



1

2

3

4

**Document Number: DSP1011**

**Date: 2009-06-19**

**Version: 1.0.2**

5 **Physical Asset Profile**

6 **Document Type: Specification**

7 **Document Status: DMTF Standard**

8 **Document Language: E**

## 9 Copyright Notice

10 Copyright © 2009 Distributed Management Task Force, Inc. (DMTF). All rights reserved.

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

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25 withdrawn or modified after publication, and shall be indemnified and held harmless by any party  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

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

31

# CONTENTS

33	Foreword .....	7
34	Introduction .....	8
35	1 Scope .....	9
36	2 Normative References.....	9
37	2.1 Approved References .....	9
38	2.2 Other References.....	9
39	3 Terms and Definitions .....	9
40	4 Symbols and Abbreviated Terms .....	11
41	5 Synopsis .....	11
42	6 Description .....	12
43	7 Implementation.....	13
44	7.1 Physical Element .....	13
45	7.2 Finding the Scoping Instance of the CIM_System or CIM_ComputerSystem Class.....	14
46	7.3 Modeling the Physical Aspects of Logical Representation of Devices .....	15
47	7.4 Support for the Physical Element's FRU Information .....	15
48	7.5 Compatibility of Physical Packages .....	15
49	7.6 Modeling System Chassis .....	16
50	7.7 Modeling Configuration Capacity.....	16
51	7.8 Modeling Physical Hierarchy .....	16
52	7.9 Modeling a Physical Memory .....	16
53	8 Methods.....	17
54	8.1 Profile Conventions for Operations.....	17
55	8.2 CIM_Card.....	17
56	8.3 CIM_Chassis.....	17
57	8.4 CIM_Chip .....	17
58	8.5 CIM_ComputerSystemPackage .....	17
59	8.6 CIM_ConfigurationCapacity .....	18
60	8.7 CIM_ConnectedTo.....	18
61	8.8 CIM_Container.....	18
62	8.9 CIM_ElementCapabilities .....	18
63	8.10 CIM_ElementCapacity .....	19
64	8.11 CIM_ElementInConnector .....	19
65	8.12 CIM_PhysicalAssetCapabilities .....	19
66	8.13 CIM_PhysicalComponent .....	19
67	8.14 CIM_PhysicalConnector .....	19
68	8.15 CIM_PhysicalFrame.....	19
69	8.16 CIM_PhysicalMemory .....	20
70	8.17 CIM_PhysicalPackage .....	20
71	8.18 CIM_Rack .....	20
72	8.19 CIM_Realizes.....	20
73	8.20 CIM_Slot .....	20
74	8.21 CIM_SystemPackaging.....	20
75	9 Use Cases .....	21
76	9.1 System Chassis FRU Information.....	21
77	9.2 Fan Package FRU Information .....	21
78	9.3 Finding the Scoping Instance for a Fan Package.....	22
79	9.4 Physical Topology and Finding the Scoping Instance .....	23
80	9.5 Physical Topology.....	24
81	9.6 Physical Memory.....	25
82	9.7 Representing Configuration Capacity .....	26
83	9.8 Representing Physical Connector .....	28
84	9.9 Determining the Part Number of a Failing Component.....	28

85	9.10	Obtaining the Physical Inventory for All Devices within a System.....	28
86	9.11	Obtaining the Physical Inventory for a System Chassis.....	28
87	9.12	Determining Whether the Slot Is Empty.....	29
88	9.13	Retrieving the Fan Capacity for the Chassis.....	29
89	9.14	Retrieving the Maximum Capacity of the Type of Fan Package within the Chassis.....	29
90	10	CIM Elements.....	29
91	10.1	CIM_Card.....	30
92	10.2	CIM_Chassis.....	31
93	10.3	CIM_Chip.....	31
94	10.4	CIM_ComputerSystemPackage.....	32
95	10.5	CIM_ConfigurationCapacity.....	32
96	10.6	CIM_ConnectedTo.....	32
97	10.7	CIM_Container.....	33
98	10.8	CIM_ElementCapabilities.....	33
99	10.9	CIM_ElementCapacity.....	33
100	10.10	CIM_ElementInConnector.....	34
101	10.11	CIM_PhysicalAssetCapabilities.....	34
102	10.12	CIM_PhysicalComponent.....	34
103	10.13	CIM_PhysicalConnector.....	35
104	10.14	CIM_PhysicalFrame.....	35
105	10.15	CIM_PhysicalMemory.....	36
106	10.16	CIM_PhysicalPackage.....	36
107	10.17	CIM_Rack.....	37
108	10.18	CIM_Realizes.....	37
109	10.19	CIM_RegisteredProfile.....	38
110	10.20	CIM_Slot.....	38
111	10.21	CIM_SystemPackaging.....	39
112		ANNEX A (informative) Change Log.....	40

113

## 114 Figures

115	Figure 1 – Physical Asset Profile: Profile Class Diagram.....	12
116	Figure 2 – System Chassis Object Diagram.....	21
117	Figure 3 – CIM_PhysicalPackage Object Diagram.....	22
118	Figure 4 – Scoping Instance: Logical Device Object Diagram.....	22
119	Figure 5 – Scoping Instance: Physical Topology Object Diagram.....	23
120	Figure 6 – Physical Asset Profile: Topology Object Diagram.....	24
121	Figure 7 – Physical Memory Topology Object Diagram.....	25
122	Figure 8 – Configuration Capacity Object Diagram.....	26
123	Figure 9 – Additional Configuration Capacity Object Diagram.....	27
124	Figure 10 – Network Port Connector Object Diagram.....	28

125

## 126 Tables

127	Table 1 – Referenced Profiles.....	11
128	Table 2 – Operations: CIM_ComputerSystemPackage.....	17
129	Table 3 – Operations: CIM_ConnectedTo.....	18
130	Table 4 – Operations: CIM_Container.....	18
131	Table 5 – Operations: CIM_ElementCapabilities.....	18
132	Table 6 – Operations: CIM_ElementCapacity.....	19

