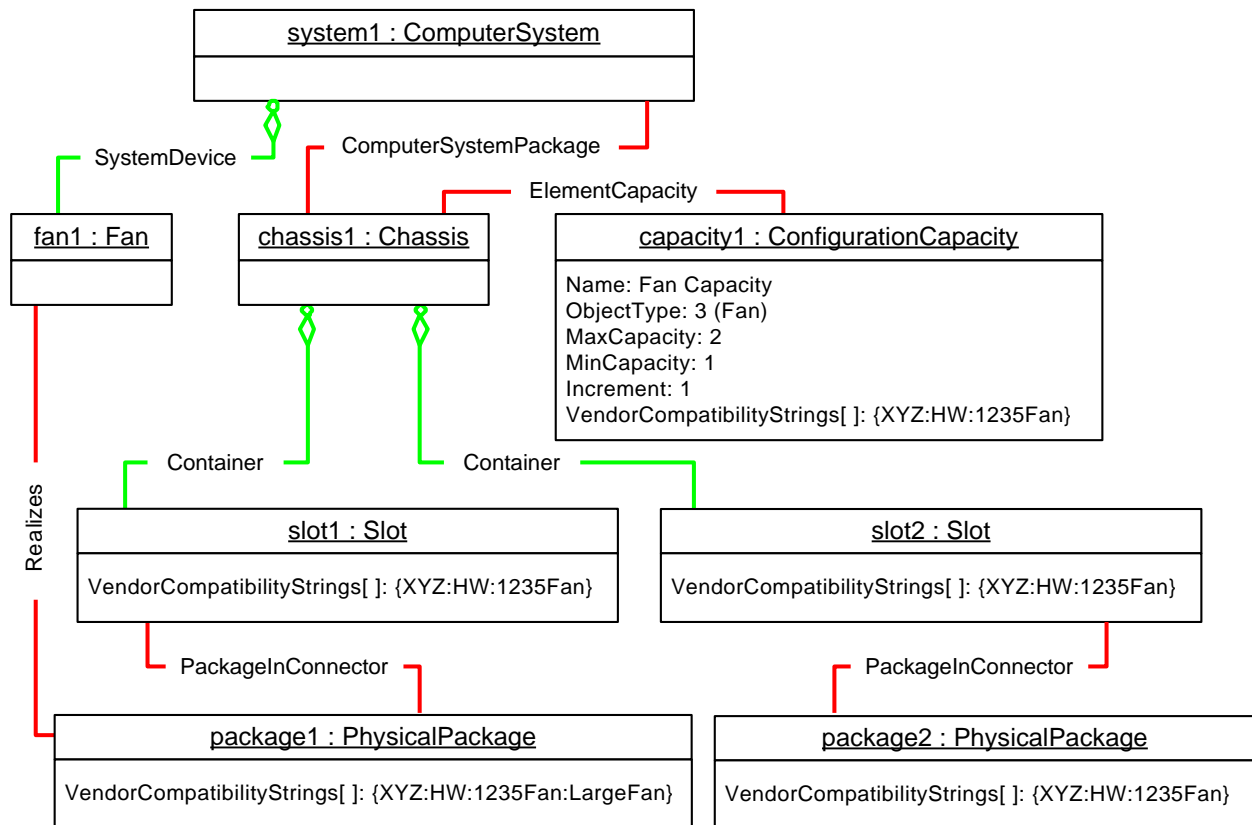
659

660

**Figure 7 – Physical memory topology object diagram**

661 **9.7 Representing configuration capacity**

662 Figure 8 represents another possible instantiation of the *Physical Asset Profile*. In this instantiation, the  
 663 chassis1 has two slots: slot1 and slot2. The slots are compatible with any type of XYZ:HW:1235Fan  
 664 packages, as advertised through the CIM\_Slot.VendorCompatibilityStrings property. slot1 and package1,  
 665 which is plugged into it, are compatible because the Delimited Substring matches for the  
 666 VendorCompatibilityStrings property. slot2 and package2, which is plugged into it, are compatible  
 667 because an element in the VendorCompatibilityStrings property of the CIM\_Slot instance is a Delimited  
 668 Substring of the element in the VendorCompatibilityStrings property of the CIM\_PhysicalPackage  
 669 instance. chassis1 also has a representation of its fan configuration capacity through capacity1. capacity1  
 670 indicates that chassis1 can have a maximum of two fans and should have at least one fan.

