



1

2

3

4

Document Number: DSP1026

Date: 2006-08-08

Version: 1.0.0f

5 **System Memory Profile**

6 **Document Type: Specification**

7 **Document Status: Preliminary Standard**

8 **Document Language: E**

System Memory Profile

9 Copyright Notice

10 Copyright © 2006 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26 withdrawn or modified after publication, and shall be indemnified and held harmless by any party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	5
31	Introduction	6
32	1 Scope	7
33	2 Normative References.....	7
34	2.1 Approved References	7
35	2.2 References under Development.....	7
36	2.3 Other 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	Representation of the.....	10
44	7.2 System Memory Size	10
45	7.3 CIM_EnabledLogicalElementCapabilities.....	11
46	7.4 Physical Asset Profile (Optional)	11
47	8 Methods.....	12
48	8.1 Profile Conventions for Operations.....	12
49	8.2 CIM_ElementCapabilities	12
50	8.3 CIM_EnabledLogicalElementCapabilities.....	13
51	8.4 CIM_Memory	13
52	8.5 CIM_SystemDevice	13
53	9 Use Cases.....	14
54	9.1 Object Diagrams	14
55	9.2 Find the System's Memory Information	16
56	9.3 Find the System's Physical Memory Information.....	16
57	9.4 Find the Total Physical System Memory Size	17
58	9.5 Find the Total Available System Memory Size	17
59	9.6 Find the Physical System Memory Size per Memory Device.....	17
60	9.7 Determine Whether ElementName for the Instance of CIM_Memory Is Modifiable.....	17
61	10 CIM Elements.....	18
62	10.1 CIM_ElementCapabilities	18
63	10.2 CIM_EnabledLogicalElementCapabilities.....	18
64	10.3 CIM_Memory	19
65	10.4 CIM_RegisteredProfile.....	19
66	10.5 CIM_SystemDevice	20
67	ANNEX A (informative) Change Log.....	21
68	ANNEX B (informative) Acknowledgments.....	22
69		
70	Figures	
71	Figure 1 – System Memory Profile: Profile Class Diagram.....	10
72	Figure 2 – System Memory Profile: Object Diagram 1	15
73	Figure 3 – System Memory Profile: Object Diagram 2	16
74		

System Memory Profile

75 Tables

76	Table 1 – Referenced Profiles	9
77	Table 2 – Operations: CIM_ElementCapabilities	12
78	Table 3 – Operations: CIM_Memory	13
79	Table 4 – Operations: CIM_SystemDevice	13
80	Table 5 – CIM Elements: System Memory Profile	18
81	Table 6 – Class: CIM_ElementCapabilities	18
82	Table 7 – Class: CIM_EnabledLogicalElementCapabilities	18
83	Table 8 – Class: CIM_Memory	19
84	Table 9 – Class: CIM_RegisteredProfile	19
85	Table 10 – Class: CIM_SystemDevice	20

86

87

Foreword

88 The *System Memory Profile* (DSP1026) was prepared by the Server Management Working Group of the
89 DMTF.

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

Introduction

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

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

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

102

System Memory Profile

103 1 Scope

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

108 2 Normative References

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

112 2.1 Approved References

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

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

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

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

117 2.2 References under Development

118 DMTF DSP1011, *Physical Asset Profile*

119 DMTF DSP1033, *Profile Registration Profile*

120 2.3 Other References

121 ISO/IEC Directives, Part 2, [Rules for the structure and drafting of International Standards](#),

122 [Common Information Model \(CIM\) Specification v2.2](#), June 14, 1999

123 [Unified Modeling Language \(UML\) from the Open Management Group \(OMG\)](#)

