



1

2

3

4

**Document Number: DSP1011**

**Date: 2008-07-22**

**Version: 1.0.1**

5 **Physical Asset Profile**

6 **Document Type: Specification**

7 **Document Status: Final Standard**

8 **Document Language: E**

## 9 Copyright Notice

10 Copyright © 2008 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 for uses consistent with this purpose, provided that correct attribution is given. As DMTF  
14 specifications may be revised from time to time, the particular version and release date should always be  
15 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26 withdrawn or modified after publication, and shall be indemnified and held harmless by any party  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.

# CONTENTS

30	Foreword .....	6
31	Introduction .....	7
32	1 Scope .....	9
33	2 Normative References.....	9
34	2.1 Approved References .....	9
35	2.2 Other References.....	9
36	3 Terms and Definitions .....	9
37	4 Symbols and Abbreviated Terms .....	11
38	5 Synopsis .....	11
39	6 Description .....	12
40	7 Implementation.....	13
41	7.1 Physical Element .....	13
42	7.2 Finding the Scoping Instance of the CIM_System or CIM_ComputerSystem class .....	14
43	7.3 Modeling the Physical Aspects of Logical Representation of Devices .....	15
44	7.4 Support for the Physical Element's FRU Information .....	15
45	7.5 Compatibility of Physical Packages .....	15
46	7.6 Modeling System Chassis .....	16
47	7.7 Modeling Configuration Capacity.....	16
48	7.8 Modeling Physical Hierarchy .....	16
49	8 Methods.....	16
50	8.1 Profile Conventions for Operations.....	16
51	8.2 CIM_Card.....	17
52	8.3 CIM_Chassis.....	17
53	8.4 CIM_Chip .....	17
54	8.5 CIM_ComputerSystemPackage .....	17
55	8.6 CIM_ConfigurationCapacity .....	17
56	8.7 CIM_ConnectedTo.....	18
57	8.8 CIM_Container.....	18
58	8.9 CIM_ElementCapabilities .....	18
59	8.10 CIM_ElementCapacity .....	19
60	8.11 CIM_ElementInConnector .....	19
61	8.12 CIM_PhysicalAssetCapabilities .....	19
62	8.13 CIM_PhysicalComponent .....	19
63	8.14 CIM_PhysicalConnector .....	20
64	8.15 CIM_PhysicalFrame.....	20
65	8.16 CIM_PhysicalMemory .....	20
66	8.17 CIM_PhysicalPackage .....	20
67	8.18 CIM_Rack .....	20
68	8.19 CIM_Realizes.....	20
69	8.20 CIM_Slot .....	20
70	8.21 CIM_SystemPackaging.....	20
71	9 Use Cases .....	21
72	9.1 System Chassis FRU Information.....	21
73	9.2 Fan Package FRU Information .....	21
74	9.3 Finding the Scoping Instance for a Fan Package.....	22
75	9.4 Physical Topology and Finding the Scoping Instance .....	23
76	9.5 Physical Topology.....	24
77	9.6 Physical Memory.....	25
78	9.7 Representing Configuration Capacity .....	26
79	9.8 Representing Physical Connector .....	28
80	9.9 Determining the Part Number of a Failing Component.....	28
81	9.10 Obtaining the Physical Inventory for All Devices within a System.....	28

82	9.11	Obtaining the Physical Inventory for a System Chassis .....	29
83	9.12	Determining Whether the Slot Is Empty.....	29
84	9.13	Retrieving the Fan Capacity for the Chassis .....	29
85	9.14	Retrieving the Maximum Capacity of the Type of Fan Package within the Chassis .....	29
86	10	CIM Elements .....	29
87	10.1	CIM_Card.....	31
88	10.2	CIM_Chassis.....	32
89	10.3	CIM_Chip .....	32
90	10.4	CIM_ComputerSystemPackage .....	33
91	10.5	CIM_ConfigurationCapacity .....	33
92	10.6	CIM_ConnectedTo.....	33
93	10.7	CIM_Container.....	34
94	10.8	CIM_ElementCapabilities .....	34
95	10.9	CIM_ElementCapacity .....	34
96	10.10	CIM_ElementInConnector .....	35
97	10.11	CIM_PhysicalAssetCapabilities .....	35
98	10.12	CIM_PhysicalComponent .....	35
99	10.13	CIM_PhysicalConnector .....	36
100	10.14	CIM_PhysicalFrame.....	36
101	10.15	CIM_PhysicalMemory .....	37
102	10.16	CIM_PhysicalPackage .....	37
103	10.17	CIM_Rack .....	38
104	10.18	CIM_Realizes.....	38
105	10.19	CIM_RegisteredProfile.....	39
106	10.20	CIM_Slot .....	39
107	10.21	CIM_SystemPackaging.....	40
108	ANNEX 1 (informative)	Change Log .....	41
109	ANNEX 1 (informative)	Acknowledgments .....	42

110

## 111 Figures

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

122

123 **Tables**

124	Table 1 – Referenced Profiles .....	11
125	Table 2 – Operations: CIM_ComputerSystemPackage.....	17
126	Table 3 – Operations: CIM_ConnectedTo .....	18
127	Table 4 – Operations: CIM_Container .....	18
128	Table 5 – Operations: CIM_ElementCapabilities .....	18
129	Table 6 – Operations: CIM_ElementCapacity.....	19
130	Table 7 – Operations: CIM_ElementInConnector .....	19
131	Table 8 – Operations: CIM_Realizes .....	20
132	Table 9 – Operations: CIM_SystemPackaging.....	20
133	Table 10 – CIM Elements: Physical Asset Profile.....	30
134	Table 11 – Class: CIM_Card.....	31
135	Table 12 – Class: CIM_Chassis.....	32
136	Table 13 – Class: CIM_Chip .....	32
137	Table 14 – Class: CIM_ComputerSystemPackage .....	33
138	Table 15 – Class: CIM_ConfigurationCapacity .....	33
139	Table 16 – Class: CIM_ConnectedTo.....	33
140	Table 17 – Class: CIM_Container.....	34
141	Table 18 – Class: CIM_ElementCapabilities.....	34
142	Table 19 – Class: CIM_ElementCapacity .....	34
143	Table 20 – Class: CIM_ElementInConnector.....	35
144	Table 21 – Class: CIM_PhysicalAssetCapabilities .....	35
145	Table 22 – Class: CIM_PhysicalComponent .....	35
146	Table 23 – Class: CIM_PhysicalConnector .....	36
147	Table 24 – Class: CIM_PhysicalFrame.....	36
148	Table 25 – Class: CIM_PhysicalMemory .....	37
149	Table 26 – Class: CIM_PhysicalPackage .....	37
150	Table 27 – Class: CIM_Rack .....	38
151	Table 28 – Class: CIM_Realizes.....	38
152	Table 29 – Class: CIM_RegisteredProfile.....	39
153	Table 30 – Class: CIM_Slot .....	39
154	Table 31 – Class: CIM_SystemPackaging.....	40

155

## Foreword

156 The *Physical Asset Profile* (DSP1011) was prepared by the Server Management Working Group.

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

159

## Introduction

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

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





170

# Physical Asset Profile

## 171 1 Scope

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

## 176 2 Normative References

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

### 180 2.1 Approved References

181 DMTF [DSP0200](#), *CIM Operations over HTTP 1.2.0*

182 DMTF [DSP0004](#), *CIM Infrastructure Specification 2.3.0*

183 DMTF [DSP1000](#), *Management Profile Specification Template 1.0.0*

184 DMTF [DSP1001](#), *Management Profile Specification Usage Guide 1.0.0*

185 DMTF [DSP1033](#), *Profile Registration Profile 1.0.0*

### 186 2.2 Other References

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

189 UML Specifications, [http://www.omg.org/technology/documents/modeling\\_spec\\_catalog.htm#UML](http://www.omg.org/technology/documents/modeling_spec_catalog.htm#UML)