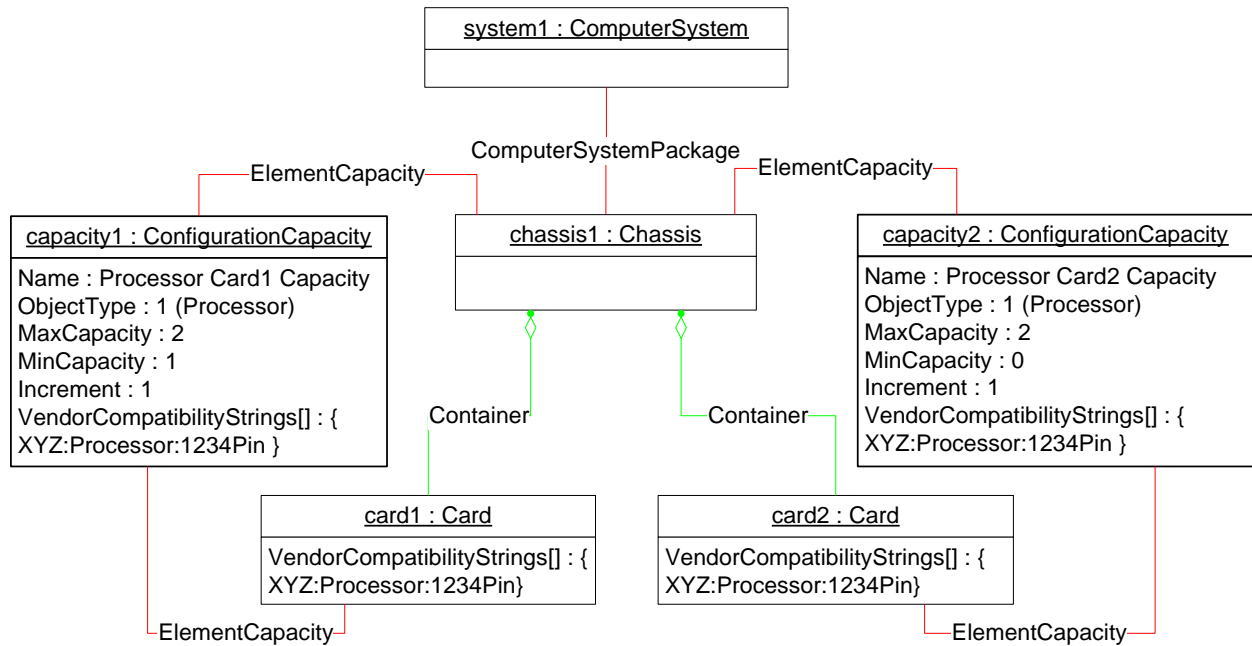


671

672

**Figure 8 – Configuration capacity object diagram**

673 Figure 9 represents another possible instantiation of the *Physical Asset Profile*. In this instantiation, the  
 674 chassis1 has two cards (card1 and card2) that hold processors. The configuration capacity for card1 is  
 675 represented by capacity1 because they are associated through the instance of CIM\_ElementCapacity. In  
 676 the same way, card2's configuration capacity is represented by capacity2. Because the  
 677 VendorCompatibilityStrings property value for capacity1 is equal to the VendorCompatibilityStrings  
 678 property value for capacity2, the maximum number of compatible processors could be determined by  
 679 adding the MaxCapacity property value of capacity1 to the MaxCapacity property value of capacity2. In  
 680 this case, the chassis1 could contain a maximum of four processors.