124 DMTF [DSP0215](#), *Server Management Managed Element Addressing Specification (SM ME Addressing)*

125 DMTF DSP0134, *System Management BIOS Reference Specification, v2.5 Preliminary*

126 3 Terms and Definitions

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

System Memory Profile

- 129 **3.1**
130 **can**
131 used for statements of possibility and capability, whether material, physical, or causal
- 132 **3.2**
133 **cannot**
134 used for statements of possibility and capability, whether material, physical, or causal
- 135 **3.3**
136 **conditional**
137 indicates requirements to be followed strictly to conform to the document when the specified conditions
138 are met
- 139 **3.4**
140 **mandatory**
141 indicates requirements to be followed strictly to conform to the document and from which no deviation is
142 permitted
- 143 **3.5**
144 **may**
145 indicates a course of action permissible within the limits of the document
- 146 **3.6**
147 **need not**
148 indicates a course of action permissible within the limits of the document
- 149 **3.7**
150 **optional**
151 indicates a course of action permissible within the limits of the document
- 152 **3.8**
153 **referencing profile**
154 indicates a profile that owns the definition of this class and can include a reference to this profile in its
155 "Referenced Profiles" table
- 156 **3.9**
157 **shall**
158 indicates requirements to be followed strictly to conform to the document and from which no deviation is
159 permitted
- 160 **3.10**
161 **shall not**
162 indicates requirements to be followed strictly to conform to the document and from which no deviation is
163 permitted
- 164 **3.11**
165 **should**
166 indicates that among several possibilities, one is recommended as particularly suitable, without
167 mentioning or excluding others, or that a certain course of action is preferred but not necessarily required
- 168 **3.12**
169 **should not**
170 indicates that a certain possibility or course of action is deprecated but not prohibited

171 **3.13**
 172 **unspecified**
 173 indicates that this profile does not define any constraints for the referenced CIM element or operation

174 **4 Symbols and Abbreviated Terms**

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

176 **4.1**
 177 **CIM**
 178 Common Information Model

179 **4.2**
 180 **FRU**
 181 Field Replaceable Unit

182

183 **5 Synopsis**

184 **Profile Name:** *System Memory Profile*

185 **Version:** 1.0.0f

186 **Organization:** DMTF

187 **CIM Schema version:** 2.10

188 **Central Class:** CIM_Memory

189 **Scoping Class:** CIM_ComputerSystem

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

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

194

Table 1 – Referenced Profiles

Profile Name	Organization	Version	Relationship	Behavior
<i>Physical Asset Profile</i>	DMTF	1.0	Optional	See section 7.4.
<i>Profile Registration Profile</i>	DMTF	1.0	Mandatory	None