190 *Unified Modeling Language (UML) from the Open Management Group (OMG)*, <http://www.omg.org/uml/>

## 191 3 Terms and Definitions

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

### 194 3.1

#### 195 **can**

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

### 197 3.2

#### 198 **cannot**

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

### 200 3.3

#### 201 **conditional**

202 indicates requirements to be followed strictly in order to conform to the document when the specified  
203 conditions are met

- 204 **3.4**  
205 **mandatory**  
206 indicates requirements to be followed strictly in order to conform to the document and from which no  
207 deviation is permitted
- 208 **3.5**  
209 **may**  
210 indicates a course of action permissible within the limits of the document
- 211 **3.6**  
212 **need not**  
213 indicates a course of action permissible within the limits of the document
- 214 **3.7**  
215 **optional**  
216 indicates a course of action permissible within the limits of the document
- 217 **3.8**  
218 **referencing profile**  
219 indicates a profile that owns the definition of this class and can include a reference to this profile in its  
220 "Referenced Profiles" table
- 221 **3.9**  
222 **shall**  
223 indicates requirements to be followed strictly in order to conform to the document and from which no  
224 deviation is permitted
- 225 **3.10**  
226 **shall not**  
227 indicates requirements to be followed in order to conform to the document and from which no deviation is  
228 permitted
- 229 **3.11**  
230 **should**  
231 indicates that among several possibilities, one is recommended as particularly suitable, without  
232 mentioning or excluding others, or that a certain course of action is preferred but not necessarily required
- 233 **3.12**  
234 **should not**  
235 indicates that a certain possibility or course of action is deprecated but not prohibited
- 236 **3.13**  
237 **unspecified**  
238 indicates that this profile does not define any constraints for the referenced CIM element or operation
- 239 **3.14**  
240 **Delimited Substring**  
241 a substring element of the VendorCompatibilityStrings property of a Physical Element or an instance of  
242 CIM\_ConfigurationCapacity. The substring starts at the beginning of the string (representing an element  
243 in the array of the VendorCompatibilityStrings property) and terminates at the end of the string, or at a  
244 character that precedes a colon (:).

245 **3.15**  
 246 **Physical Element**  
 247 an instance of a CIM\_PhysicalElement subclass (such as CIM\_PhysicalConnector, CIM\_Slot,  
 248 CIM\_PhysicalComponent, CIM\_Chip, CIM\_PhysicalMemory, CIM\_PhysicalPackage,  
 249 CIM\_PhysicalFrame, CIM\_Chassis, CIM\_Rack, and CIM\_Card) that represents a physical element

250 **3.16**  
 251 **Physical Package**  
 252 an instance of a CIM\_PhysicalPackage or CIM\_PhysicalPackage subclass (such as CIM\_PhysicalFrame,  
 253 CIM\_Chassis, CIM\_Rack, and CIM\_Card) or CIM\_PhysicalComponent or CIM\_PhysicalComponent  
 254 subclass (such as CIM\_Chip or CIM\_PhysicalMemory) that represents a package

255 **3.17**  
 256 **System Chassis**  
 257 an instance of the CIM\_PhysicalElement or CIM\_Chassis that is associated to an instance of  
 258 CIM\_System or CIM\_ComputerSystem through the CIM\_SystemPackaging or  
 259 CIM\_ComputerSystemPackage association, representing the physical package of the managed system.

260 **4 Symbols and Abbreviated Terms**

261 **4.1**  
 262 **CIM**  
 263 Common Information Model

264 **4.2**  
 265 **FRU**  
 266 Field Replaceable Unit

267 **5 Synopsis**

268 **Profile Name:** *Physical Asset*

269 **Version:** 1.0.1

270 **Organization:** DMTF

271 **CIM Schema version:** 2.18.0

272 **Central Class:** CIM\_PhysicalElement

273 **Scoping Class:** CIM\_ManagedSystemElement

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

277 **Table 1 – Referenced Profiles**

Profile Name	Organization	Version	Relationship	Behavior
<i>Profile Registration</i>	DMTF	1.0.0	Mandatory	

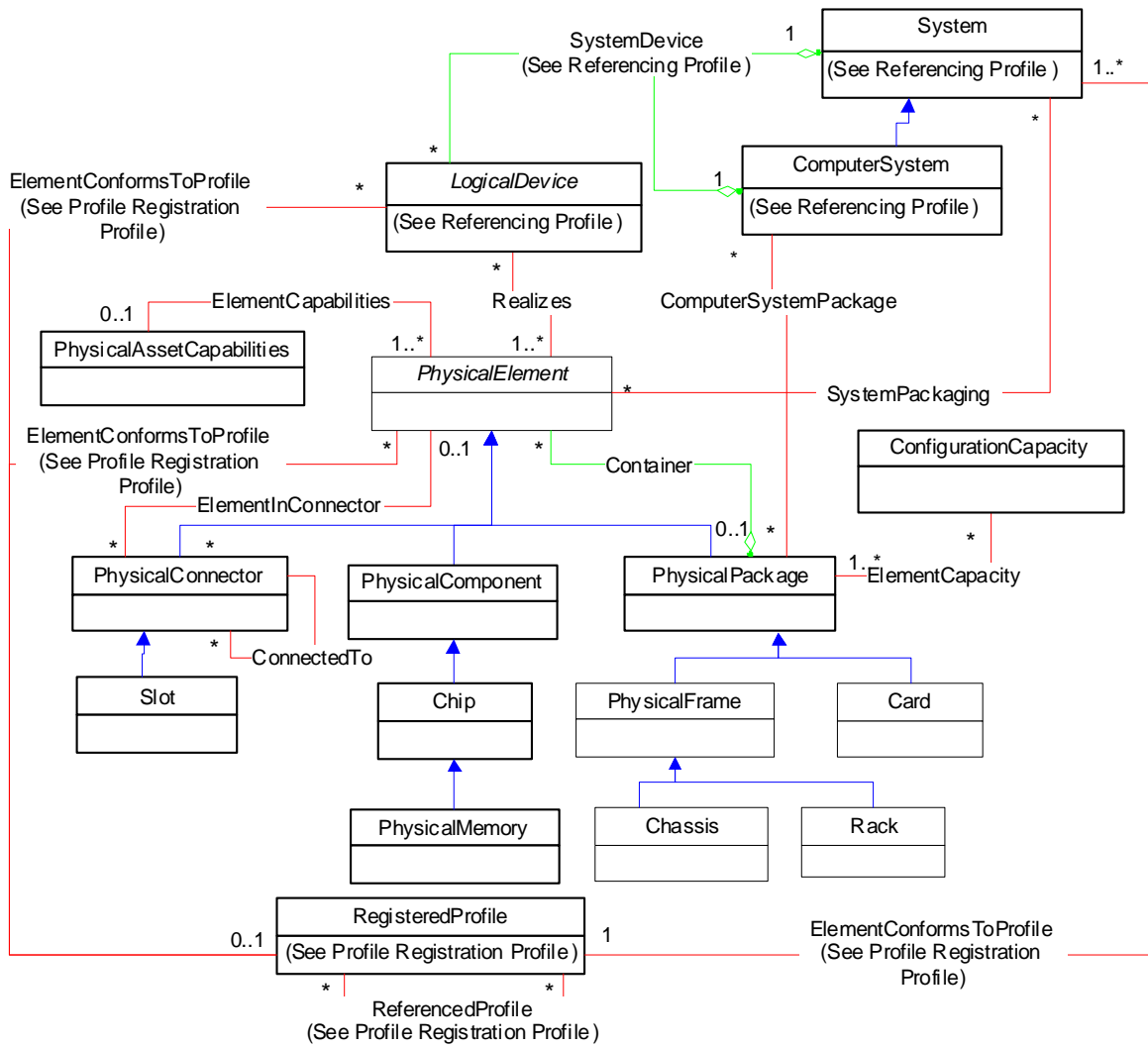
278 The Central Instance for the *Physical Asset Profile* shall be the instance of the CIM\_PhysicalElement  
 279 subclass.

