



1

2

3

4

Document Number: DSP1026

Date: 2010-06-10

Version: 1.0.1

5 **System Memory Profile**

6 **Document Type: Specification**

7 **Document Status: DMTF Standard**

8 **Document Language: en-US**

9 Copyright Notice

10 Copyright © 2006, 2007, 2010 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25 withdrawn or modified after publication, and shall be indemnified and held harmless by any party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	5
34	Introduction	6
35	1 Scope	7
36	2 Normative References.....	7
37	3 Terms and Definitions	7
38	4 Symbols and Abbreviated Terms	9
39	5 Synopsis.....	9
40	6 Description	9
41	7 Implementation.....	10
42	7.1 CIM_Memory	10
43	7.2 Representation of the System Memory Size	10
44	7.3 CIM_EnabledLogicalElementCapabilities.....	11
45	7.4 Physical Asset Profile	11
46	8 Methods.....	12
47	8.1 Profile Conventions for Operations.....	12
48	8.2 CIM_ElementCapabilities	12
49	8.3 CIM_EnabledLogicalElementCapabilities.....	12
50	8.4 CIM_Memory	12
51	8.5 CIM_SystemDevice	13
52	9 Use Cases.....	13
53	9.1 Object Diagrams	14
54	9.2 Find the System’s Memory Information	15
55	9.3 Find the System’s Physical Memory Information.....	16
56	9.4 Find the Total Physical System Memory Size	16
57	9.5 Find the Total Available System Memory Size	16
58	9.6 Find the Physical System Memory Size per Memory Device	16
59	9.7 Determine Whether ElementName for the Instance of CIM_Memory Is Modifiable.....	16
60	10 CIM Elements.....	17
61	10.1 CIM_ElementCapabilities	17
62	10.2 CIM_EnabledLogicalElementCapabilities.....	17
63	10.3 CIM_Memory	18
64	10.4 CIM_RegisteredProfile.....	18
65	10.5 CIM_SystemDevice	19
66	ANNEX A (informative) Change Log.....	20
67		
68	Figures	
69	Figure 1 – System Memory Profile: Profile Class Diagram.....	10
70	Figure 2 – System Memory Profile: Object Diagram 1	14
71	Figure 3 – System Memory Profile: Object Diagram 2	15
72		

73 **Tables**

74 Table 1 – Referenced Profiles 9

75 Table 2 – Operations: CIM_ElementCapabilities 12

76 Table 3 – Operations: CIM_Memory 13

77 Table 4 – Operations: CIM_SystemDevice 13

78 Table 5 – CIM Elements: System Memory Profile 17

79 Table 6 – Class: CIM_ElementCapabilities 17

80 Table 7 – Class: CIM_EnabledLogicalElementCapabilities 17

81 Table 8 – Class: CIM_Memory 18

82 Table 9 – Class: CIM_RegisteredProfile 18

83 Table 10 – Class: CIM_SystemDevice 19

84

85

Foreword

86 The *System Memory Profile* (DSP1026) was prepared by the Physical Platform Profiles Working Group of
87 the DMTF.

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

90 Acknowledgments

91 The authors wish to acknowledge the following people.

92 Editors:

- 93 • Hemal Shah – Broadcom
- 94 • Khachatur Papanyan – Dell

95 Contributors:

- 96 • Hemal Shah – Broadcom
- 97 • Jon Hass – Dell
- 98 • Khachatur Papanyan – Dell
- 99 • Jeff Hilland – HP
- 100 • Christina Shaw – HP
- 101 • Aaron Merkin – IBM
- 102 • Jeff Lynch – IBM
- 103 • Perry Vincent – Intel
- 104 • John Leung – Intel

105

106

Introduction

107 This document defines the classes used to describe the system memory. Also included are descriptions
108 of association classes that describe the relationship of the system memory with the memory's physical
109 aspects (such as FRU data), with the managed system, and with DMTF profile version information.

110 The information in this specification is intended to be sufficient for a provider or consumer of this data to
111 identify unambiguously the classes, properties, methods, and values that shall be instantiated and
112 manipulated to represent the system memory of managed systems and subsystems that are modeled
113 using the DMTF Common Information Model (CIM) core and extended model definitions.

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

116

System Memory Profile

117 1 Scope

118 The *System Memory Profile* extends the management capabilities of referencing profiles by adding the
119 capability to represent the total memory available to the system. The memory's relationship with the
120 memory's physical aspects, the managed system that uses the memory, and the profile's registration for
121 the schema implementation version information are also described.