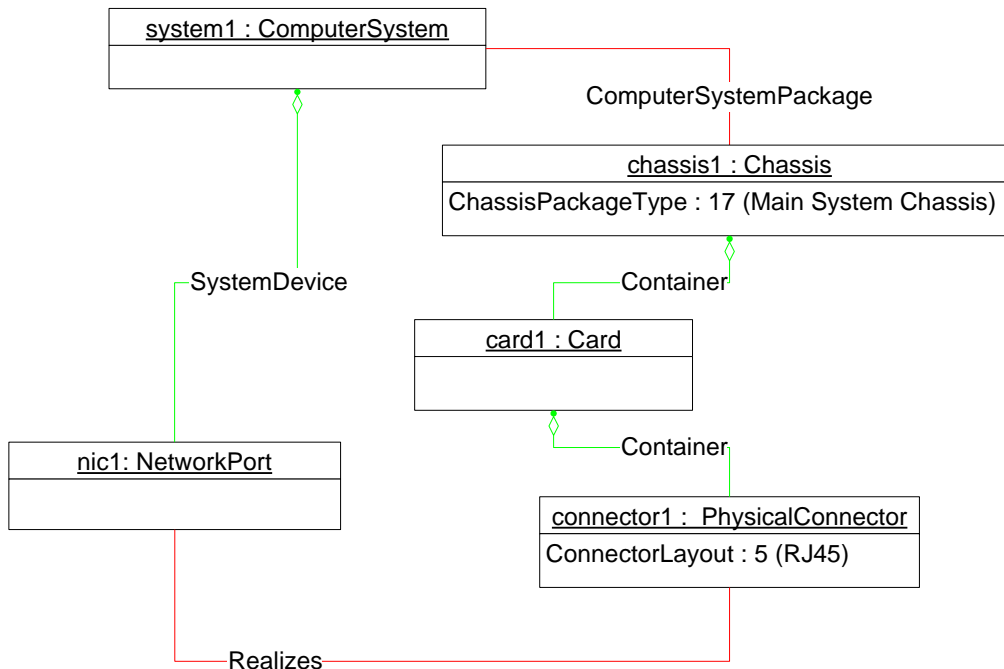
681

682

**Figure 9 – Additional configuration capacity object diagram**

## 683 9.8 Representing physical connector

684 Figure 10 represents another possible instantiation of the *Physical Asset Profile*. In this instance,  
 685 chassis1 contains a network card, card1. card1 has an RJ45 connector, connector1. connector1 is the  
 686 physical representation of nic1 network port within system1.



687

688

**Figure 10 – Network port connector object diagram**

## 689 9.9 Determining the part number of a failing component

690 Select the CIM\_PhysicalElement subclass instance that is associated through the CIM\_Realizes  
 691 association to the CIM\_LogicalDevice component that has a HealthState or OperationalStatus property  
 692 value indicating that the component is in a failure mode. Get the PartNumber property value for the  
 693 selected CIM\_PhysicalElement subclass instance.

## 694 9.10 Obtaining the physical inventory for all devices within a system

695 Select the CIM\_System instance representing the given system. Select all the CIM\_LogicalDevice  
 696 subclass instances that are associated with the CIM\_System instance through the CIM\_SystemDevice  
 697 association, and select all the CIM\_System instances associated through CIM\_SystemComponent  
 698 associations, and then follow the CIM\_SystemDevice association to select all the CIM\_LogicalDevice  
 699 subclass instances. Get all the property values of the CIM\_PhysicalElement subclass instances that are  
 700 associated to the selected CIM\_LogicalDevice subclass instances through the CIM\_Realizes association  
 701 and to the selected CIM\_System instances through the CIM\_SystemPackage association.

## 702 9.11 Obtaining the physical inventory for a System Chassis

703 Get all the property values of the Physical Package instances that are associated through the  
 704 CIM\_SystemPackaging association with the CIM\_System instance representing the given system.

705 **9.12 Determining whether the slot is empty**

706 Select all the CIM\_ElementInConnector instances that reference the CIM\_Slot instance that represents  
 707 the given slot. If no instances of CIM\_ElementInConnector that reference the CIM\_Slot instance exist,  
 708 then the slot is empty; otherwise the slot is occupied by the physical package represented by the instance  
 709 of CIM\_PhysicalPackage referenced by the CIM\_ElementInConnector association instance.

