



1

2

3

4

Document Identifier: DSP1052

Date : 2014-05-22

Version: 1.0.3

5 **Computer System Profile**

6 **Document Type: Specification**

7 **Document Status: DMTF Standard**

8 **Document Language: en-US**

9 Copyright Notice

10 Copyright © 2006–2014 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25 withdrawn or modified after publication, and shall be indemnified and held harmless by any party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	8
38	4 Symbols and Abbreviated Terms	8
39	5 Synopsis	9
40	6 Description	9
41	7 Implementation.....	10
42	7.1 Computer System	10
43	7.2 Management of Computer System Components	13
44	7.3 Software Asset Management.....	14
45	7.4 Network Interface Management.....	14
46	7.5 Record Logs.....	15
47	7.6 Management of Protocol Services.....	15
48	7.7 System Lifecycle Management.....	16
49	7.8 Smash Collections Profile.....	18
50	8 Methods.....	18
51	8.1 CIM_ComputerSystem.RequestStateChange()	18
52	8.2 CIM_TimeService.ManageTime().....	19
53	8.3 Profile Conventions for Operations.....	20
54	8.4 CIM_ComputerSystem.....	20
55	8.5 CIM_ElementCapabilities	21
56	8.6 CIM_EnabledLogicalElementCapabilities.....	21
57	8.7 CIM_HostedService	21
58	8.8 CIM_ServiceAffectsElement	22
59	8.9 CIM_TimeService	22
60	9 Use Cases.....	22
61	9.1 Object Diagrams	22
62	9.2 Find a Dedicated System.....	24
63	9.3 Correlate Instrumented Systems	24
64	9.4 Enable a System.....	25
65	9.5 Disable a System.....	25
66	9.6 Reset a System.....	25
67	9.7 Manage the System Boot Configuration.....	25
68	9.8 Determine the Number of Processors in the System	25
69	9.9 Determine If Time Management Is Supported.....	26
70	9.10 Get Time for System	26
71	9.11 Set Time for System	26
72	9.12 Determining If ElementName Can Be Modified.....	26
73	9.13 Determining If State Management Is Supported	26
74	10 CIM Elements.....	27
75	10.1 CIM_ComputerSystem.....	27
76	10.2 CIM_ElementCapabilities	28
77	10.3 CIM_EnabledLogicalElementCapabilities.....	28
78	10.4 CIM_HostedService	28
79	10.5 CIM_ServiceAffectsElement	29
80	10.6 CIM_TimeService	29
81	ANNEX A (Informative) Change Log	30

83 **Figures**

84	Figure 1 – Computer System Profile: Class Diagram	10
85	Figure 2 – Logical Topology.....	23
86	Figure 3 – Network Interfaces	24

87

88 **Tables**

89	Table 1 – Referenced Profiles	9
90	Table 2 – Predefined Identifiers for a Computer System	11
91	Table 3 – CIM_ComputerSystem.RequestStateChange() Method: Return Code Values	18
92	Table 4 – CIM_ComputerSystem.RequestStateChange() Method: Parameters	19
93	Table 5 – CIM_TimeService.ManageTime() Method: Return Code Values	19
94	Table 6 – CIM_TimeService.ManageTime() Method: Parameters	20
95	Table 7 – Operations: CIM_ComputerSystem	20
96	Table 8 – Operations: CIM_ElementCapabilities	21
97	Table 9 – Operations: CIM_HostedService	22
98	Table 10 – Operations: CIM_ServiceAffectsElement	22
99	Table 11 – CIM Elements: Computer System Profile	27
100	Table 12 – Class: CIM_ComputerSystem.....	27
101	Table 13 – Class: CIM_ElementCapabilities.....	28
102	Table 14 – Class: CIM_EnabledLogicalElementCapabilities.....	28
103	Table 15 – Class: CIM_HostedService	28
104	Table 16 – Class: CIM_ServiceAffectsElement	29
105	Table 17 – Class: CIM_TimeService	29

106

107

Foreword

108 The *Computer System Profile* (DSP1052) was prepared by the Server Management Working Group and
109 Physical Platform Profiles Working Group of the DMTF.

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

112 **Acknowledgments**

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

114 Editors:

- 115 • Steve Lee - Microsoft
- 116 • Hemal Shah – Broadcom
- 117 • Aaron Merkin – IBM
- 118 • Jeff Hilland – HP

119 Contributors:

- 120 • Jon Hass – Dell
- 121 • Khachatur Papanyan – Dell
- 122 • Jeff Hilland – HP
- 123 • Christina Shaw – HP
- 124 • Aaron Merkin – IBM
- 125 • Perry Vincent – Intel
- 126 • John Leung – Intel
- 127 • Hemal Shah – Broadcom
- 128 • David Hines – Intel
- 129 • Jim Davis – WBEM Solutions

130

131

Introduction

132 The information in this specification should be sufficient for a provider or consumer of this data to
133 unambiguously identify the classes, properties, methods, and values that shall be instantiated and
134 manipulated to represent and manage a basic computer system and subsystems that are modeled using
135 the DMTF Common Information Model (CIM) core and extended model definitions.

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

138

139 Document Conventions

140 Experimental Material

141 Experimental material has yet to receive sufficient review to satisfy the adoption requirements set forth by
142 the DMTF. Experimental material is included in this document as an aid to implementers who are
143 interested in likely future developments. Experimental material may change as implementation
144 experience is gained. It is likely that experimental material will be included in an upcoming revision of the
145 document. Until that time, experimental material is purely informational.