122 2 Normative References

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

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

128 DMTF DSP0134, *System Management BIOS Reference Specification 2.6*,
129 http://www.dmtf.org/standards/published_documents/DSP0134_2.6.pdf

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

132 DMTF DSP0215, *Server Management Managed Element Addressing Specification (SM ME Addressing)*
133 *1.0*, http://www.dmtf.org/standards/published_documents/DSP0215_1.0.pdf

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

136 DMTF DSP1011, *Physical Asset Profile 1.0*,
137 http://www.dmtf.org/standards/published_documents/DSP1011_1.0.pdf

138 DMTF DSP1033, *Profile Registration Profile 1.0*,
139 http://www.dmtf.org/standards/published_documents/DSP1033_1.0.pdf

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

142 3 Terms and Definitions

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

145 3.1

146 can

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

148 3.2

149 cannot

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

- 151 **3.3**
152 **conditional**
153 indicates requirements to be followed strictly to conform to the document when the specified conditions
154 are met
- 155 **3.4**
156 **mandatory**
157 indicates requirements to be followed strictly to conform to the document and from which no deviation is
158 permitted
- 159 **3.5**
160 **may**
161 indicates a course of action permissible within the limits of the document
- 162 **3.6**
163 **need not**
164 indicates a course of action permissible within the limits of the document
- 165 **3.7**
166 **optional**
167 indicates a course of action permissible within the limits of the document
- 168 **3.8**
169 **referencing profile**
170 indicates a profile that owns the definition of this class and can include a reference to this profile in its
171 "Referenced Profiles" table
- 172 **3.9**
173 **shall**
174 indicates requirements to be followed strictly to conform to the document and from which no deviation is
175 permitted
- 176 **3.10**
177 **shall not**
178 indicates requirements to be followed strictly to conform to the document and from which no deviation is
179 permitted
- 180 **3.11**
181 **should**
182 indicates that among several possibilities, one is recommended as particularly suitable, without
183 mentioning or excluding others, or that a certain course of action is preferred but not necessarily required
- 184 **3.12**
185 **should not**
186 indicates that a certain possibility or course of action is deprecated but not prohibited
- 187 **3.13**
188 **unspecified**
189 indicates that this profile does not define any constraints for the referenced CIM element or operation

190 **4 Symbols and Abbreviated Terms**

191 The following symbols and abbreviations are used in this document.

192 **4.1**

193 **CIM**

194 Common Information Model

195 **4.2**

196 **FRU**

197 Field Replaceable Unit

198 **5 Synopsis**

199 **Profile Name:** *System Memory*

200 **Version:** 1.0.1

201 **Organization:** DMTF

202 **CIM Schema Version:** 2.10

203 **Central Class:** CIM_Memory

204 **Scoping Class:** CIM_ComputerSystem

205 The *System Memory Profile* extends the management capability of the referencing profiles by adding the
 206 capability to describe the total memory available to a managed system. The *System Memory Profile* is a
 207 component profile.