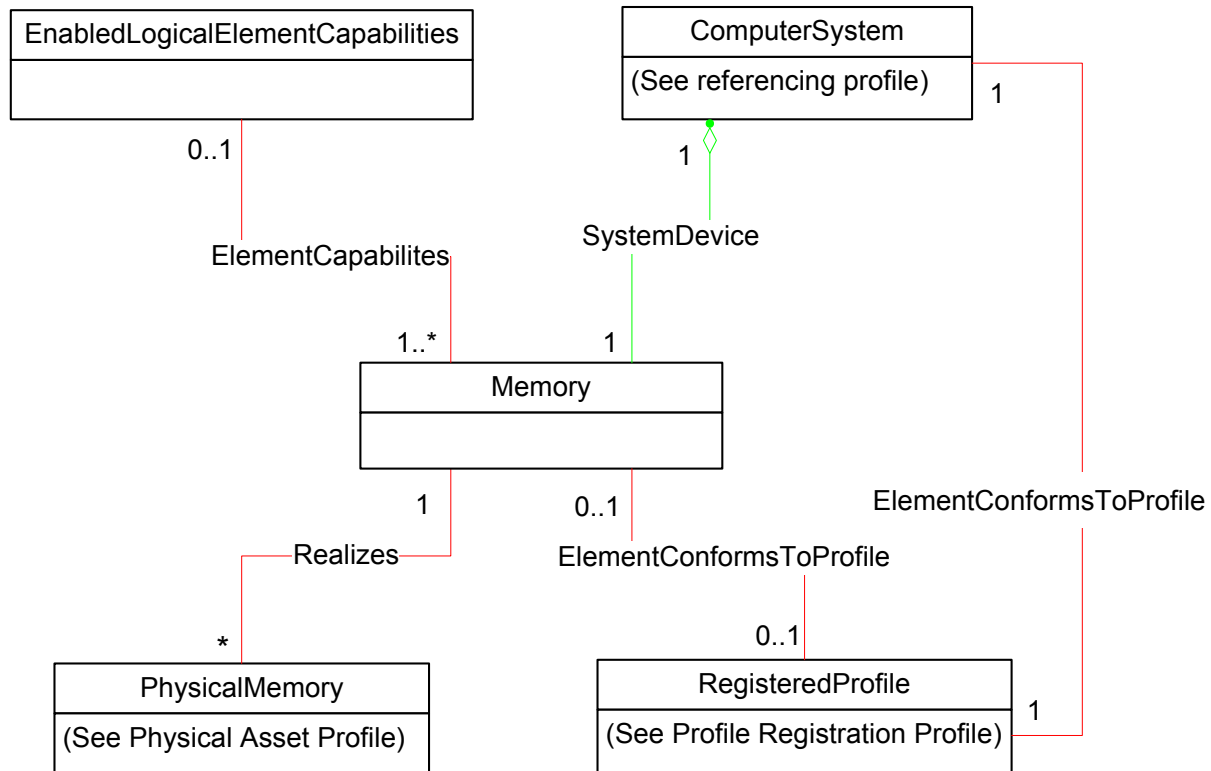
195 **6 Description**

196 The *System Memory Profile* extends the management capability of the referencing profiles by adding the
 197 capability to describe memory devices associated with a system. The CIM_Memory class describes the
 198 system's total memory that includes both available and physical memory, and corresponds to collective
 199 representation of the SMBIOS Type 16, Type 17, Type 19, and Type 20 memory structures (see section
 200 **Error! Reference source not found.**). The CIM_PhysicalMemory class describes the physical aspects of the
 201 system's memory and every instance corresponds to the individual SMBIOS Type 17 memory structure.

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

System Memory Profile

204 The CIM_Memory class describes the total system memory. The physical aspects of the memory are
205 described with the CIM_PhysicalMemory class, which is associated with the CIM_Memory class through
206 the CIM_Realizes association. The ownership of the memory is represented through the
207 CIM_SystemDevice association to the managed system, which is represented by the
208 CIM_ComputerSystem class. The DMTF version of the *System Memory Profile* that is implemented is
209 represented through the CIM_RegisteredProfile class.



210

211

Figure 1 – System Memory Profile: Profile Class Diagram

212 7 Implementation

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

216 7.1 CIM_Memory

217 A single instance of CIM_Memory shall exist in accordance with the profile instantiation and shall be
218 associated with the CIM_ComputerSystem instance that represents the managed system through an
219 instance of CIM_SystemDevice. This CIM_Memory instance is the logical representation of multiple
220 physical memories in a managed system and represents the total memory installed and available to the
221 system..

222 7.2 Representation of the System Memory Size

223 This section describes the method of total system memory size calculation. Additionally this section
224 describes the corresponding SMBIOS structures defined in [DSP0134](#) that may be used for the
225 calculation. Note that the underlying represented system does not need to support [DSP0134](#).

226 7.2.1 Total Available System Memory

227 The ConsumableBlocks property represents the number of total available system memory blocks. When
228 the number of available system memory blocks is unknown, the value of ConsumableBlocks shall be set
229 to Null.

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

232 When the [DSP0134](#) is implemented, the product of multiplying the value of the
233 CIM_Memory.ConsumableBlocks property by the value of the CIM_Memory.BlockSize property shall
234 equal the sum of the values of the products of 1 KByte with the subtraction of value of StartingAddress
235 property from the value of EndingAddress property ($\sum [1\text{KB} \times (\text{EndingAddress} - \text{StartingAddress})]$) of
236 either Memory Array Mapped Device (Type 19) structures or Memory Device Mapped Address (Type 20)
237 structures of SMBIOS specification [DSP0134](#).

238 7.2.2 Total Physical System Memory

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

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

242 When the [DSP0134](#) is implemented, the product of multiplying the value of the
243 CIM_Memory.NumberOfBlocks property by the value of the CIM_Memory.BlockSize property shall equal
244 the sum of the values of the Size property of Memory Device (Type 17) structure of SMBIOS specification
245 [DSP0134](#).

246 When the optional behavior of modeling the physical aspects of the system memory specified in section
247 7.4 is implemented, the product of multiplying the value of the CIM_Memory.NumberOfBlocks property by
248 the value of the CIM_Memory.BlockSize property shall equal the sum of the values of the
249 CIM_PhysicalMemory.Capacity property for each instance of the CIM_PhysicalMemory to which the
250 CIM_Memory instance is associated through the CIM_Realizes association.