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

284 **6 Description**

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

288 Figure 1 shows the CIM classes that are used in this profile. (For simplicity, the prefix CIM\_ has been  
 289 removed from the names of the classes.) A Physical Element (see section 3.15) describes the physical  
 290 properties, including the FRU information, of a managed element. The capabilities of the Physical  
 291 Elements are described by the properties of the CIM\_PhysicalAssetCapabilities class. The Physical  
 292 Elements could be associated to the logical representation of the managed element through the  
 293 CIM\_Realizes association. The enclosures or chassis of the managed systems are represented by a  
 294 CIM\_PhysicalElement or CIM\_Chassis instance that is associated to the  
 295 CIM\_System/CIM\_ComputerSystem instance through the  
 296 CIM\_SystemPackaging/CIM\_ComputerSystemPackage association and are referred to as a System  
 297 Chassis (see section 3.17). Configuration capacity of the System Chassis is also represented within this  
 298 profile by CIM\_ConfigurationCapacity instances.



299

300

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

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

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

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

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

- 312 • the physical aspects of a managed system, such as the FRU information for the chassis (see  
313 section 7.6)
- 314 • the physical aspects of a specific managed element, such as the FRU information of a fan (see  
315 section 7.3)
- 316 • the physical hierarchy of a managed system, such as the relationship between chassis, slots, and  
317 packages (see section 7.8)
- 318 • the configuration capacity of a managed element, such as the minimum and maximum number of  
319 certain types of packages that the managed system can handle (see section 7.7)

## 320 7 Implementation

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

### 323 7.1 Physical Element

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

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

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

## 332 7.2 Finding the Scoping Instance of the CIM\_System or CIM\_ComputerSystem 333 class

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

336 1) If the selected instance is of a Physical Package, proceed as follows:

337 1) If the Physical Package is associated to the CIM\_LogicalDevice through the CIM\_Realizes  
338 association or to the CIM\_System/CIM\_ComputerSystem through the  
339 CIM\_SystemPackaging/CIM\_ComputerSystemPackage association, the Scoping Instance of  
340 the *Physical Asset Profile* shall be either of the following:

- 341 1) the Scoping Instance of the CIM\_LogicalDevice instance that is associated to the  
342 Physical Package through the instance of CIM\_Realizes
- 343 2) the Scoping Instance of CIM\_System/CIM\_ComputerSystem instance that is  
344 associated to the Physical Package through the instance of CIM\_SystemPackaging or  
345 CIM\_ComputerSystemPackage

346

347 2) If the Physical Package is not associated to the CIM\_LogicalDevice through the CIM\_Realizes  
348 association or to the CIM\_System or CIM\_ComputerSystem through the  
349 CIM\_SystemPackaging or CIM\_ComputerSystemPackage association, proceed as follows:

350 1) If the Physical Package is the Dependent or PartComponent reference in  
351 CIM\_ElementInConnector or CIM\_Container associations, respectively, choose one of the  
352 following paths:

353 1) If the Antecedent or GroupComponent reference of the association is a Physical  
354 Package, select the Antecedent or GroupComponent referenced instance, and go to  
355 1)1).

356 2) Else if the Antecedent or GroupComponent reference of the association is a Physical  
357 Element:

358 1) If the Physical Element is associated to the CIM\_LogicalDevice through the  
359 CIM\_Realizes association, the Scoping Instance of the *Physical Asset Profile*  
360 shall be the Scoping Instance of the CIM\_LogicalDevice instance.

361 2) If the Physical Element instance is not associated to the CIM\_LogicalDevice  
362 through the CIM\_Realizes association:

363 1) If the Physical Element is the PartComponent reference in the  
364 CIM\_Container association:

365 1) If a Physical Package is the GroupComponent reference for the  
366 CIM\_Container association, select the GroupComponent referenced  
367 instance, and go to 1)1).

368 2) If a Physical Element is the GroupComponent or Antecedent reference,  
369 go to 1)2)1)2)1).

370 2) If the Physical Element is not the PartComponent or Dependent reference in  
371 a CIM\_Container association, the Scoping Instance shall be the Central  
372 Instance; thus, the Central Instance is associated to the  
373 CIM\_RegisteredProfile instance.

374 2) Else the Scoping Instance shall be the Central Instance, thus, the Central Instance is  
375 associated to the CIM\_RegisteredProfile instance.

376 2) If the instance is not a Physical Package, go to 1)2)1)2)1).

### 377 **7.3 Modeling the Physical Aspects of Logical Representation of Devices**

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

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

### 383 **7.4 Support for the Physical Element's FRU Information**

384 The Physical Element's support of FRU information shall be advertised by a  
385 CIM\_PhysicalAssetCapabilities instance associated with the Physical Element. At most, one instance of  
386 CIM\_PhysicalAssetCapabilities shall be associated with the Physical Element through the  
387 CIM\_ElementCapabilities association.

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

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

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

- 397 • Manufacturer
- 398 • Model
- 399 • PartNumber
- 400 • SerialNumber
- 401 • SKU

### 402 **7.5 Compatibility of Physical Packages**

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

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

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

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

## 415 7.6 Modeling System Chassis

416 The implementation may instantiate a System Chassis. When a System Chassis is instantiated, the  
417 System Chassis shall be associated with the instance of CIM\_System through the instance of  
418 CIM\_SystemPackaging, or with the instance of CIM\_ComputerSystem through the instance of  
419 CIM\_ComputerSystemPackage.

## 420 7.7 Modeling Configuration Capacity

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

424 When a System Chassis is present, the instrumentation shall associate all the instances of  
425 CIM\_ConfigurationCapacity to the System Chassis through the instances of CIM\_ElementCapacity.  
426 Additionally, when the configuration capacity is for a particular physical package represented by a  
427 Physical Package, the instrumentation may associate the Physical Package with the  
428 CIM\_ConfigurationCapacity through an instance of CIM\_ElementCapacity.

429 When instances of CIM Slot are instantiated, for each unique value of the  
430 CIM\_Slot.VendorCompatibilityStrings, an instance of CIM\_ConfigurationCapacity with an equal value for  
431 the CIM\_ConfigurationCapacity.VendorCompatibilityStrings property shall exist. Additional instances of  
432 CIM\_ConfigurationCapacity may exist.

433 When CIM\_Slot instances are not instantiated or the CIM\_Slot.VendorCompatibilityStrings property is not  
434 instrumented, the CIM\_ConfigurationCapacityVendorCompatibilityStrings array property shall contain an  
435 element with a Delimited Substring that is equal to a string of one of the elements from the  
436 VendorCompatibilityStrings array property of a Physical Package that can be part of the configuration.

## 437 7.8 Modeling Physical Hierarchy

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

- 440 • When a physical element resides within a package, the Physical Element shall be associated with  
441 the Physical Package through the CIM\_Container association.
- 442 • When a package is plugged or connected to a slot or connector, the Physical Package shall be  
443 associated with the CIM\_PhysicalConnector or CIM\_Slot instance through the  
444 CIM\_ElementInConnector association.
- 445 • When physical connectors or slots are connected, the CIM\_PhysicalConnector or CIM\_Slot  
446 instances shall be associated through the CIM\_ConnectedTo association.

## 447 8 Methods

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

### 450 8.1 Profile Conventions for Operations

451 Support for operations for each profile class (including associations) is specified in the following  
452 subclauses. Each of these subclauses includes either a the statement “All operations in the default list in  
453 section 8.1 are supported as described by [DSP0200 version 1.2](#)” or a table listing all of the operations  
454 that are not supported by this profile or where the profile requires behavior other than that described by  
455 [DSP0200 version 1.2](#).