208 Table 1 identifies profiles that are related to this profile.

209 **Table 1 – Referenced Profiles**

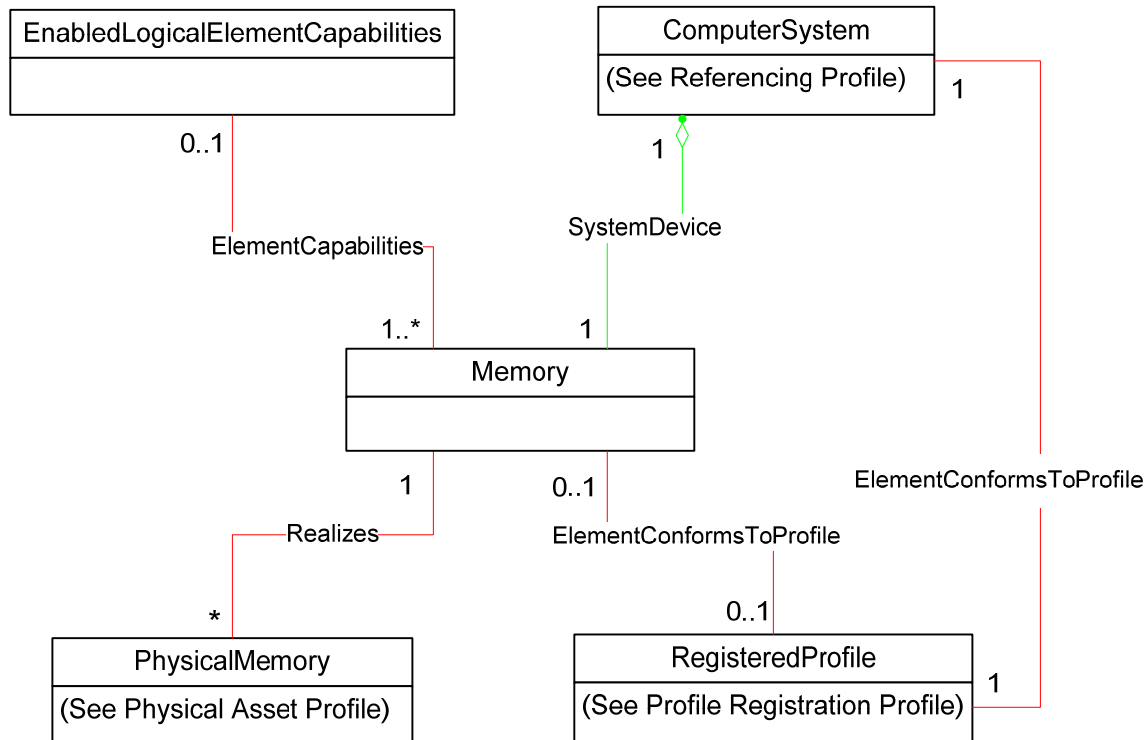
Profile Name	Organization	Version	Relationship	Behavior
Physical Asset	DMTF	1.0	Optional	See 7.4.
Profile Registration	DMTF	1.0	Mandatory	None

210 **6 Description**

211 The *System Memory Profile* extends the management capability of the referencing profiles by adding the
 212 capability to describe memory devices associated with a system. The CIM_Memory class describes the
 213 system’s total memory that includes both available and physical memory, and corresponds to collective
 214 representation of the SMBIOS Type 16, Type 17, Type 19, and Type 20 memory structures (see 7.2). The
 215 CIM_PhysicalMemory class describes the physical aspects of the system’s memory and every instance
 216 corresponds to the individual SMBIOS Type 17 memory structure.

217 Figure 1 represents the class schema for the *System Memory Profile*. For simplicity, the prefix CIM_ has
 218 been removed from the names of the classes.

219 The CIM_Memory class describes the total system memory. The physical aspects of the memory are
 220 described with the CIM_PhysicalMemory class, which is associated with the CIM_Memory class through
 221 the CIM_Realizes association. The ownership of the memory is represented through the
 222 CIM_SystemDevice association to the managed system, which is represented by the
 223 CIM_ComputerSystem class. The DMTF version of the *System Memory Profile* that is implemented is
 224 represented through the CIM_RegisteredProfile class.



225

226

Figure 1 – System Memory Profile: Profile Class Diagram

227 7 Implementation

228 This section details the requirements related to the arrangement of instances and their properties for
 229 implementations of this profile. Methods are listed in section 8 (“Methods”), and properties are listed in
 230 section 10 (“CIM Elements”).

231 7.1 CIM_Memory

232 A single instance of CIM_Memory shall exist in accordance with the profile instantiation and shall be
 233 associated with the CIM_ComputerSystem instance that represents the managed system through an
 234 instance of CIM_SystemDevice. This CIM_Memory instance is the logical representation of multiple
 235 physical memories in a managed system and represents the total memory installed and available to the
 236 system.

237 7.2 Representation of the System Memory Size

238 This section describes the method of total system memory size calculation. Additionally this section
 239 describes the corresponding SMBIOS structures defined in the *System Management BIOS Reference*
 240 *Specification* ([DSP0134](#)) that may be used for the calculation. Note that the underlying represented
 241 system does not need to support [DSP0134](#).

242 7.2.1 Total Available System Memory

243 The ConsumableBlocks property represents the number of total available system memory blocks. When
 244 the number of available system memory blocks is unknown, the value of ConsumableBlocks shall be set
 245 to Null.

246 The total available memory of the system shall be the product of multiplying the value of the
247 CIM_Memory.ConsumableBlocks property by the value of the CIM_Memory.BlockSize property.