133 Table 7 – Operations: CIM\_ElementInConnector ..... 19

134 Table 8 – Operations: CIM\_Realizes ..... 20

135 Table 9 – Operations: CIM\_SystemPackaging ..... 20

136 Table 10 – CIM Elements: Physical Asset Profile ..... 29

137 Table 11 – Class: CIM\_Card ..... 30

138 Table 12 – Class: CIM\_Chassis ..... 31

139 Table 13 – Class: CIM\_Chip ..... 31

140 Table 14 – Class: CIM\_ComputerSystemPackage ..... 32

141 Table 15 – Class: CIM\_ConfigurationCapacity ..... 32

142 Table 16 – Class: CIM\_ConnectedTo ..... 32

143 Table 17 – Class: CIM\_Container ..... 33

144 Table 18 – Class: CIM\_ElementCapabilities ..... 33

145 Table 19 – Class: CIM\_ElementCapacity ..... 33

146 Table 20 – Class: CIM\_ElementInConnector ..... 34

147 Table 21 – Class: CIM\_PhysicalAssetCapabilities ..... 34

148 Table 22 – Class: CIM\_PhysicalComponent ..... 34

149 Table 23 – Class: CIM\_PhysicalConnector ..... 35

150 Table 24 – Class: CIM\_PhysicalFrame ..... 35

151 Table 25 – Class: CIM\_PhysicalMemory ..... 36

152 Table 26 – Class: CIM\_PhysicalPackage ..... 36

153 Table 27 – Class: CIM\_Rack ..... 37

154 Table 28 – Class: CIM\_Realizes ..... 37

155 Table 29 – Class: CIM\_RegisteredProfile ..... 38

156 Table 30 – Class: CIM\_Slot ..... 38

157 Table 31 – Class: CIM\_SystemPackaging ..... 39



159

## Foreword

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

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

## 164 Acknowledgments

165 The authors wish to acknowledge the following people.

### 166 Editors:

- 167 • Jon Hass – Dell Inc.
- 168 • Khachatur Papanyan – Dell Inc.
- 169 • Jeff Hilland – HP
- 170 • Hemal Shah – Broadcom Corporation

### 171 Contributors:

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

182

## Introduction

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

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



193

# Physical Asset Profile

## 194 **1 Scope**

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

## 199 **2 Normative References**

200 The following referenced documents are indispensable for the application of this document. For dated  
201 references, only the edition cited applies. For undated references, the latest edition of the referenced  
202 document (including any amendments) applies.

### 203 **2.1 Approved References**

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

206 DMTF DSP0200, *CIM Operations over HTTP 1.2*,  
207 [http://www.dmtf.org/standards/published\\_documents/DSP0200\\_1.2.pdf](http://www.dmtf.org/standards/published_documents/DSP0200_1.2.pdf)

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

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

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

### 214 **2.2 Other References**

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

## 217 **3 Terms and Definitions**

218 For the purposes of this document, the following terms and definitions apply. For the purposes of this  
219 document, the terms and definitions in [DSP1033](#) and [DSP1001](#) also apply.

### 220 **3.1**

#### 221 **can**

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

### 223 **3.2**

#### 224 **cannot**

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

- 226 **3.3**  
227 **conditional**  
228 indicates requirements to be followed strictly in order to conform to the document when the specified  
229 conditions are met
- 230 **3.4**  
231 **mandatory**  
232 indicates requirements to be followed strictly in order to conform to the document and from which no  
233 deviation is permitted
- 234 **3.5**  
235 **may**  
236 indicates a course of action permissible within the limits of the document
- 237 **3.6**  
238 **need not**  
239 indicates a course of action permissible within the limits of the document
- 240 **3.7**  
241 **optional**  
242 indicates a course of action permissible within the limits of the document
- 243 **3.8**  
244 **referencing profile**  
245 indicates a profile that owns the definition of this class and can include a reference to this profile in its  
246 "Referenced Profiles" table
- 247 **3.9**  
248 **shall**  
249 indicates requirements to be followed strictly in order to conform to the document and from which no  
250 deviation is permitted
- 251 **3.10**  
252 **shall not**  
253 indicates requirements to be followed in order to conform to the document and from which no deviation is  
254 permitted
- 255 **3.11**  
256 **should**  
257 indicates that among several possibilities, one is recommended as particularly suitable, without  
258 mentioning or excluding others, or that a certain course of action is preferred but not necessarily required
- 259 **3.12**  
260 **should not**  
261 indicates that a certain possibility or course of action is deprecated but not prohibited
- 262 **3.13**  
263 **unspecified**  
264 indicates that this profile does not define any constraints for the referenced CIM element or operation
- 265 **3.14**  
266 **Delimited Substring**  
267 a substring element of the VendorCompatibilityStrings property of a Physical Element or an instance of  
268 CIM\_ConfigurationCapacity. The substring starts at the beginning of the string (representing an element

269 in the array of the VendorCompatibilityStrings property) and terminates at the end of the string, or at a  
 270 character that precedes a colon (:).

271 **3.15**

272 **Physical Element**

273 an instance of a CIM\_PhysicalElement subclass (such as CIM\_PhysicalConnector, CIM\_Slot,  
 274 CIM\_PhysicalComponent, CIM\_Chip, CIM\_PhysicalMemory, CIM\_PhysicalPackage,  
 275 CIM\_PhysicalFrame, CIM\_Chassis, CIM\_Rack, and CIM\_Card) that represents a physical element

276 **3.16**

277 **Physical Package**

278 an instance of a CIM\_PhysicalPackage or a CIM\_PhysicalPackage subclass (such as  
 279 CIM\_PhysicalFrame, CIM\_Chassis, CIM\_Rack, and CIM\_Card) or CIM\_PhysicalComponent or  
 280 CIM\_PhysicalComponent subclass (such as CIM\_Chip or CIM\_PhysicalMemory) that represents a  
 281 package

282 **3.17**

283 **System Chassis**

284 an instance of the CIM\_PhysicalElement or CIM\_Chassis that is associated to an instance of  
 285 CIM\_System or CIM\_ComputerSystem through the CIM\_SystemPackaging or  
 286 CIM\_ComputerSystemPackage association, representing the physical package of the managed system.

287 **4 Symbols and Abbreviated Terms**

288 **4.1**

289 **CIM**

290 Common Information Model

291 **4.2**

292 **FRU**

293 Field Replaceable Unit

294 **5 Synopsis**

295 **Profile Name:** Physical Asset

296 **Version:** 1.0.2

297 **Organization:** DMTF

298 **CIM Schema version:** 2.22

299 **Central Class:** CIM\_PhysicalElement

300 **Scoping Class:** CIM\_ManagedSystemElement

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

304 **Table 1 – Referenced Profiles**

Profile Name	Organization	Version	Relationship	Behavior
Profile Registration	DMTF	1.0	Mandatory	