456 The default list of operations is as follows:

- 457 • GetInstance
- 458 • Associators
- 459 • AssociatorNames
- 460 • References
- 461 • ReferenceNames
- 462 • EnumerateInstances
- 463 • EnumerateInstanceNames

464 A compliant implementation shall support all of the operations in the default list for each class, unless the  
 465 “Requirement” column states something other than *Mandatory*.

466 **8.2 CIM\_Card**

467 All operations in the default list in section 8.1 are supported as described by [DSP0200 version 1.2](#).

468 **8.3 CIM\_Chassis**

469 All operations in the default list in section 8.1 are supported as described by [DSP0200 version 1.2](#).

470 **8.4 CIM\_Chip**

471 All operations in the default list in section 8.1 are supported as described by [DSP0200 version 1.2](#).

472 **8.5 CIM\_ComputerSystemPackage**

473 Table 2 lists operations that either have special requirements beyond those from [DSP0200 version 1.2](#) or  
 474 shall not be supported.

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

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

476 **8.6 CIM\_ConfigurationCapacity**

477 All operations in the default list in section 8.1 are supported as described by [DSP0200 version 1.2](#).

478 **8.7 CIM\_ConnectedTo**

479 Table 3 lists operations that either have special requirements beyond those from [DSP0200 version 1.2](#) or  
 480 shall not be supported.

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

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

482 **8.8 CIM\_Container**

483 Table 4 lists operations that either have special requirements beyond those from [DSP0200 version 1.2](#) or  
 484 shall not be supported.

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

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

486 **8.9 CIM\_ElementCapabilities**

487 Table 5 lists operations that either have special requirements beyond those from [DSP0200 version 1.2](#) or  
 488 shall not be supported.

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

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

## 490 8.10 CIM\_ElementCapacity

491 Table 6 lists operations that either have special requirements beyond those from [DSP0200 version 1.2](#) or  
492 shall not be supported.

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

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

## 494 8.11 CIM\_ElementInConnector

495 Table 7 lists operations that either have special requirements beyond those from [DSP0200 version 1.2](#) or  
496 shall not be supported.

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

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

## 498 8.12 CIM\_PhysicalAssetCapabilities

499 All operations in the default list in section 8.1 are supported as described by [DSP0200 version 1.2](#).

## 500 8.13 CIM\_PhysicalComponent

501 All operations in the default list in section 8.1 are supported as described by [DSP0200 version 1.2](#).

## 502 8.14 CIM\_PhysicalConnector

503 All operations in the default list in section 8.1 are supported as described by [DSP0200 version 1.2](#).

## 504 8.15 CIM\_PhysicalFrame

505 All operations in the default list in section 8.1 are supported as described by [DSP0200 version 1.2](#).

## 506 8.16 CIM\_PhysicalMemory

507 All operations in the default list in section 8.1 are supported as described by [DSP0200 version 1.2](#).

## 508 8.17 CIM\_PhysicalPackage

509 All operations in the default list in section 8.1 are supported as described by [DSP0200 version 1.2](#).

## 510 8.18 CIM\_Rack

511 All operations in the default list in section 8.1 are supported as described by [DSP0200 version 1.2](#).

## 512 8.19 CIM\_Realizes

513 Table 8 lists operations that either have special requirements beyond those from [DSP0200 version 1.2](#) or  
514 shall not be supported.

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

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

## 516 8.20 CIM\_Slot

517 All operations in the default list in section 8.1 are supported as described by [DSP0200 version 1.2](#).

## 518 8.21 CIM\_SystemPackaging

519 Table 9 lists operations that either have special requirements beyond those from [DSP0200 version 1.2](#) or  
520 shall not be supported.

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

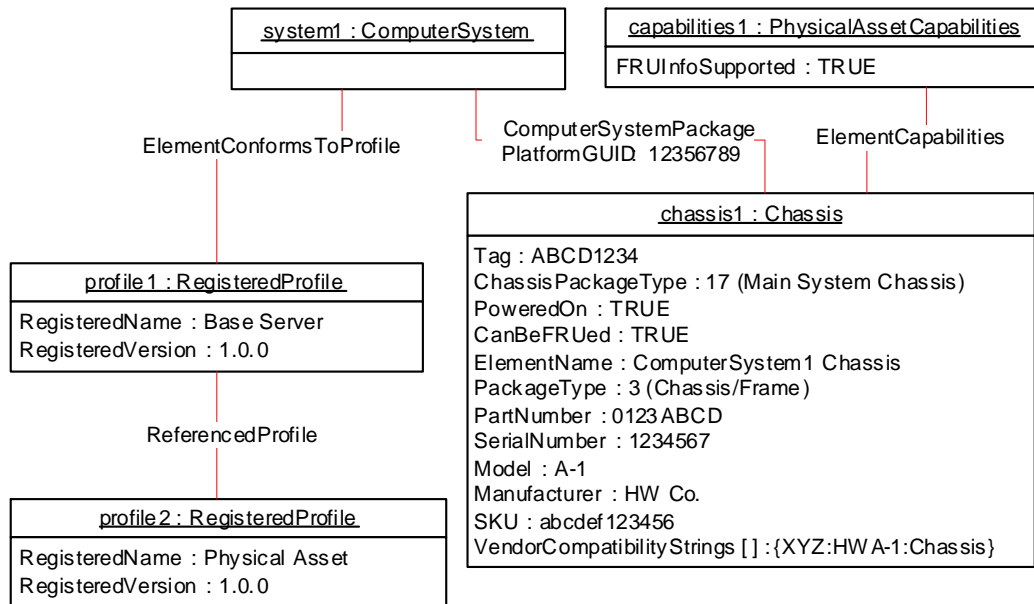
Operation	Requirement	Messages
EnumerateInstanceNames	Unspecified	None
Associators	Unspecified	None
AssociatorNames	Unspecified	None
References	Unspecified	None
ReferenceNames	Unspecified	None