248 When [DSP0134](#) is implemented, the product of multiplying the value of the
249 CIM_Memory.ConsumableBlocks property by the value of the CIM_Memory.BlockSize property shall
250 equal the sum of the values of the products of 1 KByte with the subtraction of value of StartingAddress
251 property from the value of EndingAddress property ($\sum [1\text{KB} \times (\text{EndingAddress} - \text{StartingAddress})]$) of
252 either Memory Array Mapped Device (Type 19) structures whose Memory Array Handle field points to a
253 Physical Memory Array (Type 16) structure with Use field set to 03h (System memory) or Memory Device
254 Mapped Address (Type 20) structures whose Memory Device Handle field points to a Memory Device
255 (Type 17) structure that has Physical Memory Array Handle field pointing to a Physical Memory Array
256 (Type 16) structure with Use field set to 03h (System memory).

257 7.2.2 Total Physical System Memory

258 The NumberOfBlocks property represents the number of total physical system memory blocks.

259 The total physical memory of the system shall be the product of multiplying the value of the
260 CIM_Memory.NumberOfBlocks property by the value of the CIM_Memory.BlockSize property.

261 When the [DSP0134](#) is implemented, the product of multiplying the value of the
262 CIM_Memory.NumberOfBlocks property by the value of the CIM_Memory.BlockSize property shall equal
263 the sum of the values of the Size property of Memory Device (Type 17) structure that has Physical
264 Memory Array Handle field pointing to a Physical Memory Array (Type 16) structure with Use field set to
265 03h (System memory).

266 When the optional behavior of modeling the physical aspects of the system memory specified in 7.4 is
267 implemented, the product of multiplying the value of the CIM_Memory.NumberOfBlocks property by the
268 value of the CIM_Memory.BlockSize property shall equal the sum of the values of the
269 CIM_PhysicalMemory.Capacity property for each instance of the CIM_PhysicalMemory to which the
270 CIM_Memory instance is associated through the CIM_Realizes association.

271 7.3 CIM_EnabledLogicalElementCapabilities

272 When the CIM_EnabledLogicalElementCapabilities class is instantiated, the instance of
273 CIM_EnabledLogicalElementCapabilities shall be associated with the CIM_Memory instance through an
274 instance of CIM_ElementCapabilities and used for advertising the capabilities of the CIM_Memory
275 instance.

276 At most one instance of CIM_EnabledLogicalElementCapabilities shall be associated with a given
277 instance of CIM_Memory.

278 7.3.1 CIM_EnabledLogicalElementCapabilities.ElementNameEditSupported

279 The ElementNameEditSupported property shall have a value of TRUE when the implementation supports
280 client modification of the CIM_Memory.ElementName property.

281 7.3.2 CIM_EnabledLogicalElementCapabilities.MaxElementNameLen

282 The MaxElementNameLen property shall be implemented when the ElementNameEditSupported
283 property has a value of TRUE.

284 7.4 Physical Asset Profile

285 The [Physical Asset Profile](#) may be implemented to model the physical aspects of the system memory,
286 including the asset information.

287 When the system memory's physical aspects are represented, each CIM_PhysicalMemory instance(s)
 288 shall be instantiated and associated with the instance of CIM_Memory through instance(s) of
 289 CIM_Realizes.

290 8 Methods

291 This section details the requirements for supporting intrinsic operations for the CIM elements defined by
 292 this profile. No extrinsic methods are defined for this profile.

293 8.1 Profile Conventions for Operations

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

296 The default list of operations is as follows:

- 297 • GetInstance
- 298 • Associators
- 299 • AssociatorNames
- 300 • References
- 301 • ReferenceNames
- 302 • EnumerateInstances
- 303 • EnumerateInstanceNames

304 8.2 CIM_ElementCapabilities

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

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

309 **Table 2 – Operations: CIM_ElementCapabilities**

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

310 8.3 CIM_EnabledLogicalElementCapabilities

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

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

313 8.4 CIM_Memory

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

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

318 **Table 3 – Operations: CIM_Memory**

Operation	Requirement	Messages
ModifyInstance	Conditional. See 8.4.1.	None

319 **8.4.1 CIM_Memory — ModifyInstance**

320 This section details the requirements for the ModifyInstance operation applied to an instance of
 321 CIM_Memory. The ModifyInstance operation may be supported.

322 The ModifyInstance operation shall be supported and CIM_Memory.ElementName shall be modifiable
 323 when an instance of CIM_EnabledLogicalElementCapabilities is associated to the instance of
 324 CIM_Memory, and the ElementNameEditSupported property of the
 325 CIM_EnabledLogicalElementCapabilities instance has a value of TRUE. See 8.4.1.1.