251 7.3 CIM_EnabledLogicalElementCapabilities

252 When the CIM_EnabledLogicalElementCapabilities class is instantiated, the instance of
253 CIM_EnabledLogicalElementCapabilities shall be associated with the CIM_Memory instance through an
254 instance of CIM_ElementCapabilities and used for advertising the capabilities of the CIM_Memory
255 instance.

256 At most one instance of CIM_EnabledLogicalElementCapabilities shall be associated with a given
257 instance of CIM_Memory.

258 7.3.1 CIM_EnabledLogicalElementCapabilities.ElementNameEditSupported

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

261 7.3.2 CIM_EnabledLogicalElementCapabilities.MaxElementNameLen

262 The MaxElementNameLen property shall be implemented when the ElementNameEditSupported
263 property has a value of TRUE.

264 7.4 Physical Asset Profile (Optional)

265 The *Physical Asset Profile* may be implemented to model the physical aspects of the system memory,
266 including the asset information.

System Memory Profile

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

270 8 Methods

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

273 8.1 Profile Conventions for Operations

274 Support for operations for each profile class (including associations) is specified in the following
275 subclauses. Each subclause includes either the statement "All operations in the default list in section 8.1
276 are supported as described by [DSP0200 version 1.2](#)" or a table listing all of the operations that are not
277 supported by this profile or where the profile requires behavior other than that described by
278 [DSP0200 version 1.2](#).

279 The default list of operations is as follows:

- 280 • GetInstance
- 281 • EnumerateInstances
- 282 • EnumerateInstanceNames
- 283 • Associators
- 284 • AssociatorNames
- 285 • References
- 286 • ReferenceNames

287 A compliant implementation shall support all of the operations in the default list for each class, unless the
288 "Requirement" column states something other than *Mandatory*.

289 8.2 CIM_ElementCapabilities

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

292 **Table 2 – Operations: CIM_ElementCapabilities**

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

293 8.3 CIM_EnabledLogicalElementCapabilities

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

295 8.4 CIM_Memory

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

298 **Table 3 – Operations: CIM_Memory**

Operation	Requirement	Messages
ModifyInstance	Optional. See section 8.4.1.	None

299 8.4.1 CIM_Memory—ModifyInstance

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

302 The ModifyInstance operation shall be supported and CIM_Memory.ElementName shall be modifiable
303 when an instance of CIM_EnabledLogicalElementCapabilities is associated to the instance of
304 CIM_Memory, and the ElementNameEditSupported property of the
305 CIM_EnabledLogicalElementCapabilities instance has a value of TRUE. See section 8.4.1.1.

306 8.4.1.1 CIM_Memory.ElementName

307 When an instance of CIM_EnabledLogicalElementCapabilities is associated to the instance of
308 CIM_Memory, and the ElementNameEditSupported property of the
309 CIM_EnabledLogicalElementCapabilities instance has a value of TRUE, the implementation shall allow
310 the ModifyInstance operation to change the value of the ElementName property of the CIM_Memory
311 instance. The ModifyInstance operation shall enforce the length restriction specified in the
312 MaxElementNameLen property of the CIM_EnabledLogicalElementCapabilities instance.

313 When the ElementNameEditSupported property of the CIM_EnabledLogicalElementCapabilities instance
314 has a value of FALSE or when there is no instance of CIM_EnabledLogicalElementCapabilities
315 associated with the instance of CIM_Memory, the implementation shall not allow the ModifyInstance
316 operation to change the value of the ElementName property of the CIM_Memory instance.