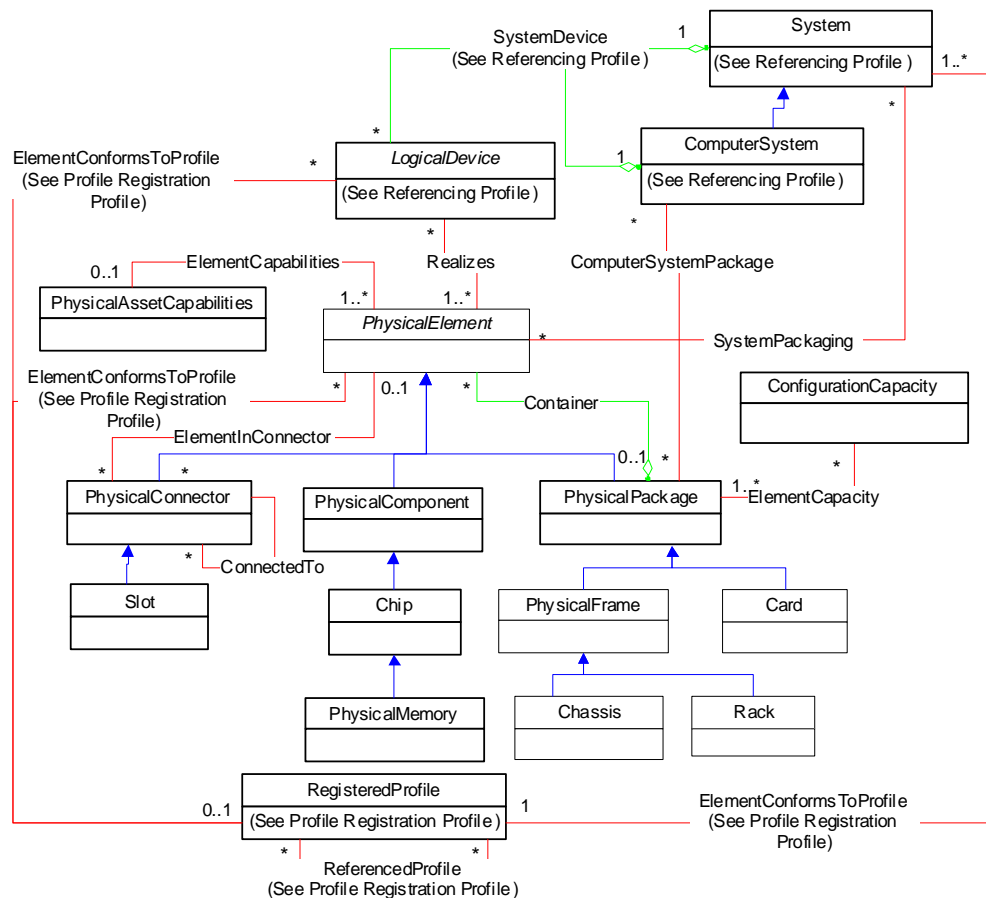
305 The Central Instance for the *Physical Asset Profile* shall be the instance of the CIM\_PhysicalElement  
 306 subclass.

307 The Scoping Instance for the *Physical Asset Profile* shall be the instance of the  
 308 CIM\_ManagedSystemElement. Note that this may include the subclass of CIM\_System, the  
 309 CIM\_ComputerSystem class. The Scoping Instance is determined using the algorithm described in  
 310 section 7.2.

## 311 6 Description

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

315 Figure 1 shows the CIM classes that are used in this profile. (For simplicity, the prefix CIM\_ has been  
 316 removed from the names of the classes.) A Physical Element (see section 3.15) describes the physical  
 317 properties, including the FRU information, of a managed element. The capabilities of the Physical  
 318 Elements are described by the properties of the CIM\_PhysicalAssetCapabilities class. The Physical  
 319 Elements could be associated to the logical representation of the managed element through the  
 320 CIM\_Realizes association. The enclosures or chassis of the managed systems are represented by a  
 321 CIM\_PhysicalElement or CIM\_Chassis instance that is associated to the  
 322 CIM\_System/CIM\_ComputerSystem instance through the  
 323 CIM\_SystemPackaging/CIM\_ComputerSystemPackage association and are referred to as a System  
 324 Chassis (see section 3.17). Configuration capacity of the System Chassis is also represented within this  
 325 profile by CIM\_ConfigurationCapacity instances.



326

327

**Figure 1 – Physical Asset Profile: Profile Class Diagram**

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

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

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

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

- 339 • the physical aspects of a managed system, such as the FRU information for the chassis (see  
340 section 7.6)
- 341 • the physical aspects of a specific managed element, such as the FRU information of a fan (see  
342 section 7.3)
- 343 • the physical hierarchy of a managed system, such as the relationship between chassis, slots, and  
344 packages (see section 7.8)
- 345 • the configuration capacity of a managed element, such as the minimum and maximum number of  
346 certain types of packages that the managed system can handle (see section 7.7)

## 347 **7 Implementation**

348 This section details the requirements related to the arrangement of instances and their properties for  
349 implementations of this profile.

### 350 **7.1 Physical Element**

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

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

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

## 359 7.2 Finding the Scoping Instance of the CIM\_System or CIM\_ComputerSystem 360 Class

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

363 I. If the selected instance is of a Physical Package, proceed as follows:

364 A. If the Physical Package is associated to the CIM\_LogicalDevice through the CIM\_Realizes  
365 association or to the CIM\_System/CIM\_ComputerSystem through the  
366 CIM\_SystemPackaging/CIM\_ComputerSystemPackage association, the Scoping Instance of  
367 the *Physical Asset Profile* shall be either of the following:

368 a. the Scoping Instance of the CIM\_LogicalDevice instance that is associated to the  
369 Physical Package through the instance of CIM\_Realizes

370 b. the Scoping Instance of CIM\_System/CIM\_ComputerSystem instance that is  
371 associated to the Physical Package through the instance of CIM\_SystemPackaging or  
372 CIM\_ComputerSystemPackage

373 B. If the Physical Package is not associated to the CIM\_LogicalDevice through the CIM\_Realizes  
374 association or to the CIM\_System or CIM\_ComputerSystem through the  
375 CIM\_SystemPackaging or CIM\_ComputerSystemPackage association, proceed as follows:

376 1. If the Physical Package is the Dependent or PartComponent reference in  
377 CIM\_ElementInConnector or CIM\_Container associations, respectively, choose one of the  
378 following paths:

379 a. If the Antecedent or GroupComponent reference of the association is a Physical  
380 Package, select the Antecedent or GroupComponent referenced instance, and go to  
381 I.A.

382 b. Else if the Antecedent or GroupComponent reference of the association is a Physical  
383 Element:

384 (1) If the Physical Element is associated to the CIM\_LogicalDevice through the  
385 CIM\_Realizes association, the Scoping Instance of the *Physical Asset Profile*  
386 shall be the Scoping Instance of the CIM\_LogicalDevice instance.

387 (2) If the Physical Element instance is not associated to the CIM\_LogicalDevice  
388 through the CIM\_Realizes association:

389 (a) If the Physical Element is the PartComponent reference in the  
390 CIM\_Container association:

391 1) If a Physical Package is the GroupComponent reference for the  
392 CIM\_Container association, select the GroupComponent referenced  
393 instance, and go to I.A.

394 2) If a Physical Element is the GroupComponent or Antecedent reference,  
395 go to I.B.1.b(1).

396 (b) If the Physical Element is not the PartComponent or Dependent reference in  
397 a CIM\_Container association, the Scoping Instance shall be the Central  
398 Instance; thus, the Central Instance is associated to the  
399 CIM\_RegisteredProfile instance.