326 **8.4.1.1 CIM_Memory.ElementName**

327 When an instance of CIM_EnabledLogicalElementCapabilities is associated to the instance of
 328 CIM_Memory, and the ElementNameEditSupported property of the
 329 CIM_EnabledLogicalElementCapabilities instance has a value of TRUE, the implementation shall allow
 330 the ModifyInstance operation to change the value of the ElementName property of the CIM_Memory
 331 instance. The ModifyInstance operation shall enforce the length restriction specified in the
 332 MaxElementNameLen property of the CIM_EnabledLogicalElementCapabilities instance.

333 When the ElementNameEditSupported property of the CIM_EnabledLogicalElementCapabilities instance
 334 has a value of FALSE or when there is no instance of CIM_EnabledLogicalElementCapabilities
 335 associated with the instance of CIM_Memory, the implementation shall not allow the ModifyInstance
 336 operation to change the value of the ElementName property of the CIM_Memory instance.

337 **8.5 CIM_SystemDevice**

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

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

342 **Table 4 – Operations: CIM_SystemDevice**

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

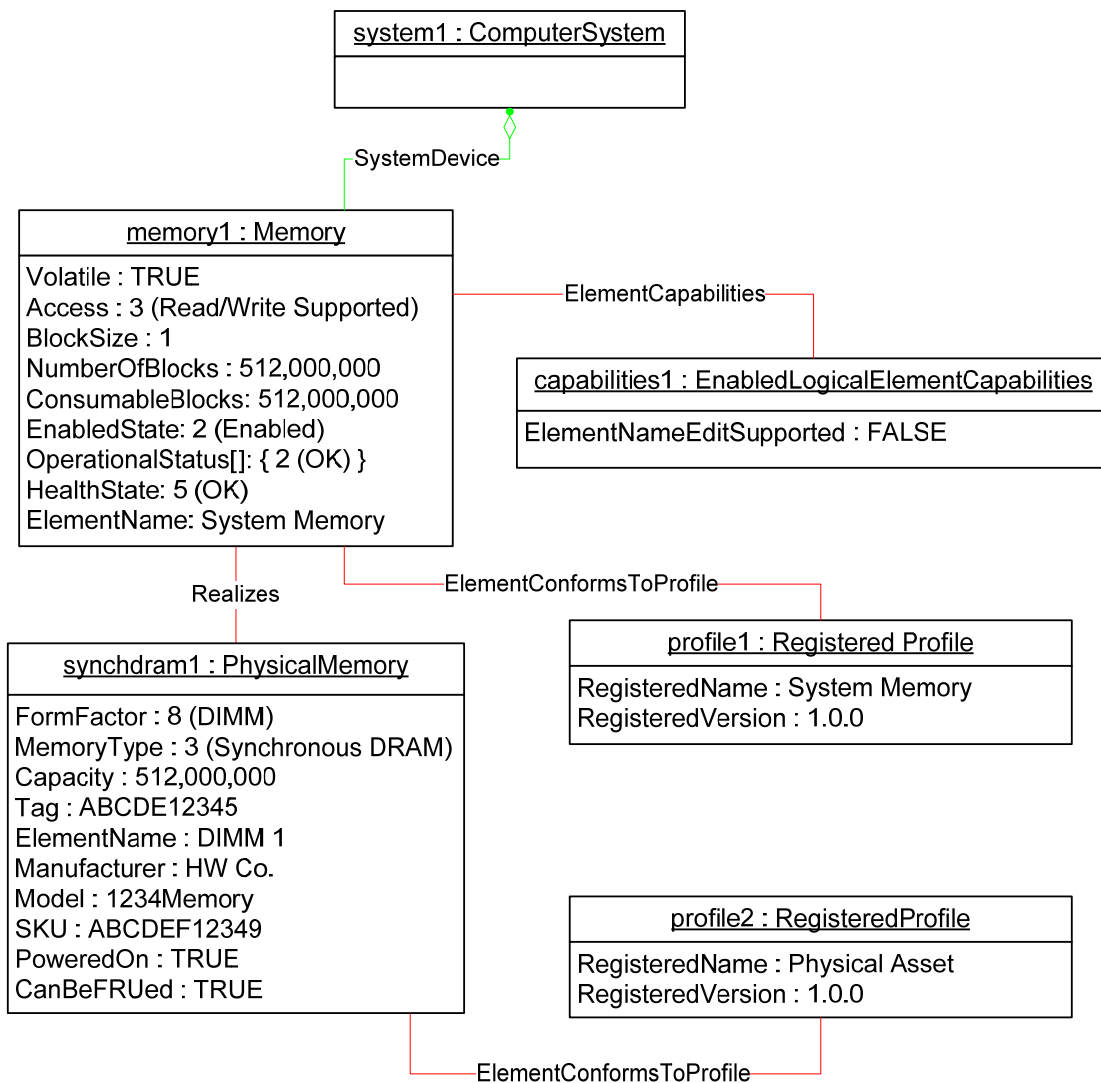
343 **9 Use Cases**

344 This section contains object diagrams and use cases for the *System Memory Profile*.

345 **9.1 Object Diagrams**

346 Figure 2 represents a possible instantiation of the *System Memory Profile*. In this case, the computer
 347 system represented by the system1 instance of CIM_ComputerSystem owns and uses memory
 348 represented by the memory1 instance of CIM_Memory. memory1 contains information such as the
 349 volatility, accessibility, total physical and available size, and operational and health status of the memory.
 350 The property ConsumableBlocks represents the memory blocks available for consumption by the system
 351 and is equal to the total physical system memory blocks represented by the NumberOfBlocks property.
 352 The physical aspects such as memory type, physical size, form factor, and other FRU data are described
 353 by the synchdram1 instance of CIM_PhysicalMemory. profile1 shows the version of the current *System*
 354 *Memory Profile* implementation.