710 **9.13 Retrieving the fan capacity for the chassis**

711 For the CIM\_Chassis instance that represents the given chassis, select the associated instances of  
 712 CIM\_ConfigurationCapacity through the CIM\_ElementCapacity associations. Select  
 713 CIM\_ConfigurationCapacity instances that have the CIM\_ConfigurationCapacity.ObjectType property of 3  
 714 (Fan).

715 **9.14 Retrieving the maximum capacity of the type of fan package within the**  
 716 **chassis**

717 The particular type of fan package is identified through the given string, which is an element of the  
 718 VendorCompatibilityStrings array property of the Physical Package representing the fan package.

719 Select all the instances of CIM\_ConfigurationCapacity associated with the CIM\_Chassis instance through  
 720 instances of CIM\_ElementCapacity where the VendorCompatibilityStrings array property of the instance  
 721 of CIM\_ConfigurationCapacity contains elements equal to the given string. Add all the values for the  
 722 MaxCapacity property of the selected CIM\_ConfigurationCapacity instances.

723 **10 CIM Elements**

724 Table 10 shows the mandatory instances of CIM Elements for this profile. Instances of the following CIM  
 725 Elements shall be implemented as described in Table 10. Clauses 7 (“Implementation”) and 8 (“Methods”)  
 726 may impose additional requirements on these elements.

727 This profile contains definitions for non-abstract parent and child classes. All class definitions are treated  
 728 as leaf class definitions and the convention used is to replicate the properties in the following tables.

729 **Table 10 – CIM Elements: Physical Asset Profile**

Element Name	Requirement	Description
<b>Classes</b>		
CIM_Card	Conditional	See 7.1 and 10.1.
CIM_Chassis	Conditional	See 7.1 and 10.2.
CIM_Chip	Conditional	See 7.1 and 10.3.
CIM_ComputerSystemPackage	Conditional	See 7.1 and 10.4.
CIM_ConfigurationCapacity	Optional	See 7.7 and 10.5.
CIM_ConnectedTo	Optional	See 10.6.
CIM_Container	Optional	See 7.1 and 10.7.
CIM_ElementCapabilities	Conditional	See 10.8.
CIM_ElementCapacity	Conditional	See 7.7 and 10.9.
CIM_ElementInConnector	Optional	See 7.1 and 10.10.
CIM_PhysicalAssetCapabilities	Optional	See 7.4 and 10.11.
CIM_PhysicalComponent	Conditional	See 7.1 and 10.12.
CIM_PhysicalConnector	Conditional	See 7.1 and 10.13.

Element Name	Requirement	Description
CIM_PhysicalFrame	Conditional	See 7.1 and 10.14.
CIM_PhysicalMemory	Conditional	See 7.1 and 10.15.
CIM_PhysicalPackage	Conditional	See 7.1 and 10.16.
CIM_Rack	Conditional	See 7.1 and 10.17.
CIM_Realizes	Conditional	See 7.1 and 10.18.
CIM_RegisteredProfile	Mandatory	See 10.19.
CIM_Slot	Conditional	See 7.1 and 10.20.
CIM_SystemPackaging	Conditional	See 7.1 and 10.21.
Indications		
None defined in this profile		

730 NOTE Abstract classes are not shown in the tables in the following clauses.

## 731 10.1 CIM\_Card

732 CIM\_Card represents the processor card and its FRU data. Table 11 contains the requirements for  
733 properties of the instance.

734 **Table 11 – Class: CIM\_Card**

Elements	Requirement	Notes
Tag	Mandatory	<b>Key</b>
CreationClassName	Mandatory	<b>Key</b>
HostingBoard	Optional	This property should be implemented.
PackageType	Mandatory	None
Manufacturer	Conditional	See 7.4.
Model	Conditional	See 7.4.
SerialNumber	Conditional	See 7.4.
PartNumber	Conditional	See 7.4.
SKU	Conditional	See 7.4.
CanBeFRUed	Optional	This property should be implemented when the Physical Element can be replaced in the field.
VendorCompatibilityStrings	Optional	See 7.5.
Version	Optional	The property shall be the hardware version.
Name	Optional	
ElementName	Mandatory	This property shall be formatted as a free-form string of variable length (pattern “.*”).