317 8.5 CIM_SystemDevice

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

320 **Table 4 – Operations: CIM_SystemDevice**

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

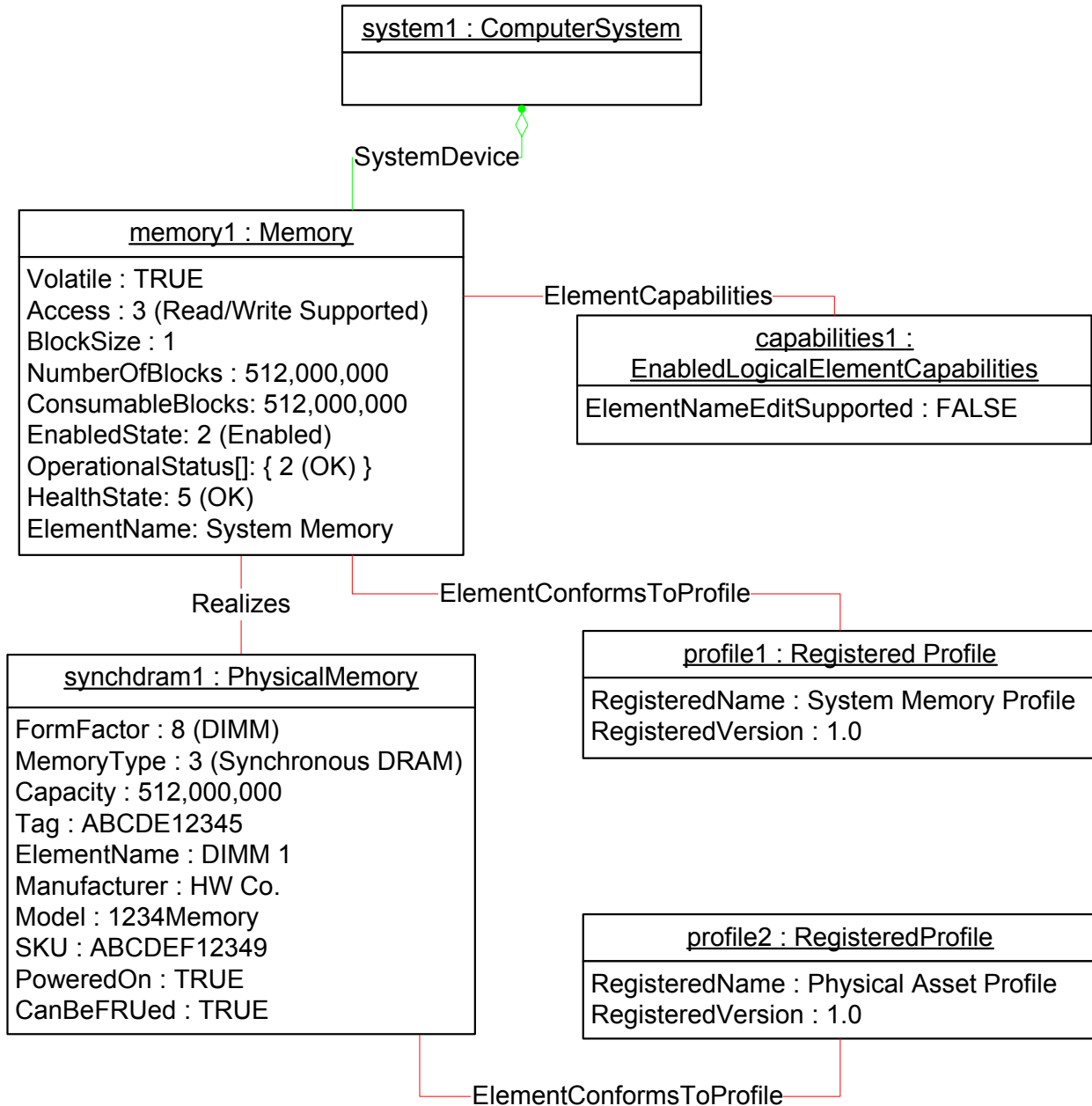
321 **9 Use Cases**

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

323 **9.1 Object Diagrams**

324 Figure 2 represents a possible instantiation of the *System Memory Profile*. In this case, the computer
325 system represented by the system1 instance of CIM_ComputerSystem owns and uses memory
326 represented by the memory1 instance of CIM_Memory. memory1 contains information such as the
327 volatility, accessibility, total physical and available size, and operational and health status of the memory.
328 The property ConsumableBlocks represents the memory blocks available for consumption by the system
329 and is equal to the total physical system memory blocks represented by the NumberOfBlocks property.
330 The physical aspects such as memory type, physical size, form factor, and other FRU data are described
331 by the synchdram1 instance of CIM_PhysicalMemory. profile1 shows the version of the current *System*
332 *Memory Profile* implementation.