355 For simplicity, the prefix CIM_ has been removed from the names of the classes.



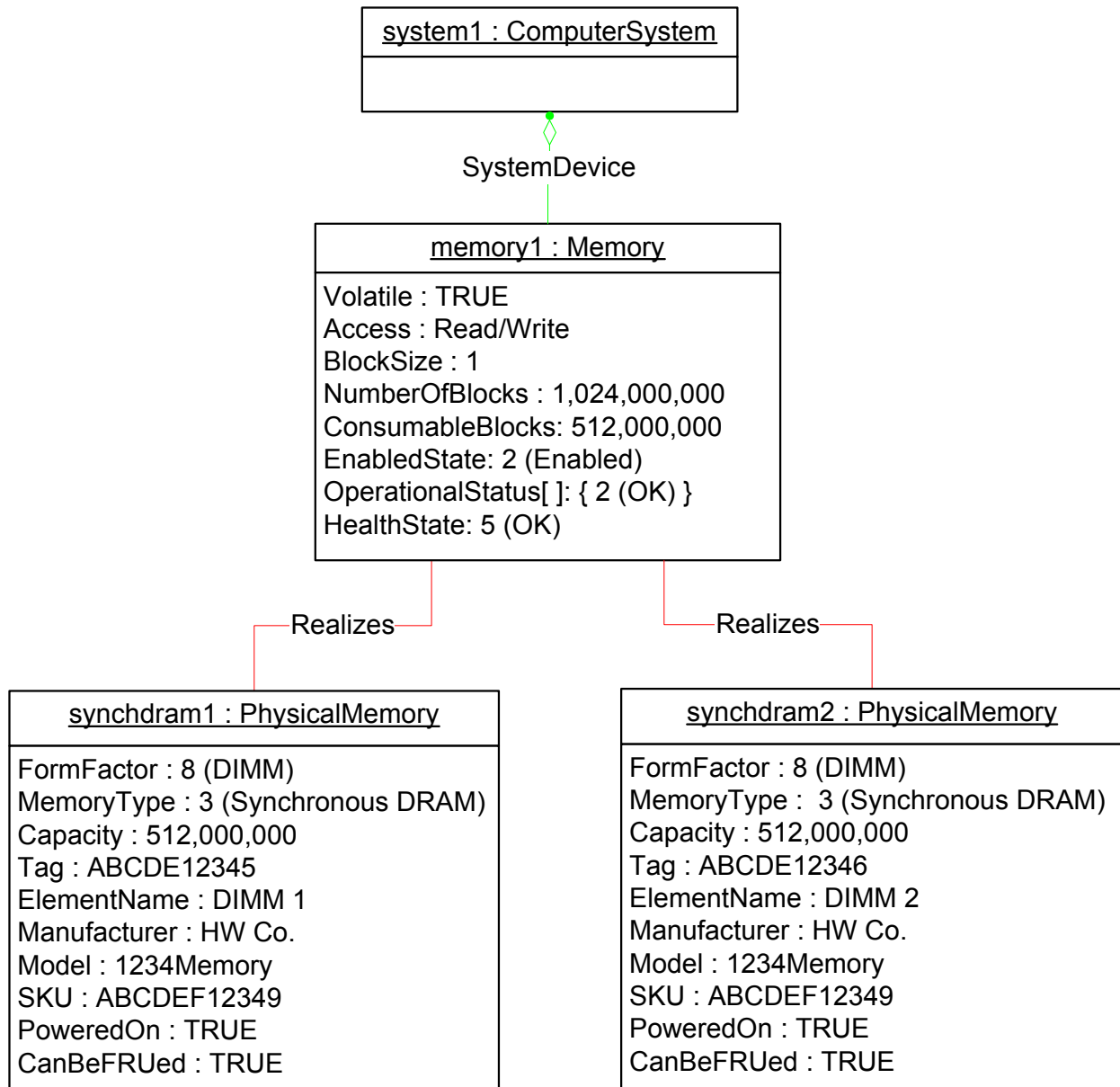
356

357 **Figure 2 – System Memory Profile: Object Diagram 1**

358 Figure 3 also represents a possible instantiation of the *System Memory Profile*. In this case, the computer
 359 system represented by the system1 instance of CIM_ComputerSystem owns and uses memory
 360 represented by the memory1 instance of CIM_Memory. memory1, as in Figure 2, contains logical
 361 information about the system memory, but in this case, memory1 is a logical representation of two
 362 physical memories, synchdram1 and synchdram2. Thus, the size of system memory, represented by the

363 properties BlockSize and NumberOfBlocks of memory1, is the sum of the physical sizes, represented by
 364 the Capacity property of synchdram1 and synchdram2. But because system1 has system memory
 365 redundancy, the available system memory represented by the properties of BlockSize and
 366 ConsumableBlocks is half of the total physical system memory. profile1 shows the version of the current
 367 *System Memory Profile* implementation.

368 For simplicity, the prefix CIM_ has been removed from the names of the classes.



369

370

Figure 3 – System Memory Profile: Object Diagram 2

371 **9.2 Find the System’s Memory Information**

372 A client can find the system’s memory information as follows:

- 373 1) Select the instance of CIM_Memory that is associated with the given instance of
374 CIM_ComputerSystem through the CIM_SystemDevice association.
- 375 2) Select the values of the properties of the CIM_Memory instance.

376 9.3 Find the System's Physical Memory Information

377 A client can find the system's physical memory information as follows:

- 378 1) Select all the instances of CIM_Realizes that reference the instance of CIM_Memory.
- 379 2) Select the CIM_PhysicalMemory instances that are referenced by the instances of
380 CIM_Realizes.
- 381 3) Select the values of the properties of the selected CIM_PhysicalMemory instances.

382 9.4 Find the Total Physical System Memory Size