400 2. Else the Scoping Instance shall be the Central Instance, thus, the Central Instance is  
401 associated to the CIM\_RegisteredProfile instance.

402 II. If the instance is not a Physical Package, go to I.B.1.b(1).

### 403 **7.3 Modeling the Physical Aspects of Logical Representation of Devices**

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

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

### 409 **7.4 Support for the Physical Element's FRU Information**

410 The Physical Element's support of FRU information shall be advertised by a  
411 CIM\_PhysicalAssetCapabilities instance associated with the Physical Element. At most, one instance of  
412 CIM\_PhysicalAssetCapabilities shall be associated with the Physical Element through the  
413 CIM\_ElementCapabilities association.

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

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

419 When FRU information is supported, the implementation shall populate the properties of the Physical  
420 Element below with non-null, non-blank values. At least one of these properties shall be non-null, non-  
421 blank of the pattern "[^WSP]+". If the SKU property is non-null, it shall be used to convey the FRU  
422 number. Some combination of the properties below should be used for replacement part information.

- 423 • Manufacturer
- 424 • Model
- 425 • PartNumber
- 426 • SerialNumber
- 427 • SKU

### 428 **7.5 Compatibility of Physical Packages**

429 When the Physical Package is instantiated, the implementation may represent the compatibility of the  
430 Physical Package. In that case, the conditions and requirements in this section shall apply.

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

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

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

## 441 **7.6 Modeling System Chassis**

442 The implementation may instantiate a System Chassis. When a System Chassis is instantiated, the  
443 System Chassis shall be associated with the instance of CIM\_System through the instance of  
444 CIM\_SystemPackaging, or with the instance of CIM\_ComputerSystem through the instance of  
445 CIM\_ComputerSystemPackage.

## 446 **7.7 Modeling Configuration Capacity**

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

450 When a System Chassis is present, the instrumentation shall associate all the instances of  
451 CIM\_ConfigurationCapacity to the System Chassis through the instances of CIM\_ElementCapacity.  
452 Additionally, when the configuration capacity is for a particular physical package represented by a  
453 Physical Package, the instrumentation may associate the Physical Package with the  
454 CIM\_ConfigurationCapacity through an instance of CIM\_ElementCapacity.

455 When instances of CIM Slot are instantiated, for each unique value of the  
456 CIM\_Slot.VendorCompatibilityStrings, an instance of CIM\_ConfigurationCapacity with an equal value for  
457 the CIM\_ConfigurationCapacity.VendorCompatibilityStrings property shall exist. Additional instances of  
458 CIM\_ConfigurationCapacity may exist.

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

## 463 **7.8 Modeling Physical Hierarchy**

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

- 466 • When a physical element resides within a package, the Physical Element shall be associated  
467 with the Physical Package through the CIM\_Container association.
- 468 • When a package is plugged or connected to a slot or connector, the Physical Package shall be  
469 associated with the CIM\_PhysicalConnector or CIM\_Slot instance through the  
470 CIM\_ElementInConnector association.
- 471 • When physical connectors or slots are connected, the CIM\_PhysicalConnector or CIM\_Slot  
472 instances shall be associated through the CIM\_ConnectedTo association.

## 473 **7.9 Modeling a Physical Memory**

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

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

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



## 483 8 Methods

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

### 486 8.1 Profile Conventions for Operations

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

489 The default list of operations is as follows:

- 490 • GetInstance
- 491 • Associators
- 492 • AssociatorNames
- 493 • References
- 494 • ReferenceNames
- 495 • EnumerateInstances
- 496 • EnumerateInstanceNames

### 497 8.2 CIM\_Card

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

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

### 500 8.3 CIM\_Chassis

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

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

### 503 8.4 CIM\_Chip

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

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

### 506 8.5 CIM\_ComputerSystemPackage

507 Table 2 lists implementation requirements for operations. If implemented, these operations shall be  
508 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 2, all operations in  
509 the default list in 8.1 shall be implemented as defined in [DSP0200](#).

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

511

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

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

## 512 8.6 CIM\_ConfigurationCapacity

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

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

## 515 8.7 CIM\_ConnectedTo

516 Table 3 lists implementation requirements for operations. If implemented, these operations shall be  
 517 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 3, all operations in  
 518 the default list in 8.1 shall be implemented as defined in [DSP0200](#).

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

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

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

## 521 8.8 CIM\_Container

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

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

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

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

## 527 8.9 CIM\_ElementCapabilities

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

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

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

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

## 533 8.10 CIM\_ElementCapacity

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

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

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

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

## 539 8.11 CIM\_ElementInConnector

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

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

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

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

## 545 8.12 CIM\_PhysicalAssetCapabilities

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

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

## 548 8.13 CIM\_PhysicalComponent

549 All operations in 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 8.14 CIM\_PhysicalConnector

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

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

## 554 8.15 CIM\_PhysicalFrame

555 All operations in 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 **8.16 CIM\_PhysicalMemory**

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

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

### 560 **8.17 CIM\_PhysicalPackage**

561 All operations in 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 **8.18 CIM\_Rack**

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

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

### 566 **8.19 CIM\_Realizes**

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

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

571

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

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

### 572 **8.20 CIM\_Slot**

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

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

### 575 **8.21 CIM\_SystemPackaging**

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

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

580

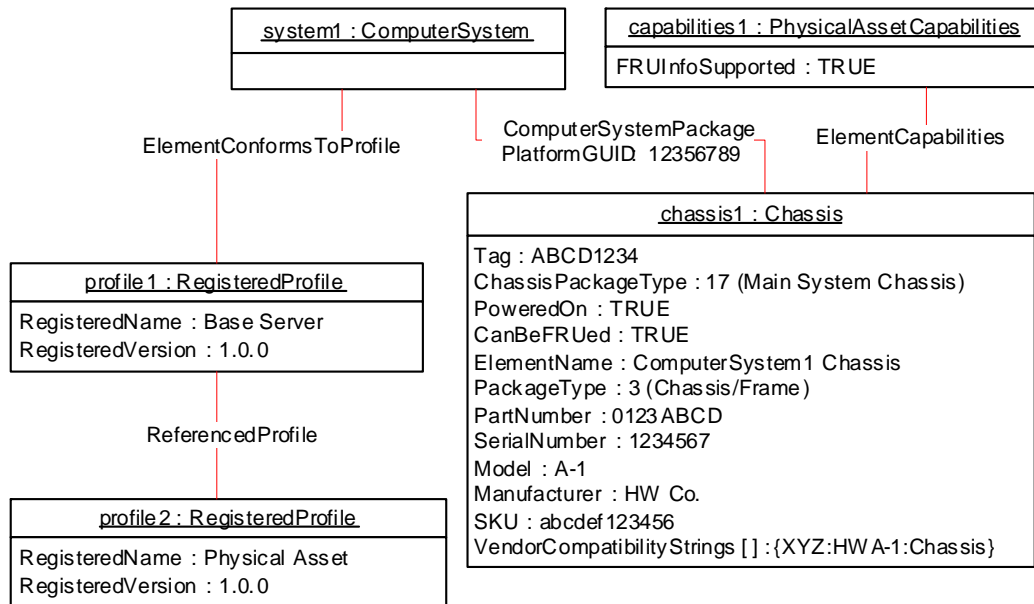
**Table 9 – Operations: CIM\_SystemPackaging**