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



334

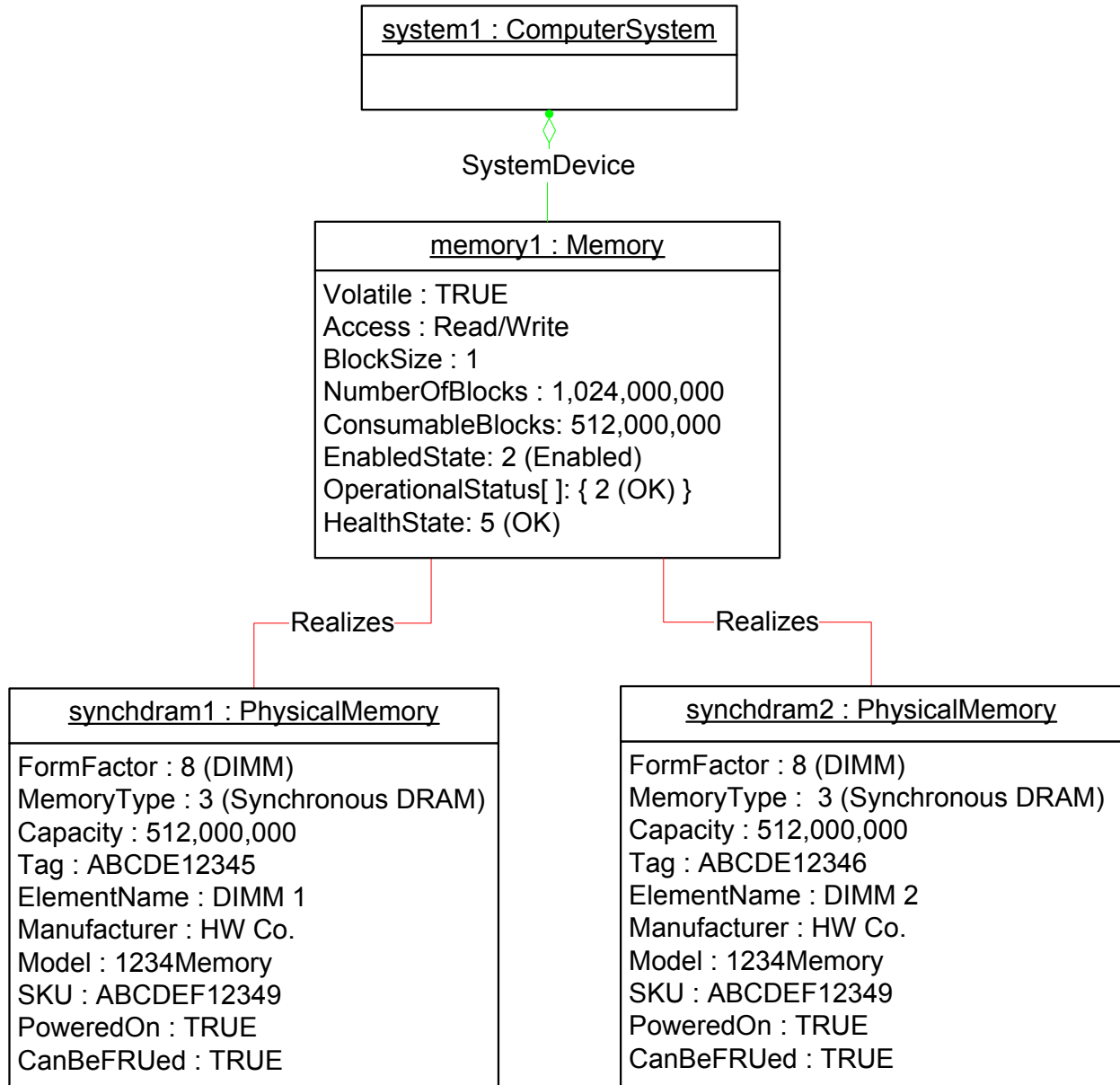
335

Figure 2 – System Memory Profile: Object Diagram 1

336 Figure 3 also represents a possible instantiation of the *System Memory Profile*. In this case, the computer
 337 system represented by the system1 instance of CIM_ComputerSystem owns and uses memory
 338 represented by the memory1 instance of CIM_Memory. memory1, as in Figure 2, contains logical
 339 information about the system memory, but in this case, memory1 is a logical representation of two
 340 physical memories, synchdram1 and synchdram2. Thus, the size of system memory, represented by the
 341 properties BlockSize and NumberOfBlocks of memory1, is the sum of the physical sizes, represented by
 342 the Capacity property of synchdram1 and synchdram2. But because system1 has system memory
 343 redundancy, the available system memory represented by the properties of BlockSize and
 344 ConsumableBlocks is half of the total physical system memory. profile1 shows the version of the current
 345 *System Memory Profile* implementation.