383 A client can determine the total system memory size as follows:

- 384 1) Select the instance of CIM_Memory that is associated with the given instance of
385 CIM_ComputerSystem through the CIM_SystemDevice association.
- 386 2) For the instance of CIM_Memory, select the BlockSize and NumberOfBlocks properties and
387 multiply their values together to show the total system memory in bytes.

388 9.5 Find the Total Available System Memory Size

389 A client can determine the total available system memory size as follows:

- 390 1) Select the instance of CIM_Memory that is associated with the given instance of
391 CIM_ComputerSystem through the CIM_SystemDevice association.
- 392 2) For the instances of CIM_Memory, select the BlockSize and ConsumableBlocks properties and
393 multiply their values together to show the total available system memory in bytes.

394 9.6 Find the Physical System Memory Size per Memory Device

395 A client can determine the total physical system memory size as follows:

- 396 1) Select all of the instances of CIM_Realizes that reference the instance of the CIM_Memory.
- 397 2) Select the CIM_PhysicalMemory instances that are referenced by the CIM_Realizes instances.
- 398 3) The Capacity property of the selected instances of CIM_PhysicalMemory to show the size of
399 the physical system memory in bytes per memory device.

400 9.7 Determine Whether ElementName for the Instance of CIM_Memory Is 401 Modifiable

402 A client can determine whether it can modify the ElementName property of the CIM_Memory instance as
403 follows:

- 404 1) Select the instance of CIM_EnabledLogicalElementCapabilities that is associated with the
405 instance of CIM_Memory through the CIM_ElementCapabilities association.
- 406 2) Determine if the ElementNameEditSupported property has value of TRUE.

407 If the value is TRUE, the client can invoke the ModifyInstance operation to modify the value of the
408 CIM_Memory.ElementName property. If the value is FALSE or the instance of
409 CIM_EnabledLogicalElementCapabilities does not exist, the client cannot modify the value of the
410 CIM_Memory.ElementName property.