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

581 **9 Use Cases**

582 **9.1 System Chassis FRU Information**

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

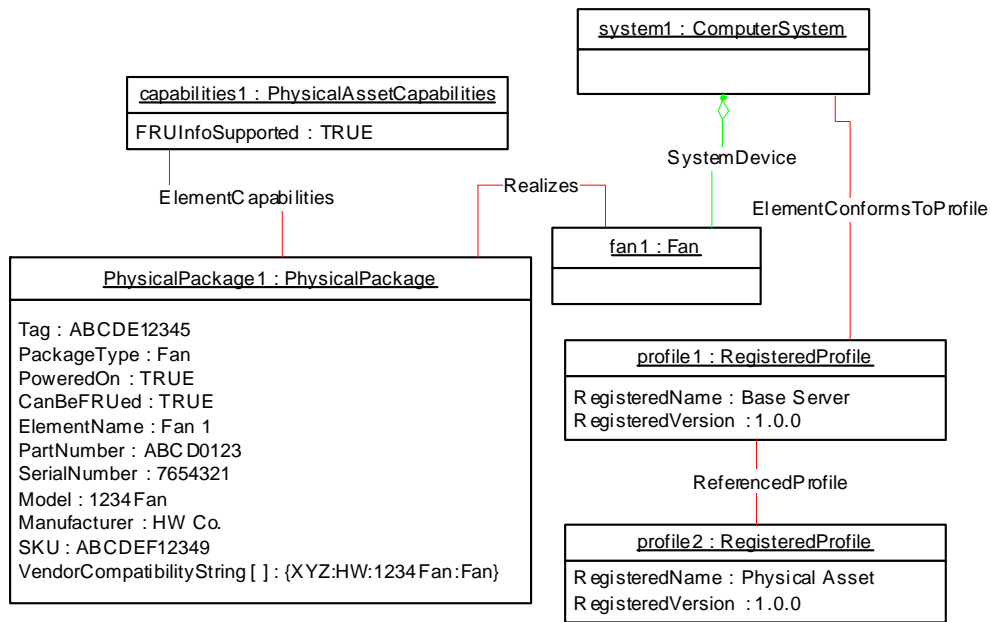


590

591 **Figure 2 – System Chassis Object Diagram**

592 **9.2 Fan Package FRU Information**

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



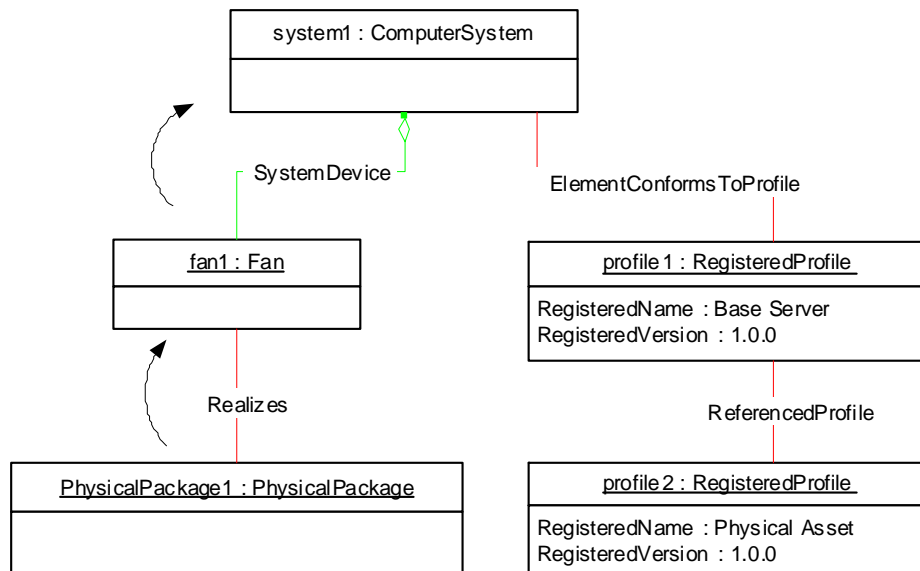
599

600

Figure 3 – CIM\_PhysicalPackage Object Diagram

601 **9.3 Finding the Scoping Instance for a Fan Package**

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



609

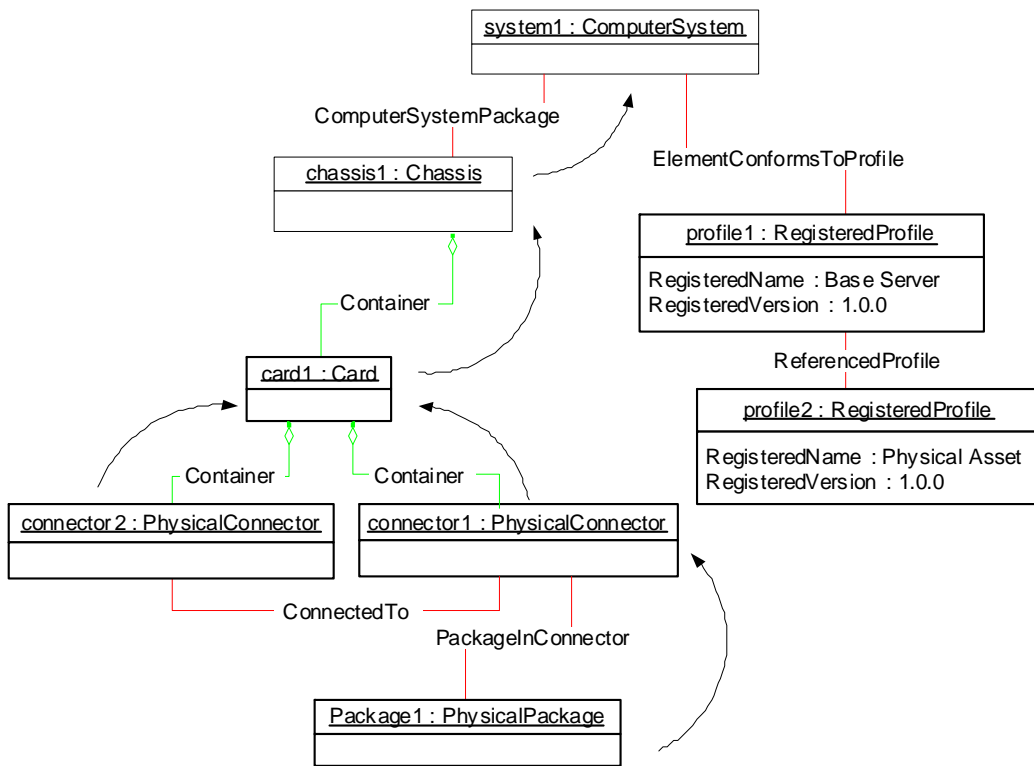
610

Figure 4 – Scoping Instance: Logical Device Object Diagram

611 **9.4 Physical Topology and Finding the Scoping Instance**

612 Figure 5 represents another possible instantiation of the *Physical Asset Profile*. To find the Scoping  
 613 Instance of package1, because package1 is referenced by the CIM\_ElementInConnector.Dependent  
 614 property, the client needs to select connector1, which is referenced by the  
 615 CIM\_ElementInConnector.Antecedent property. Then, because connector1 is referenced by the  
 616 CIM\_Container.PartComponent property, the client needs to select card1, which is referenced by the  
 617 CIM\_Container.GroupComponent. Then, because card1 is referenced by the  
 618 CIM\_Container.PartComponent property, the client needs to select chassis1, which is referenced by the  
 619 CIM\_Container.GroupComponent. Then, because chassis1 is associated to system1 through the  
 620 CIM\_ComputerSystemPackage association, system1 is the Scoping Instance of package1. The client can  
 621 traverse through the CIM\_ElementConformsToProfile and, subsequently, the CIM\_ReferencedProfile  
 622 association, to find profile2, which advertises the *Physical Asset Profile* information.

623 NOTE: To enable finding the Scoping Instance of connector2, the implementation has instantiated an instance of  
 624 CIM\_Container that references card1 and connector2. Merely instantiating the instance of CIM\_ConnectedTo  
 625 referencing connector2 will not conform to the algorithm described in section 7.2.