735 **10.2 CIM\_Chassis**

736 CIM\_Chassis represents the chassis and its FRU data. Table 12 contains the requirements for properties  
 737 of the instance.

738 **Table 12 – Class: CIM\_Chassis**

Elements	Requirement	Notes
Tag	Mandatory	<b>Key</b>
CreationClassName	Mandatory	<b>Key</b>
PackageType	Mandatory	This property shall match 3 (Chassis/Frame).
ChassisPackageType	Mandatory	None
Manufacturer	Conditional	See 7.4.
Model	Conditional	See 7.4.
SerialNumber	Conditional	See 7.4.
PartNumber	Conditional	See 7.4.
SKU	Conditional	See 7.4.
CanBeFRUed	Optional	This property should be implemented when the Physical Element can be replaced in the field.
VendorCompatibilityStrings	Optional	See 7.5.
Version	Optional	The property shall be the hardware version.
Name	Optional	
ElementName	Mandatory	This property shall be formatted as a free-form string of variable length (pattern “.*”).

739 **10.3 CIM\_Chip**

740 CIM\_Chip represents the chip and its FRU data. Table 13 contains the requirements for properties of the  
 741 instance.

742 **Table 13 – Class: CIM\_Chip**

Elements	Requirement	Notes
Tag	Mandatory	<b>Key</b>
CreationClassName	Mandatory	<b>Key</b>
Manufacturer	Conditional	See 7.4.
Model	Conditional	See 7.4.
SerialNumber	Conditional	See 7.4.
PartNumber	Conditional	See 7.4.
SKU	Conditional	See 7.4.
CanBeFRUed	Optional	This property should be implemented when the Physical Element can be replaced in the field.
ElementName	Mandatory	This property shall be formatted as a free-form string of variable length (pattern “.*”).

743 **10.4 CIM\_ComputerSystemPackage**

744 CIM\_ComputerSystemPackage associates CIM\_ComputerSystem, representing the managed system,  
745 with a System Chassis. Table 14 contains the requirements for properties of the instance.

746 **Table 14 – Class: CIM\_ComputerSystemPackage**

Elements	Requirement	Notes
Antecedent	Mandatory	<b>Key:</b> This property shall reference the System Chassis. Cardinality * (indicating zero or many references)
Dependent	Mandatory	<b>Key:</b> This property shall reference the CIM_ComputerSystem representing the managed system. Cardinality * (indicating zero or many references)
PlatformGUID	Mandatory	This property shall match “^[0.9A.F]{32}\$” or, when unknown, shall match “0” .

747 **10.5 CIM\_ConfigurationCapacity**

748 CIM\_ConfigurationCapacity advertises the possible configuration of a System Chassis. Table 15 contains  
749 the requirements for properties of the instance.

750 **Table 15 – Class: CIM\_ConfigurationCapacity**

Elements	Requirement	Notes
Name	Mandatory	<b>Key</b>
ElementName	Mandatory	None
ObjectType	Mandatory	None
OtherTypeDescription	Conditional	This property shall be implemented when ObjectType matches 0 (Other).
MinimumCapacity	Optional	This property should be implemented.
MaximumCapacity	Mandatory	0 shall mean unknown.
Increment	Mandatory	0 shall mean unknown.
VendorCompatibilityStrings	Optional	See 7.5.

751 **10.6 CIM\_ConnectedTo**

752 CIM\_ConnectedTo associates the CIM\_PhysicalConnector or CIM\_Slot instances that represent  
753 connectors that are connected together. Table 16 contains the requirements for properties of the  
754 instance.

755 **Table 16 – Class: CIM\_ConnectedTo**

Elements	Requirement	Notes
Antecedent	Mandatory	<b>Key:</b> This property shall reference the CIM_PhysicalConnector or CIM_Slot instance. Cardinality * (indicating zero or many references)
Dependent	Mandatory	<b>Key:</b> This property shall reference the CIM_PhysicalConnector or CIM_Slot instance. Cardinality * (indicating zero or many references)

756 **10.7 CIM\_Container**

757 CIM\_Container associates a Physical Package with Physical Elements representing the physical  
 758 elements that reside within the package. Table 17 contains the requirements for properties of the  
 759 instance.