411 **10 CIM Elements**

412 Table 5 shows the instances of CIM Elements for this profile. Instances of the CIM Elements shall be
 413 implemented as described in Table 5. Sections 7 (“Implementation”) and 8 (“Methods”) may impose
 414 additional requirements on these elements.

415 **Table 5 – CIM Elements: System Memory Profile**

Element Name	Requirement	Description
Classes		
CIM_ElementCapabilities	Conditional	See 10.1.
CIM_EnabledLogicalElementCapabilities	Optional	See 7.3 and 10.2.
CIM_Memory	Mandatory	See 7.1 and 10.3.
CIM_RegisteredProfile	Mandatory	See 10.4.
CIM_SystemDevice	Mandatory	See 10.5.
Indications		
None defined in this profile		

416 **10.1 CIM_ElementCapabilities**

417 CIM_ElementCapabilities associates the CIM_Memory instance with the
 418 CIM_EnabledLogicalElementCapabilities instance that describes the capabilities of CIM_Memory.
 419 CIM_ElementCapabilities is mandatory when the CIM_EnabledLogicalElementCapabilities instance is
 420 instantiated. Table 6 contains the requirements for elements of this class.

421 **Table 6 – Class: CIM_ElementCapabilities**

Elements	Requirement	Notes
ManagedElement	Mandatory	Key: This property shall reference the instance of CIM_Memory. Cardinality 1..*, indicating one or many references
Capabilities	Mandatory	Key: This property shall reference the instance of CIM_EnabledLogicalElementCapabilities. Cardinality 0..1, indicating zero or one reference

422 **10.2 CIM_EnabledLogicalElementCapabilities**

423 CIM_EnabledLogicalElementCapabilities represents the capabilities of the system memory. Table 7
 424 contains the requirements for elements of this class.

425 **Table 7 – Class: CIM_EnabledLogicalElementCapabilities**

Elements	Requirement	Notes
InstanceID	Mandatory	Key
ElementNameEditSupported	Mandatory	See 7.3.1.
MaxElementNameLen	Conditional	See 7.3.2.

426 **10.3 CIM_Memory**

427 CIM_Memory represents the logical properties of memory in a managed system. For more
 428 implementation details, see 7.1. Table 8 contains the requirements for elements of this class.

429 **Table 8 – Class: CIM_Memory**

Elements	Requirement	Notes
SystemCreationClassName	Mandatory	Key
SystemName	Mandatory	Key
CreationClassName	Mandatory	Key
DeviceID	Mandatory	Key
Volatile	Mandatory	None
Access	Mandatory	None
BlockSize	Mandatory	None
NumberOfBlocks	Mandatory	See 7.2.
ConsumableBlocks	Mandatory	See 7.2.
EnabledState	Mandatory	This property shall match 2 (Enabled).
RequestedState	Mandatory	This property shall match 12 (Not Applicable).
OperationalStatus	Mandatory	None
HealthState	Mandatory	None
ElementName	Mandatory	This property shall be formatted as a free-form string of variable length (pattern “. *”).

430 **10.4 CIM_RegisteredProfile**

431 CIM_RegisteredProfile is defined by the [Profile Registration Profile](#). The requirements denoted in Table 9
 432 are in addition to those mandated by the [Profile Registration Profile](#).

433 **Table 9 – Class: CIM_RegisteredProfile**

Elements	Requirement	Notes
RegisteredName	Mandatory	This property shall have a value of “System Memory”.
RegisteredVersion	Mandatory	This property shall have a value of “1.0.1”.
RegisteredOrganization	Mandatory	This property shall have a value of 2 (DMTF).

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

437

438 **10.5 CIM_SystemDevice**

439 CIM_SystemDevice associates the CIM_Memory instance with the CIM_ComputerSystem instance of
 440 which CIM_Memory is a member. Table 10 contains the requirements for elements of this class.

441 **Table 10 – Class: CIM_SystemDevice**

Elements	Requirement	Notes
GroupComponent	Mandatory	Key: This property shall reference the CIM_ComputerSystem instance of which the CIM_Memory instance is a member. Cardinality 1, indicating one reference
PartComponent	Mandatory	Key: This property shall reference the CIM_Memory instance. Cardinality 1, indicating one reference

442
443
444
445

ANNEX A (informative)

Change Log

Version	Date	Description
1.0.0f	2006-08-08	Preliminary Standard
1.0.0	2007-10-12	Final Standard
1.0.1	2010-06-10	DMTF Standard Errata release to correct algorithms in 7.2.1 and 7.2.2.

446
447