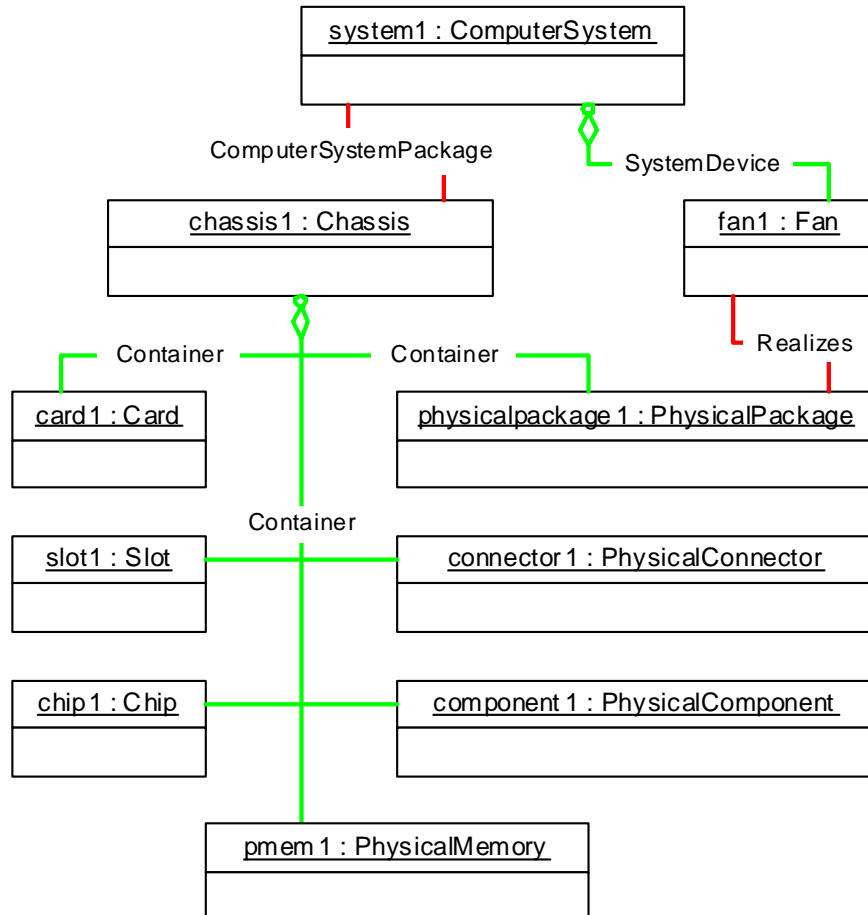


626

627 **Figure 5 – Scoping Instance: Physical Topology Object Diagram**

628 **9.5 Physical Topology**

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



633

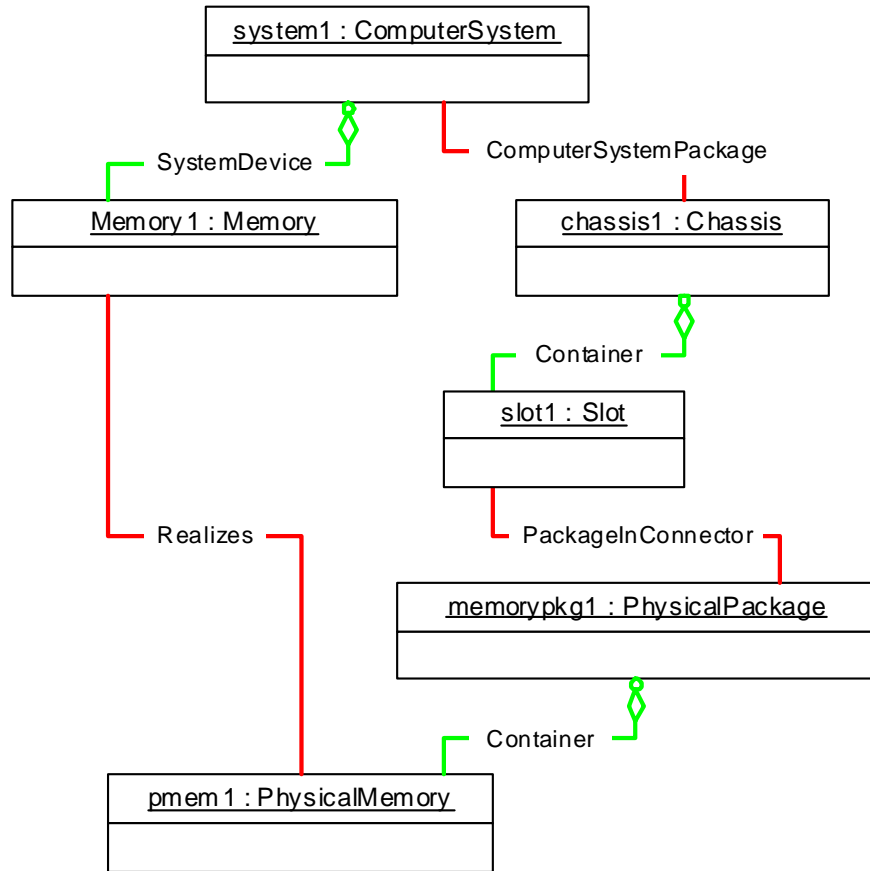
634

**Figure 6 – Physical Asset Profile: Topology Object Diagram**



635 **9.6 Physical Memory**

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



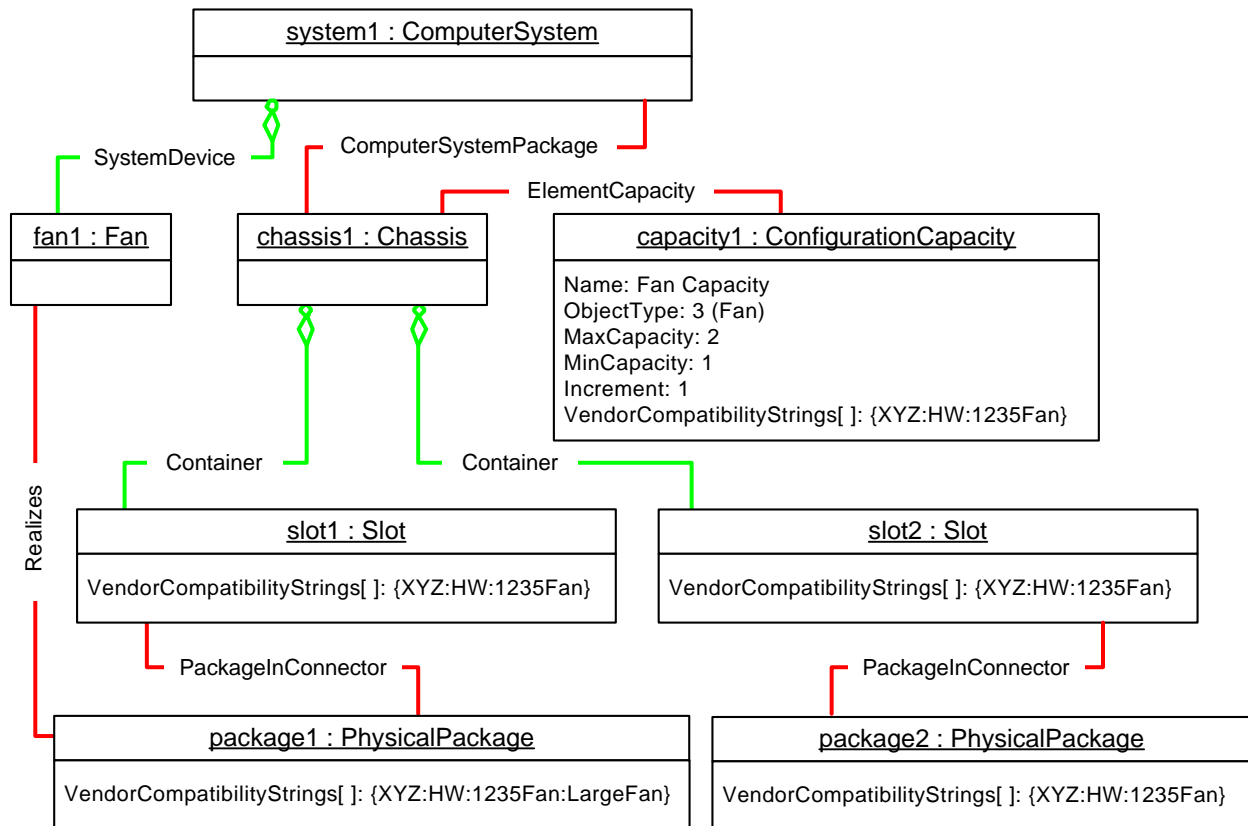
640

641

**Figure 7 – Physical Memory Topology Object Diagram**

## 642 9.7 Representing Configuration Capacity

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

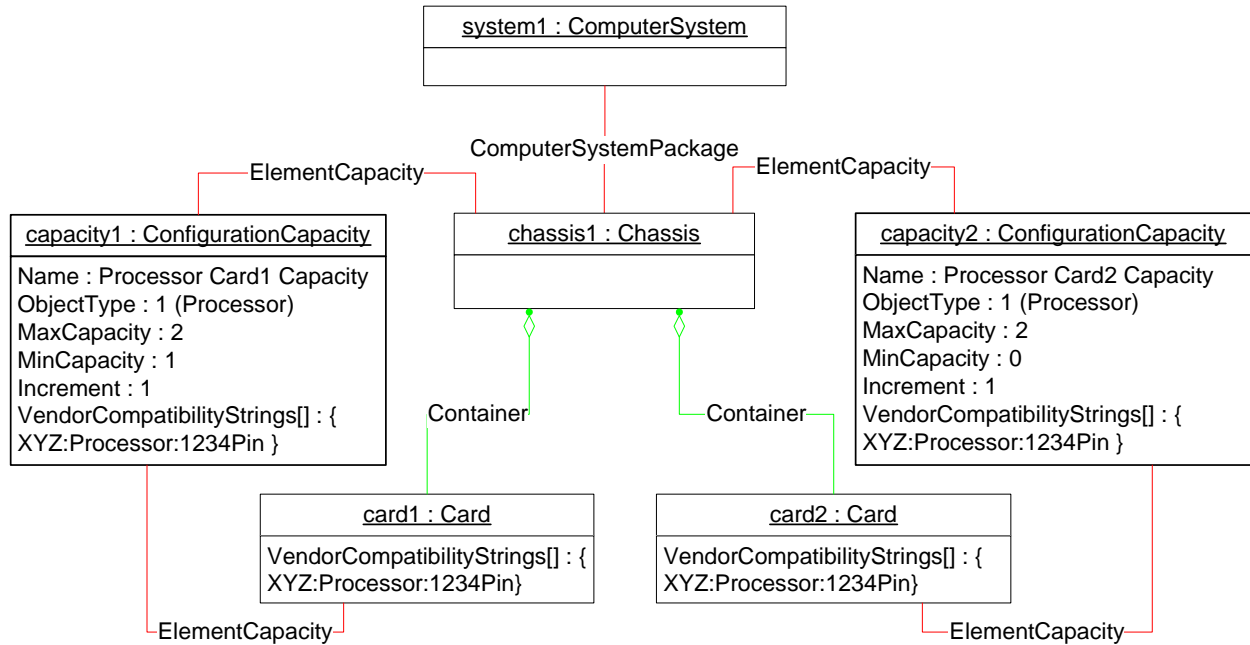


652

653

**Figure 8 – Configuration Capacity Object Diagram**

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



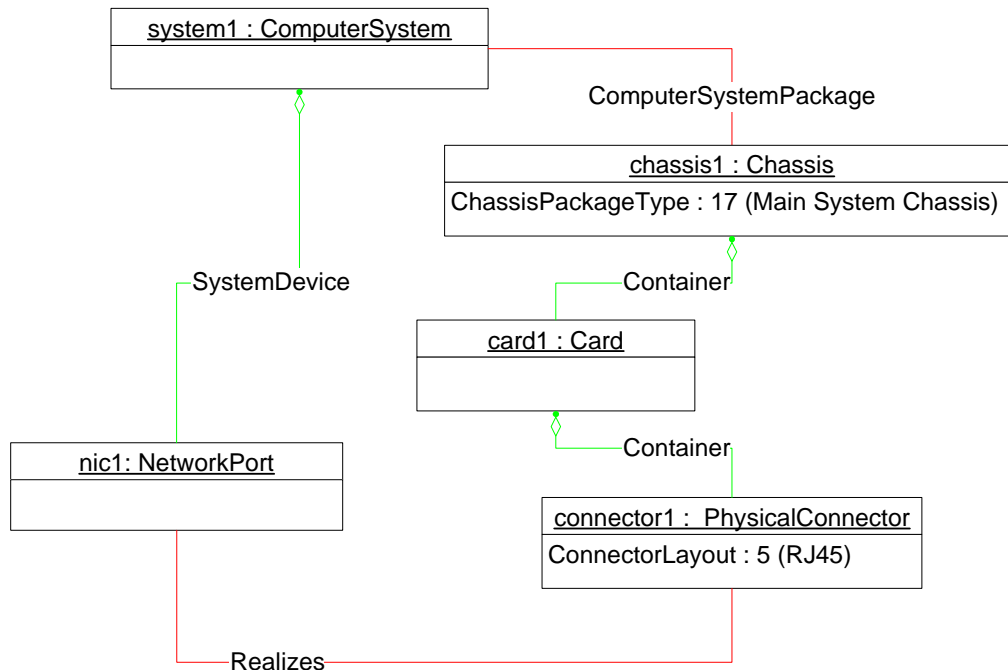
662

663

**Figure 9 – Additional Configuration Capacity Object Diagram**

## 664 9.8 Representing Physical Connector

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



668

669 **Figure 10 – Network Port Connector Object Diagram**

## 670 9.9 Determining the Part Number of a Failing Component

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

## 675 9.10 Obtaining the Physical Inventory for All Devices within a System

676 Select the CIM\_System instance representing the given system. Select all the CIM\_LogicalDevice  
 677 subclass instances that are associated with the CIM\_System instance through the CIM\_SystemDevice  
 678 association, and select all the CIM\_System instances associated through CIM\_SystemComponent  
 679 associations, and then follow the CIM\_SystemDevice association to select all the CIM\_LogicalDevice  
 680 subclass instances. Get all the property values of the CIM\_PhysicalElement subclass instances that are  
 681 associated to the selected CIM\_LogicalDevice subclass instances through the CIM\_Realizes association  
 682 and to the selected CIM\_System instances through the CIM\_SystemPackage association.

## 683 9.11 Obtaining the Physical Inventory for a System Chassis

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

686 **9.12 Determining Whether the Slot Is Empty**

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

691 **9.13 Retrieving the Fan Capacity for the Chassis**

692 For the CIM\_Chassis instance that represents the given chassis, select the associated instances of  
 693 CIM\_ConfigurationCapacity through the CIM\_ElementCapacity associations. Select  
 694 CIM\_ConfigurationCapacity instances that have the CIM\_ConfigurationCapacity.ObjectType property of 3  