760 **Table 17 – Class: CIM\_Container**

Elements	Requirement	Notes
GroupComponent	Mandatory	<b>Key:</b> This property shall reference the Physical Package that represents the container. Cardinality 0.1 (indicating zero or one reference)
PartComponent	Mandatory	<b>Key:</b> This property shall reference the Physical Element that is contained within the package. Cardinality * (indicating zero or many references)

761 **10.8 CIM\_ElementCapabilities**

762 CIM\_ElementCapabilities associates Physical Elements with the CIM\_PhysicalAssetCapabilities  
 763 instances that advertise the physical capabilities. CIM\_ElementCapabilities shall be instantiated when an  
 764 instance of CIM\_PhysicalAssetCapabilities exists. Table 18 contains the requirements for properties of  
 765 the instance.

766 **Table 18 – Class: CIM\_ElementCapabilities**

Elements	Requirement	Notes
ManagedElement	Mandatory	<b>Key:</b> This property shall reference the Physical Element. Cardinality 1.* (indicating one or many references)
Capabilities	Mandatory	<b>Key:</b> This property shall reference the CIM_PhysicalAssetCapabilities class. Cardinality 0.1 (indicating zero or one reference)

767 **10.9 CIM\_ElementCapacity**

768 CIM\_ElementCapacity associates CIM\_ConfigurationCapacity instances with a System Chassis. Table 19  
 769 contains the requirements for properties of the instance.

770 **Table 19 – Class: CIM\_ElementCapacity**

Elements	Requirement	Notes
Capacity	Mandatory	<b>Key:</b> This property shall reference the CIM_ConfigurationCapacity instance. Cardinality * indicating zero or many references
Element	Mandatory	<b>Key:</b> This property shall reference the System Chassis or Physical Package. Cardinality 1.* (indicating one or many references)

771 **10.10 CIM\_ElementInConnector**

772 CIM\_ElementInConnector associates a CIM\_PhysicalConnector or CIM\_Slot instance, representing the  
 773 connector or slot, with Physical Packages (instances of CIM\_PhysicalPackage or  
 774 CIM\_PhysicalComponent). Table 20 contains the requirements for properties of the instance.

775 **Table 20 – Class: CIM\_ElementInConnector**

Elements	Requirement	Notes
Antecedent	Mandatory	<b>Key:</b> This property shall reference the instance of CIM_PhysicalConnector or CIM_Slot. Cardinality * (indicating zero or many references)
Dependent	Mandatory	<b>Key:</b> This property shall reference the CIM_PhysicalPackage or CIM_PhysicalComponent. Cardinality 0.1 (indicating zero or one reference)

776 **10.11 CIM\_PhysicalAssetCapabilities**

777 CIM\_PhysicalAssetCapabilities advertises whether the associated instance of a CIM\_PhysicalElement  
 778 subclass contains FRU data. Table 21 contains the requirements for properties of the instance.

779 **Table 21 – Class: CIM\_PhysicalAssetCapabilities**

Elements	Requirement	Notes
InstanceID	Mandatory	<b>Key</b>
ElementName	Mandatory	None
FRUInfoSupported	Mandatory	See 7.4.

780 **10.12 CIM\_PhysicalComponent**

781 CIM\_PhysicalComponent represents any physical element that cannot be further decomposed, such as  
 782 ASIC or tape, and its FRU data. Table 22 contains the requirements for properties of the instance.

783 **Table 22 – Class: CIM\_PhysicalComponent**

Elements	Requirement	Notes
Tag	Mandatory	<b>Key</b>
CreationClassName	Mandatory	<b>Key</b>
Manufacturer	Conditional	See 7.4.
Model	Conditional	See 7.4.
SerialNumber	Conditional	See 7.4.
PartNumber	Conditional	See 7.4.
SKU	Conditional	See 7.4.
CanBeFRUed	Optional	This property should be implemented when the Physical Element can be replaced in the field.
ElementName	Mandatory	This property shall be formatted as a free-form string of variable length (pattern “. *”).