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

System Memory Profile



347

348

Figure 3 – System Memory Profile: Object Diagram 2

349 9.2 Find the System's Memory Information

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

- 351 1) Select the instance of CIM_Memory that is associated with the given instance of
- 352 CIM_ComputerSystem through the CIM_SystemDevice association.
- 353 2) Select the values of the properties of the CIM_Memory instance.

354 9.3 Find the System's Physical Memory Information

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

- 356 1) Select all the instances of CIM_Realizes that reference the instance of CIM_Memory.

- 357 2) Select the CIM_PhysicalMemory instances that are referenced by the instances of
358 CIM_Realizes.
359 3) Select the values of the properties of the selected CIM_PhysicalMemory instances.

360 **9.4 Find the Total Physical System Memory Size**

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

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

366 **9.5 Find the Total Available System Memory Size**

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

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

372 **9.6 Find the Physical System Memory Size per Memory Device**

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

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

378 **9.7 Determine Whether ElementName for the Instance of CIM_Memory Is** 379 **Modifiable**

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

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

385 If the value is TRUE, the client can invoke the ModifyInstance operation to modify the value of the
386 CIM_Memory.ElementName property. If the value is FALSE or the instance of
387 CIM_EnabledLogicalElementCapabilities does not exist, the client cannot modify the value of the
388 CIM_Memory.ElementName property.

System Memory Profile

389 10 CIM Elements

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

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

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

394 10.1 CIM_ElementCapabilities

395 CIM_ElementCapabilities associates the CIM_Memory instance with the
396 CIM_EnabledLogicalElementCapabilities instance that describes the capabilities of CIM_Memory.
397 CIM_ElementCapabilities is mandatory when the CIM_EnabledLogicalElementCapabilities instance is
398 instantiated. Table 6 contains the requirements for elements of this class.

399 **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

400 10.2 CIM_EnabledLogicalElementCapabilities

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

403 **Table 7 – Class: CIM_EnabledLogicalElementCapabilities**

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

404 **10.3 CIM_Memory**

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

407 **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 section Error! Reference source not found.
ConsumableBlocks	Mandatory	See section Error! Reference source not found.
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 “.*”).

408 **10.4 CIM_RegisteredProfile**

409 CIM_RegisteredProfile is defined by the *Profile Registration Profile*. The requirements denoted in Table 9
 410 are in addition to those mandated by the *Profile Registration Profile*.

411 **Table 9 – Class: CIM_RegisteredProfile**

Elements	Requirement	Notes
RegisteredName	Mandatory	This property shall have a value of “System Memory Profile”.
RegisteredVersion	Mandatory	This property shall have a value of “1.0.0f”.
RegisteredOrganization	Mandatory	This property shall have a value of 2 (DMTF).

System Memory Profile

412 10.5 CIM_SystemDevice

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

415 **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

416
417
418
419**ANNEX A
(informative)****Change Log**

Version	Date	Description
0.0		Document formulation. Initial diagram and class property/method information.
0.1		Initial draft. Content mostly complete, not necessarily correct.
0.7.0	2005/01/11	Updated per December 2005 SMWF F2F discussion. Also updated with new diagram format and physical breakout.
0.7.1	2005/08/26	Simplified System Memory Profile's class schema and added all the necessary information per Profile Usage Guide 1.0.0c.
0.8.1	2006/02/02	Converted to ISO template.

420

421
422
423
424

ANNEX B (informative)

Acknowledgments

425 The authors wish to acknowledge the following people.

426 **Editor:**

- 427 • Khachatur Papanyan – Dell Inc.

428 **Contributors:**

- 429 • Jon Hass – Dell
- 430 • Khachatur Papanyan – Dell
- 431 • Jeff Hilland – HP
- 432 • Christina Shaw – HP
- 433 • Aaron Merkin – IBM
- 434 • Jeff Lynch – IBM
- 435 • Perry Vincent – Intel
- 436 • John Leung – Intel
- 437