 695 (Fan).

696 **9.14 Retrieving the Maximum Capacity of the Type of Fan Package within the**  
 697 **Chassis**

698 The particular type of fan package is identified through the given string, which is an element of the  
 699 VendorCompatibilityStrings array property of the Physical Package representing the fan package.  
 700 Select all the instances of CIM\_ConfigurationCapacity associated with the CIM\_Chassis instance through  
 701 instances of CIM\_ElementCapacity where the VendorCompatibilityStrings array property of the instance  
 702 of CIM\_ConfigurationCapacity contains elements equal to the given string. Add all the values for the  
 703 MaxCapacity property of the selected CIM\_ConfigurationCapacity instances.

704 **10 CIM Elements**

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

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

710 **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		

711 NOTE: Abstract classes are not shown in the tables in the following sections.

## 712 10.1 CIM\_Card

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

715 **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 “.*”).

716 **10.2 CIM\_Chassis**

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

719 **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 “.*”).

720 **10.3 CIM\_Chip**

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

723 **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 “.*”).

724 **10.4 CIM\_ComputerSystemPackage**

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

727 **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” .

728 **10.5 CIM\_ConfigurationCapacity**

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

731 **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.

732 **10.6 CIM\_ConnectedTo**

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

736 **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)



737 **10.7 CIM\_Container**

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

741 **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)

742 **10.8 CIM\_ElementCapabilities**

743 CIM\_ElementCapabilities associates Physical Elements with the CIM\_PhysicalAssetCapabilities  
 744 instances that advertise the physical capabilities. CIM\_ElementCapabilities shall be instantiated when an  
 745 instance of CIM\_PhysicalAssetCapabilities exists. Table 18 contains the requirements for properties of  
 746 the instance.

747 **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)

748 **10.9 CIM\_ElementCapacity**

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