784 **10.13 CIM\_PhysicalConnector**

785 CIM\_PhysicalConnector represents the physical connector. Table 23 contains the requirements for  
 786 properties of the instance.

787 **Table 23 – Class: CIM\_PhysicalConnector**

Elements	Requirement	Notes
Tag	Mandatory	<b>Key</b>
CreationClassName	Mandatory	<b>Key</b>
ConnectorLayout	Mandatory	None
Manufacturer	Conditional	See 7.4.
Model	Conditional	See 7.4.
SerialNumber	Conditional	See 7.4.
PartNumber	Conditional	See 7.4.
SKU	Conditional	See 7.4.
ElementName	Mandatory	This property shall be formatted as a free-form string of variable length (pattern “.*”).

788 **10.14 CIM\_PhysicalFrame**

789 CIM\_PhysicalFrame represents the frame and its FRU data. Table 24 contains the requirements for  
 790 properties of the instance.

791 **Table 24 – Class: CIM\_PhysicalFrame**

Elements	Requirement	Notes
Tag	Mandatory	<b>Key</b>
CreationClassName	Mandatory	<b>Key</b>
PackageType	Mandatory	None
Manufacturer	Conditional	See 7.4.
Model	Conditional	See 7.4.
SerialNumber	Conditional	See 7.4.
PartNumber	Conditional	See 7.4.
SKU	Conditional	See 7.4.
VendorCompatibilityStrings	Optional	See 7.5.
CanBeFRUed	Optional	This property should be implemented when the Physical Element can be replaced in the field.
Version	Optional	The property shall be the hardware version.
Name	Optional	
ElementName	Mandatory	This property shall be formatted as a free-form string of variable length (pattern “.*”).

792 **10.15 CIM\_PhysicalMemory**

793 CIM\_PhysicalMemory represents the physical memory and its FRU data. Table 25 contains the  
794 requirements for properties of the instance.

795 **Table 25 – Class: CIM\_PhysicalMemory**

Elements	Requirement	Notes
Tag	Mandatory	<b>Key</b>
CreationClassName	Mandatory	<b>Key</b>
FormFactor	Mandatory	None
MemoryType	Mandatory	None
Speed	Mandatory	None
Capacity	Mandatory	None
BankLabel	Mandatory	None
Manufacturer	Conditional	See 7.4.
Model	Conditional	See 7.4.
SerialNumber	Conditional	See 7.4.
PartNumber	Conditional	See 7.4.
SKU	Conditional	See 7.4.
CanBeFRUed	Optional	This property should be implemented when the Physical Element can be replaced in the field.
ElementName	Mandatory	This property shall be formatted as a free-form string of variable length (pattern “.*”).

796 **10.16 CIM\_PhysicalPackage**

797 CIM\_PhysicalPackage represents the physical package and its FRU data. Table 26 contains the  
798 requirements for properties of the instance.

799 **Table 26 – Class: CIM\_PhysicalPackage**

Elements	Requirement	Notes
Tag	Mandatory	<b>Key</b>
CreationClassName	Mandatory	<b>Key</b>
PackageType	Mandatory	None
Manufacturer	Conditional	See 7.4.
Model	Conditional	See 7.4.
SerialNumber	Conditional	See 7.4.
PartNumber	Conditional	See 7.4.
SKU	Conditional	See 7.4.
VendorCompatibilityStrings	Optional	See 7.5.
CanBeFRUed	Optional	This property should be implemented when the Physical Element can be replaced in the field.
Version	Optional	The property shall be the hardware version.
Name	Optional	
ElementName	Mandatory	This property shall be formatted as a free-form string of variable length (pattern “.*”).

800 **10.17 CIM\_Rack**

801 CIM\_Rack represents the rack and its FRU data. Table 27 contains the requirements for properties of the  
802 instance.