522

523 **9 Use Cases**

524 **9.1 System Chassis FRU Information**

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

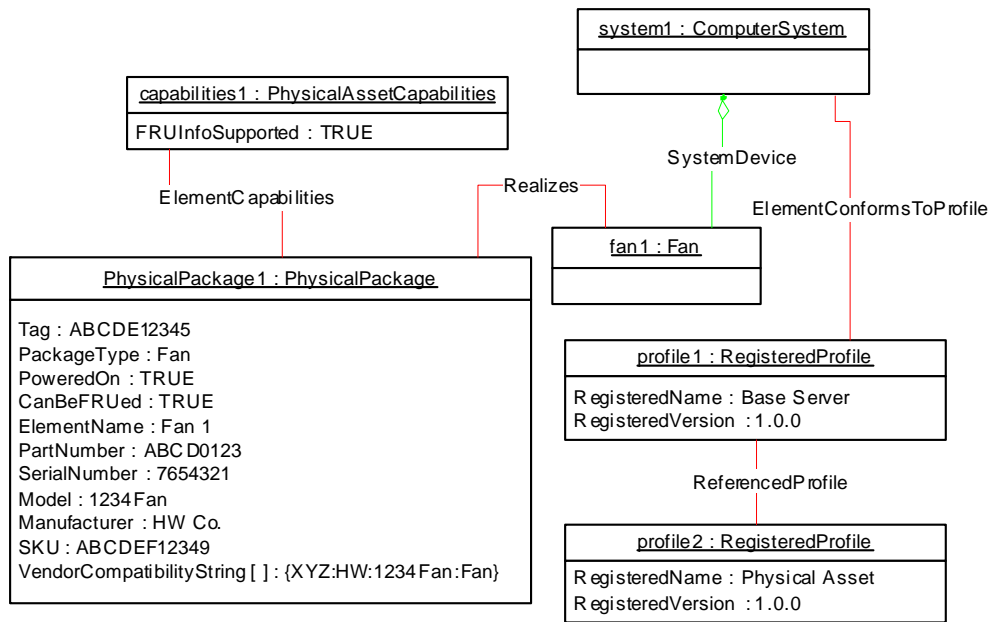


532

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

534 **9.2 Fan Package FRU Information**

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



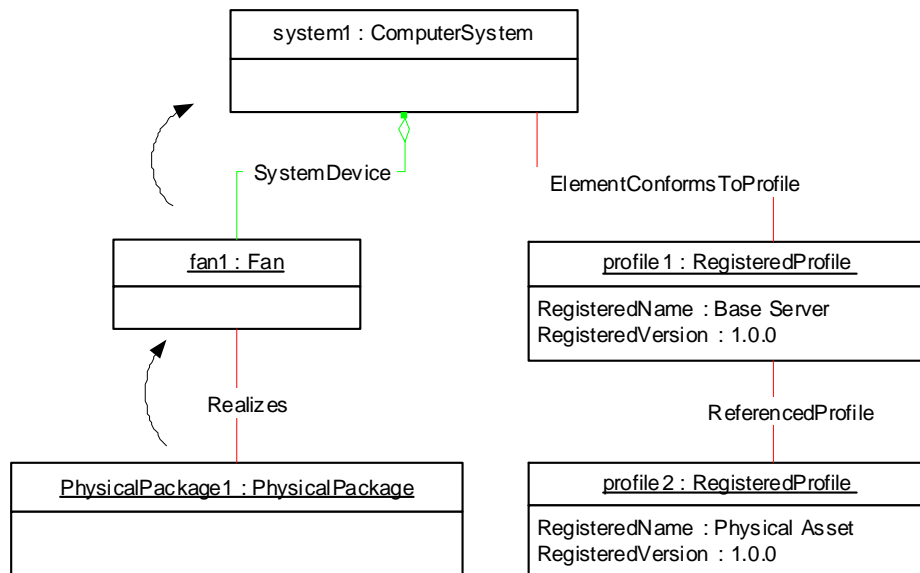
541

542

Figure 3 – CIM\_PhysicalPackage Object Diagram

543 **9.3 Finding the Scoping Instance for a Fan Package**

544 Figure 4 represents another possible instantiation of *Physical Asset Profile*. To find the Scoping Instance  
 545 of PhysicalPackage1, the client needs to select the fan1 associated through the CIM\_Realizes  
 546 association and then find the Scoping Instance for fan1. As defined in the *Fan Profile*, the Scoping  
 547 Instance of fan1 is the CIM\_ComputerSystem instance associated to fan1 through the  
 548 CIM\_SystemDevice association: system1. Thus, system1 is the Scoping Instance of PhysicalPackage1.  
 549 By traversing through the CIM\_ElementConformsToProfile and subsequently the CIM\_ReferencedProfile  
 550 association, the client can find profile2, which advertises the *Physical Asset Profile* information.



551

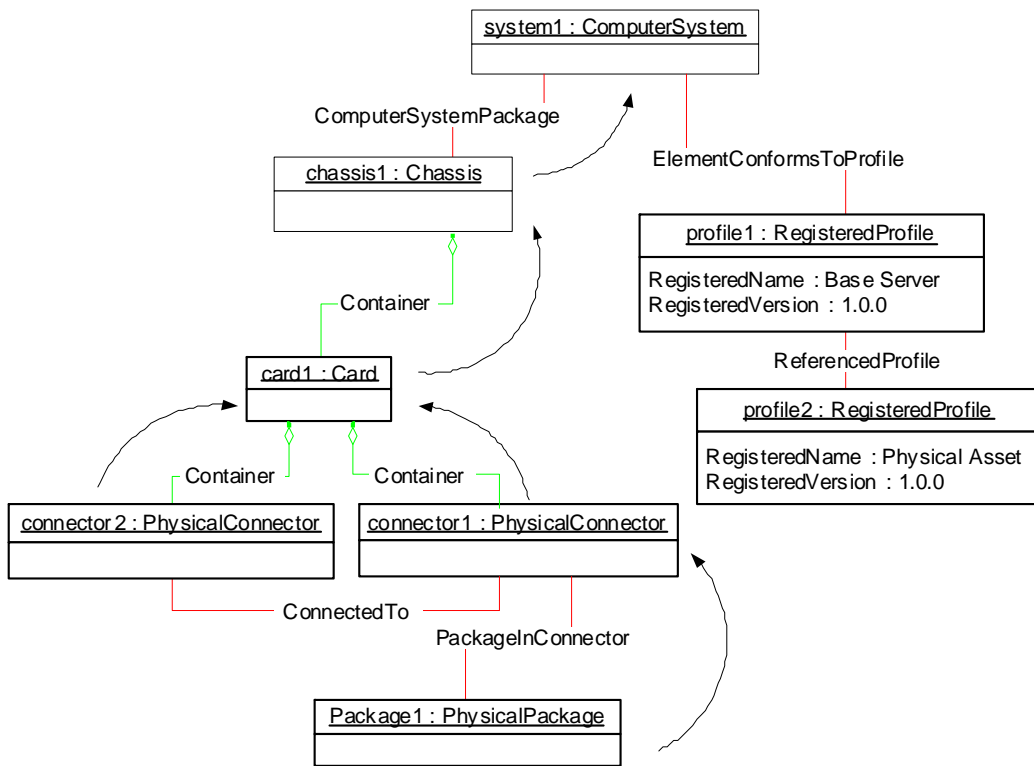
552

Figure 4 – Scoping Instance: Logical Device Object Diagram

553 **9.4 Physical Topology and Finding the Scoping Instance**

554 Figure 5 represents another possible instantiation of the *Physical Asset Profile*. To find the Scoping  
 555 Instance of package1, because package1 is referenced by the CIM\_ElementInConnector.Dependent  
 556 property, the client needs to select connector1, which is referenced by the  
 557 CIM\_ElementInConnector.Antecedent property. Then, because connector1 is referenced by the  
 558 CIM\_Container.PartComponent property, the client needs to select card1, which is referenced by the  
 559 CIM\_Container.GroupComponent. Then, because card1 is referenced by the  
 560 CIM\_Container.PartComponent property, the client needs to select chassis1, which is referenced by the  
 561 CIM\_Container.GroupComponent. Then, because chassis1 is associated to system1 through the  
 562 CIM\_ComputerSystemPackage association, system1 is the Scoping Instance of package1. The client can  
 563 traverse through the CIM\_ElementConformsToProfile and, subsequently, the CIM\_ReferencedProfile  
 564 association, to find profile2, which advertises the *Physical Asset Profile* information.