751 **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)

752 **10.10 CIM\_ElementInConnector**

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

756 **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)

757 **10.11 CIM\_PhysicalAssetCapabilities**

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

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

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

761 **10.12 CIM\_PhysicalComponent**

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

764 **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 “. *”).

765 **10.13 CIM\_PhysicalConnector**

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

768 **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 “.*”).

769 **10.14 CIM\_PhysicalFrame**

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

772 **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 “.*”).

773 **10.15 CIM\_PhysicalMemory**

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

776 **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 “.*”).

777 **10.16 CIM\_PhysicalPackage**

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

780 **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 “.*”).

781 **10.17 CIM\_Rack**

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

784 **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 “.*”).

785 **10.18 CIM\_Realizes**

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

788 **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

789 **10.19 CIM\_RegisteredProfile**

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

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

793 **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 "1.0.0".
RegisteredOrganization	Mandatory	This property shall have a value of 2 (DMTF).

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

797 **10.20 CIM\_Slot**

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

800 **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 ".*").

801 **10.21 CIM\_SystemPackaging**

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

804 **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)

805

806  
807  
808  
809  
810

## ANNEX A (informative)

### Change Log

Version	Date	Description
1.0.0b	06/28/2006	Preliminary Standard
1.0.0	12/11/2007	Final Standard
1.0.1	06/09/2008	Incorporated errata submitted for the Final Standard.
1.0.2	4/6/2009	DMTF Standard Release Incorporated errata on CIM_PhysicalMemory.Speed property values for unknown or variable speeds.

811  
812