803 **Table 27 – Class: CIM\_Rack**

Elements	Requirement	Notes
Tag	Mandatory	<b>Key</b>
CreationClassName	Mandatory	<b>Key</b>
TypeOfRack	Mandatory	None
PackageType	Mandatory	This property shall match 2 (Rack).
Manufacturer	Conditional	See 7.4.
Model	Conditional	See 7.4.
SerialNumber	Conditional	See 7.4.
PartNumber	Conditional	See 7.4.
SKU	Conditional	See 7.4.
VendorCompatibilityStrings	Optional	See 7.5.
CanBeFRUed	Optional	This property should be implemented when the Physical Element can be replaced in the field.
Version	Optional	The property shall be the hardware version.
Name	Optional	
ElementName	Mandatory	This property shall be formatted as a free-form string of variable length (pattern “.*”).

804 **10.18 CIM\_Realizes**

805 CIM\_Realizes associates an instance of a CIM\_LogicalDevice subclass, representing the logical device,  
806 with a Physical Element. Table 28 contains the requirements for properties of the instance.

807 **Table 28 – Class: CIM\_Realizes**

Elements	Requirement	Notes
Antecedent	Mandatory	<b>Key:</b> shall reference the Physical Element. Cardinality 1.* indicating one or many references
Dependent	Mandatory	<b>Key:</b> shall reference the instance of subclass of CIM_LogicalDevice Cardinality * indicating zero or many references

808 **10.19 CIM\_RegisteredProfile**

809 The CIM\_RegisteredProfile class is defined by the [Profile Registration Profile](#). Table 29 contains the  
810 requirements for properties of the class.

811 The requirements listed in Table 29 are in addition to those mandated by the [Profile Registration Profile](#).

812 **Table 29 – Class: CIM\_RegisteredProfile**

Elements	Requirement	Description
RegisteredName	Mandatory	This property shall have a value of "Physical Asset".
RegisteredVersion	Mandatory	This property shall have a value of "1.0.3".
RegisteredOrganization	Mandatory	This property shall have a value of 2 (DMTF).

813 NOTE Previous versions of this document included the suffix "Profile" for the RegisteredName value. If  
814 implementations querying for the RegisteredName value find the suffix "Profile," they should ignore the suffix, with  
815 any surrounding white spaces, before any comparison is done with the value as specified in this document.

816 **10.20 CIM\_Slot**

817 CIM\_Slot represents the slot and its FRU data. Table 30 contains the requirements for properties of the  
818 instance.

819 **Table 30 – Class: CIM\_Slot**

Elements	Requirement	Notes
Tag	Mandatory	<b>Key</b>
CreationClassName	Mandatory	<b>Key</b>
Number	Mandatory	None
ConnectorLayout	Mandatory	None
Manufacturer	Conditional	See 7.4.
Model	Conditional	See 7.4.
SerialNumber	Conditional	See 7.4.
PartNumber	Conditional	See 7.4.
SKU	Conditional	See 7.4.
VendorCompatibilityStrings	Optional	See 7.5.
ElementName	Mandatory	This property shall be formatted as a free-form string of variable length (pattern ".*").



820 **10.21 CIM\_SystemPackaging**

821 CIM\_SystemPackaging associates CIM\_System, which represents the managed system, with a System  
 822 Chassis. Table 31 contains the requirements for properties of the instance.

823 **Table 31 – Class: CIM\_SystemPackaging**

Elements	Requirement	Notes
Antecedent	Mandatory	<b>Key:</b> This property shall reference the System Chassis. Cardinality * (indicating zero or many references)
Dependent	Mandatory	<b>Key:</b> This property shall reference the CIM_System representing the managed system. Cardinality * (indicating zero or many references)

824

825  
826  
827  
828

## ANNEX A (informative)

### Change log

Version	Date	Description
1.0.0	2007-12-11	
1.0.1	2008-06-09	Incorporated errata submitted for the Final Standard.
1.0.2	2009-04-06	DMTF Standard Release Incorporated errata on CIM_PhysicalMemory.Speed property values for unknown or variable speeds.
1.0.3	2016-05-17	Incorporated errata for clarifying white space related requirements for FRU properties.

829  
830