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

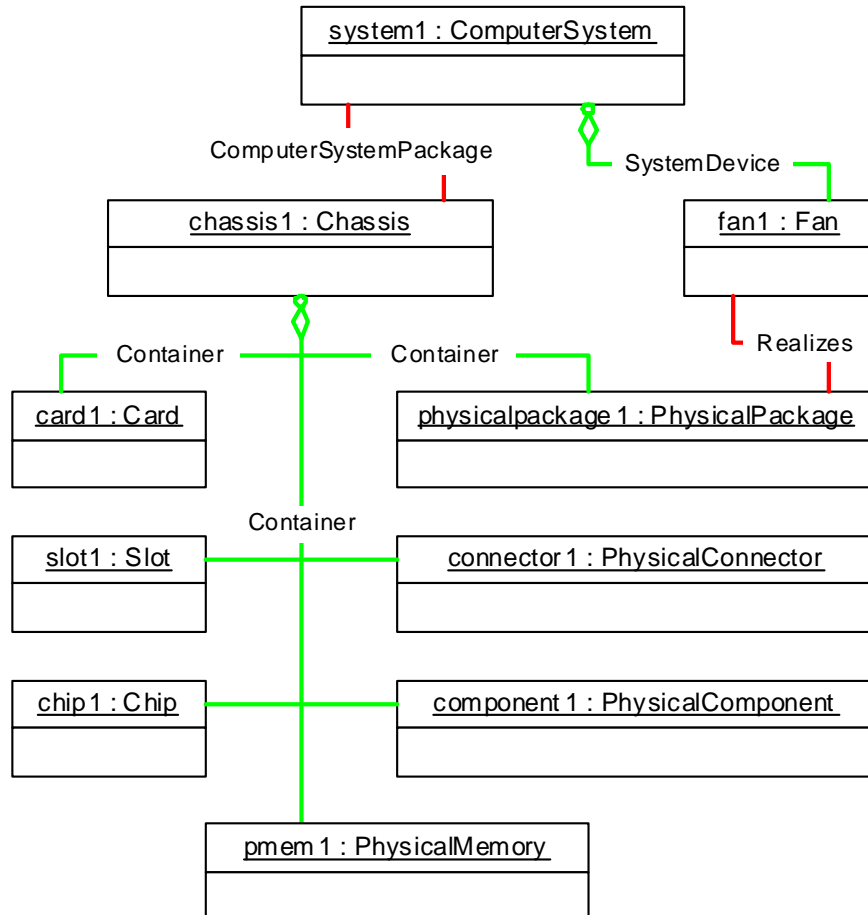


568

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

570 **9.5 Physical Topology**

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



575

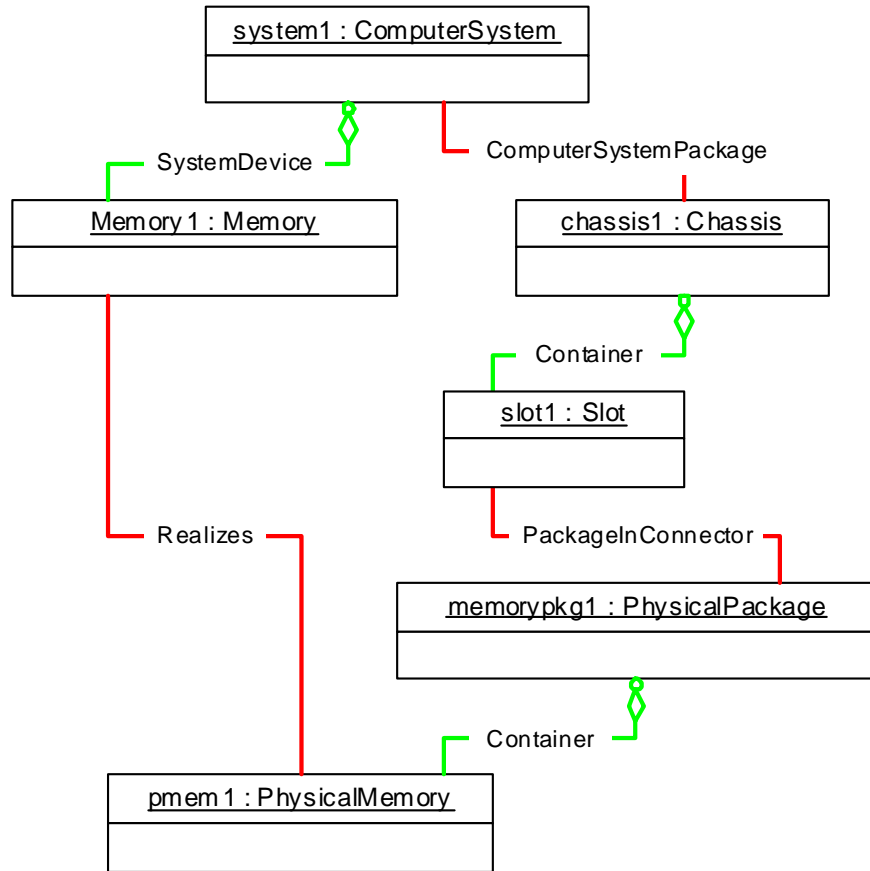
576

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



577 **9.6 Physical Memory**

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



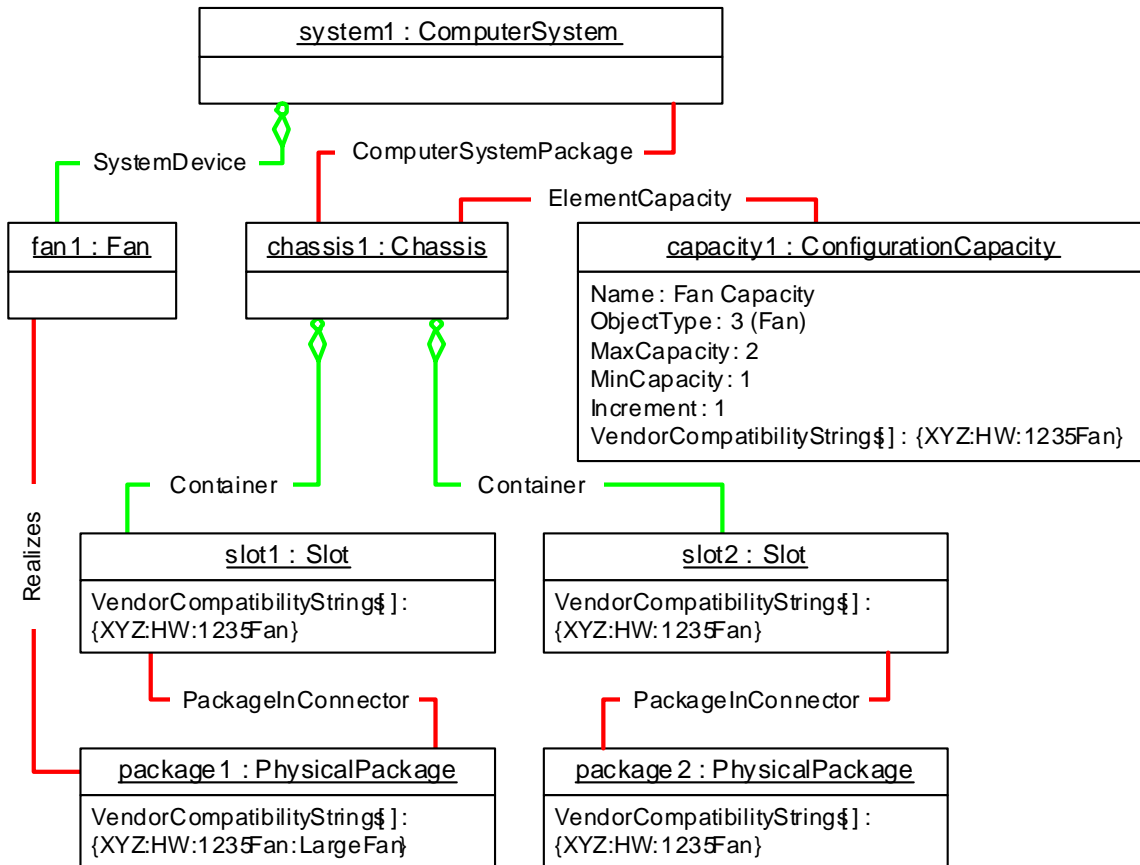
582

583

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

584 **9.7 Representing Configuration Capacity**

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

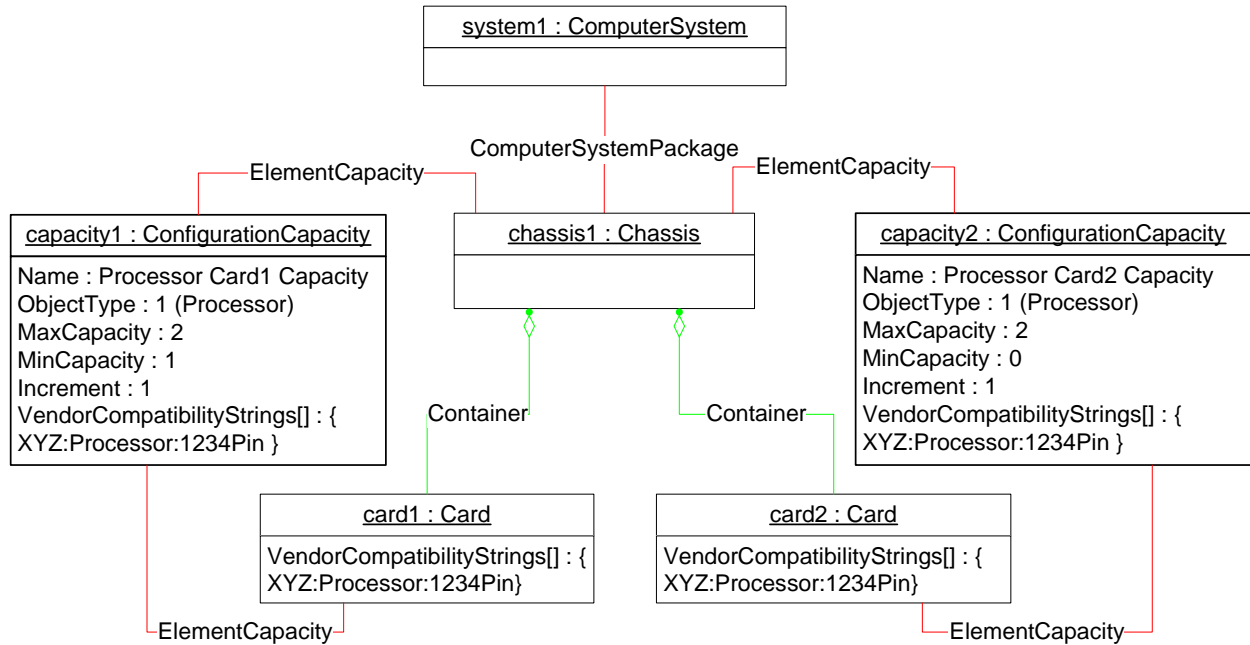


594

595

**Figure 8 – Configuration Capacity Object Diagram**

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



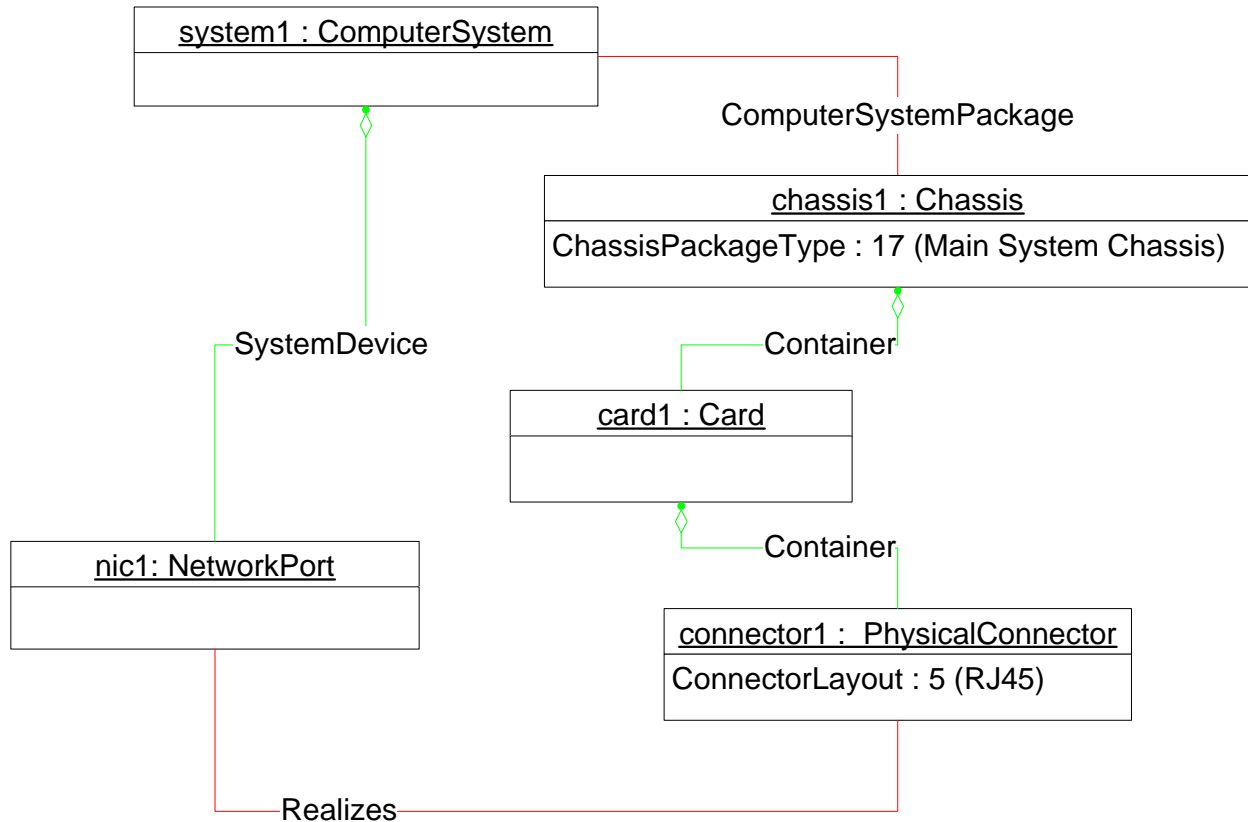
604

605

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

## 606 9.8 Representing Physical Connector

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



610

611

Figure 10 – Network Port Connector Object Diagram

## 612 9.9 Determining the Part Number of a Failing Component

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

## 617 9.10 Obtaining the Physical Inventory for All Devices within a System

618 Select the CIM\_System instance representing the given system. Select all the CIM\_LogicalDevice  
 619 subclass instances that are associated with the CIM\_System instance through the CIM\_SystemDevice  
 620 association, and select all the CIM\_System instances associated through CIM\_SystemComponent  
 621 associations, and then follow the CIM\_SystemDevice association to select all the CIM\_LogicalDevice  
 622 subclass instances. Get all the property values of the CIM\_PhysicalElement subclass instances that are  
 623 associated to the selected CIM\_LogicalDevice subclass instances through the CIM\_Realizes association  
 624 and to the selected CIM\_System instances through the CIM\_SystemPackage association.

### 625 **9.11 Obtaining the Physical Inventory for a System Chassis**

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

### 628 **9.12 Determining Whether the Slot Is Empty**

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

### 633 **9.13 Retrieving the Fan Capacity for the Chassis**

634 For the CIM\_Chassis instance that represents the given chassis, select the associated instances of  
635 CIM\_ConfigurationCapacity through the CIM\_ElementCapacity associations. Select  
636 CIM\_ConfigurationCapacity instances that have the CIM\_ConfigurationCapacity.ObjectType property of 3  
637 (Fan).

### 638 **9.14 Retrieving the Maximum Capacity of the Type of Fan Package within the** 639 **Chassis**

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

642 Select all the instances of CIM\_ConfigurationCapacity associated with the CIM\_Chassis instance through  
643 instances of CIM\_ElementCapacity where the VendorCompatibiltyStrings array property of the instance of  
644 CIM\_ConfigurationCapacity contains elements equal to the given string. Add all the values for the  
645 MaxCapacity property of the selected CIM\_ConfigurationCapacity instances.

## 646 **10 CIM Elements**

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

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

652

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

Element Name	Requirement	Description
<b>Classes</b>		
CIM_Card	Conditional	See sections 7.1 and 10.1.
CIM_Chassis	Conditional	See sections 7.1 and 10.2.
CIM_Chip	Conditional	See sections 7.1 and 10.3.
CIM_ComputerSystemPackage	Conditional	See sections 7.1 and 10.4.
CIM_ConfigurationCapacity	Optional	See sections 7.7 and 10.5.
CIM_ConnectedTo	Optional	See section 10.6.
CIM_Container	Optional	See sections 7.1 and 10.7.
CIM_ElementCapabilities	Conditional	See section 10.8.
CIM_ElementCapacity	Conditional	See sections 7.7 and 10.9.
CIM_ElementInConnector	Optional	See sections 7.1 and 10.10.
CIM_PhysicalAssetCapabilities	Optional	See sections 7.4 and 10.11.
CIM_PhysicalComponent	Conditional	See sections 7.1 and 10.12.
CIM_PhysicalConnector	Conditional	See sections 7.1 and 10.13.
CIM_PhysicalFrame	Conditional	See sections 7.1 and 10.14.
CIM_PhysicalMemory	Conditional	See sections 7.1 and 10.15.
CIM_PhysicalPackage	Conditional	See sections 7.1 and 10.16.
CIM_Rack	Conditional	See sections 7.1 and 10.17.
CIM_Realizes	Conditional	See sections 7.1 and 10.18.
CIM_RegisteredProfile	Mandatory	See section 10.19.
CIM_Slot	Conditional	See sections 7.1 and 10.20.
CIM_SystemPackaging	Conditional	See sections 7.1 and 10.21.
<b>Indications</b>		
None defined in this profile		

653

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

654 **10.1 CIM\_Card**

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

657 **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 section 7.4.
Model	Conditional	See section 7.4.
SerialNumber	Conditional	See section 7.4.
PartNumber	Conditional	See section 7.4.
SKU	Conditional	See section 7.4.
CanBeFRUed	Optional	This property should be implemented when the Physical Element can be replaced in the field.
VendorCompatibilityStrings	Optional	See section 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 “.*”).

658 **10.2 CIM\_Chassis**

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

661 **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 section 7.4.
Model	Conditional	See section 7.4.
SerialNumber	Conditional	See section 7.4.
PartNumber	Conditional	See section 7.4.
SKU	Conditional	See section 7.4.
CanBeFRUed	Optional	This property should be implemented when the Physical Element can be replaced in the field.
VendorCompatibilityStrings	Optional	See section 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 “.*”).

662 **10.3 CIM\_Chip**

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

665 **Table 13 – Class: CIM\_Chip**

Elements	Requirement	Notes
Tag	Mandatory	<b>Key</b>
CreationClassName	Mandatory	<b>Key</b>
Manufacturer	Conditional	See section 7.4.
Model	Conditional	See section 7.4.
SerialNumber	Conditional	See section 7.4.
PartNumber	Conditional	See section 7.4.
SKU	Conditional	See section 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 “.*”).



666 **10.4 CIM\_ComputerSystemPackage**

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

669 **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” .

670 **10.5 CIM\_ConfigurationCapacity**

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

673 **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 section 7.5.

674 **10.6 CIM\_ConnectedTo**

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

678 **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)

679 **10.7 CIM\_Container**

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

683 **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)

684 **10.8 CIM\_ElementCapabilities**

685 CIM\_ElementCapabilities associates Physical Elements with the CIM\_PhysicalAssetCapabilities  
686 instances that advertise the physical capabilities. CIM\_ElementCapabilities shall be instantiated when an  
687 instance of CIM\_PhysicalAssetCapabilities exists. Table 18 contains the requirements for properties of  
688 the instance.

689 **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)

690 **10.9 CIM\_ElementCapacity**

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

693 **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)

694 **10.10 CIM\_ElementInConnector**

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

698 **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)

699 **10.11 CIM\_PhysicalAssetCapabilities**

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

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

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

703 **10.12 CIM\_PhysicalComponent**

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

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

Elements	Requirement	Notes
Tag	Mandatory	<b>Key</b>
CreationClassName	Mandatory	<b>Key</b>
Manufacturer	Conditional	See section 7.4.
Model	Conditional	See section 7.4.
SerialNumber	Conditional	See section 7.4.
PartNumber	Conditional	See section 7.4.
SKU	Conditional	See section 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 “. *”).

707 **10.13 CIM\_PhysicalConnector**

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

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

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

711 **10.14 CIM\_PhysicalFrame**

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

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

Elements	Requirement	Notes
Tag	Mandatory	<b>Key</b>
CreationClassName	Mandatory	<b>Key</b>
PackageType	Mandatory	None
Manufacturer	Conditional	See section 7.4.
Model	Conditional	See section 7.4.
SerialNumber	Conditional	See section 7.4.
PartNumber	Conditional	See section 7.4.
SKU	Conditional	See section 7.4.
VendorCompatibilityStrings	Optional	See section 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 “.*”).

715 **10.15 CIM\_PhysicalMemory**

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

718 **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 section 7.4.
Model	Conditional	See section 7.4.
SerialNumber	Conditional	See section 7.4.
PartNumber	Conditional	See section 7.4.
SKU	Conditional	See section 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 “.*”).

719 **10.16 CIM\_PhysicalPackage**

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

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

Elements	Requirement	Notes
Tag	Mandatory	<b>Key</b>
CreationClassName	Mandatory	<b>Key</b>
PackageType	Mandatory	None
Manufacturer	Conditional	See section 7.4.
Model	Conditional	See section 7.4.
SerialNumber	Conditional	See section 7.4.
PartNumber	Conditional	See section 7.4.
SKU	Conditional	See section 7.4.
VendorCompatibilityStrings	Optional	See section 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 “.*”).

723 **10.17 CIM\_Rack**

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

726 **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 section 7.4.
Model	Conditional	See section 7.4.
SerialNumber	Conditional	See section 7.4.
PartNumber	Conditional	See section 7.4.
SKU	Conditional	See section 7.4.
VendorCompatibilityStrings	Optional	See section 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 “. *”).

727 **10.18 CIM\_Realizes**

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

730 **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

731 **10.19 CIM\_RegisteredProfile**

732 The CIM\_RegisteredProfile class is defined by the *Profile Registration Profile*. Table 29 contains the  
 733 requirements for properties of the class.

734 The requirements listed in Table 29 are in addition to those mandated by the *Profile Registration Profile*.

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

Elements	Requirement	Description
RegisteredName	Mandatory	This property shall have a value of “ <i>Physical Asset</i> ”.
RegisteredVersion	Mandatory	This property shall have a value of “1.0.0”.
RegisteredOrganization	Mandatory	This property shall have a value of 2 (DMTF).

736 NOTE: Previous versions of this document included the suffix ‘Profile’ for the RegisteredName value. If  
 737 implementations querying for the RegisteredName value find the suffix ‘Profile’, they should ignore the suffix, with any  
 738 surrounding white spaces, before any comparison is done with the value as specified in this document.

739 **10.20 CIM\_Slot**

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

742 **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 section 7.4,
Model	Conditional	See section 7.4,
SerialNumber	Conditional	See section 7.4,
PartNumber	Conditional	See section 7.4,
SKU	Conditional	See section 7.4,
VendorCompatibilityStrings	Optional	See section 7.5,
ElementName	Mandatory	This property shall be formatted as a free-form string of variable length (pattern “.*”).

743 **10.21 CIM\_SystemPackaging**

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

746 **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)

747



748  
749  
750  
751  
752

## **ANNEX 1 (informative)**

### **Change Log**

<b>Version</b>	<b>Date</b>	<b>Description</b>
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.

753

## ANNEX 1 (informative)

754  
755  
756  
757  
758

### Acknowledgments

759 The authors wish to acknowledge the following people.

760 **Editors:**

- 761 • Jon Hass – Dell Inc.
- 762 • Khachatur Papanyan – Dell Inc.
- 763 • Jeff Hilland - HP

764 **Contributors:**

- 765 • Jon Hass – Dell
- 766 • Khachatur Papanyan – Dell
- 767 • Jeff Hilland – HP
- 768 • Christina Shaw – HP
- 769 • Aaron Merkin – IBM
- 770 • Jeff Lynch – IBM
- 771 • Arvind Kumar – Intel
- 772 • Perry Vincent – Intel
- 773 • John Leung – Intel

774  
775