146 The following typographical convention indicates experimental material:

147 **EXPERIMENTAL**

148 Experimental material appears here.

149 **EXPERIMENTAL**

150 In places where this typographical convention cannot be used (for example, tables or figures), the
151 "EXPERIMENTAL" label is used alone.

152

Computer System Profile

1 Scope

The *Computer System Profile* is the autonomous profile that defines the minimum top-level object model needed to define a basic computing platform. This profile is intended to be the base profile for specialization for the modeling of specific types of computer systems such as virtual machines, servers, desktops, and mobile computers. The *Computer System Profile* identifies component profiles for integration of additional management functionality including system configuration, boot control, and other provisioning capabilities.

2 Normative References

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

DMTF DSP0004, *CIM Infrastructure Specification 2.5*,
http://www.dmtf.org/standards/published_documents/DSP0004_2.5.pdf

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

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

DMTF DSP1005, *CLP Service Profile 1.0*,
http://www.dmtf.org/standards/published_documents/DSP1005_1.0.pdf

DMTF DSP1006, *SMASH Collections Profile 1.0*,
http://www.dmtf.org/standards/published_documents/DSP1006_1.0.pdf

DMTF DSP1009, *Sensors Profile 1.0*,
http://www.dmtf.org/standards/published_documents/DSP1009_1.0.pdf

DMTF DSP1010, *Record Log Profile 1.0*,
http://www.dmtf.org/standards/published_documents/DSP1010_1.0.pdf

DMTF DSP1012, *Boot Control Profile 1.0*,
http://www.dmtf.org/standards/published_documents/DSP1012_1.0.pdf

DMTF DSP1014, *Ethernet Port Profile 1.0*,
http://www.dmtf.org/standards/published_documents/DSP1014_1.0.pdf

DMTF DSP1016, *Telnet Service Profile 1.0*,
http://www.dmtf.org/standards/published_documents/DSP1016_1.0.pdf

DMTF DSP1017, *SSH Service Profile 1.0*,
http://www.dmtf.org/standards/published_documents/DSP1017_1.0.pdf

DMTF DSP1022, *CPU Profile 1.0*,
http://www.dmtf.org/standards/published_documents/DSP1022_1.0.pdf

DMTF DSP1023, *Software Inventory Profile 1.0*,
http://www.dmtf.org/standards/published_documents/DSP1023_1.0.pdf

- 191 DMTF DSP1024, *Text Console Redirection Profile 1.0*,
192 http://www.dmtf.org/standards/published_documents/DSP1024_1.0.pdf
- 193 DMTF DSP1025, *Software Update Profile 1.0*,
194 http://www.dmtf.org/standards/published_documents/DSP1025_1.0.pdf
- 195 DMTF DSP1026, *System Memory Profile 1.0*,
196 http://www.dmtf.org/standards/published_documents/DSP1026_1.0.pdf
- 197 DMTF DSP1033, *Profile Registration Profile 1.0*,
198 http://www.dmtf.org/standards/published_documents/DSP1033_1.0.pdf
- 199 DMTF DSP1036, *IP Interface Profile 1.0*,
200 http://www.dmtf.org/standards/published_documents/DSP1036_1.0.pdf
- 201 DMTF DSP1037, *DHCP Client Profile 1.0*,
202 http://www.dmtf.org/standards/published_documents/DSP1037_1.0.pdf
- 203 DMTF DSP1038, *DNS Client Profile 1.0*,
204 http://www.dmtf.org/standards/published_documents/DSP1038_1.0.pdf
- 205 ISO/IEC Directives, Part 2, *Rules for the structure and drafting of International Standards*
206 <http://isotc.iso.org/livelink/livelink.exe?func=ll&objId=4230456&objAction=browse&sort=subtype>

207 **3 Terms and Definitions**

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

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

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

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

221 For the purposes of this document, the terms defined in [DSP0004](#), [DSP0200](#), [DSP1001](#), and [DSP1033](#)
222 apply to this document.

223 **4 Symbols and Abbreviated Terms**

224 The following abbreviations are used in this document.

225 **4.1**

226 **IP**

227 Internet Protocol

228 **4.2**

229 **SSH**

230 Secure Shell

231 5 Synopsis

232 **Profile Name:** Computer System

233 **Version:** 1.0.1

234 **Organization:** DMTF

235 **CIM Schema Version:** 2.35

236 **Central Class:** CIM_ComputerSystem

237 **Scoping Class:** CIM_ComputerSystem

238 This abstract profile specification shall not be directly implemented; implementations shall be based on a
239 profile specification that specializes the requirements of this profile.

240 The *Computer System Profile* is an autonomous profile that provides the capability to manage a general-
241 purpose computer system. It is an appropriate target for management for clients that are interested in
242 performing management tasks that are common across diverse computing platforms such as virtual
243 machines, servers, and desktop platforms.

244 The Central Class of the *Computer System Profile* shall be CIM_ComputerSystem. The Central Instance
245 shall be an instance of CIM_ComputerSystem. The Scoping Class shall be CIM_ComputerSystem. The
246 Scoping Instance shall be the Central Instance. Table 1 lists profiles upon which this profile has a
247 dependency. The list in Table 1 is not the complete list of profiles that are allowed to be associated with
248 the *Computer System Profile*, as dictated by the requirements of those profiles. Other profiles shall not be
249 prohibited from being associated with or scoped to the ComputerSystem Central Instance of this profile.

250

Table 1 – Referenced Profiles

Profile Name	Organization	Version	Relationship	Behavior
Boot Control	DMTF	1.0	Optional	See 7.7.2.
CLP Service	DMTF	1.0	Optional	See 7.6.1.
CPU	DMTF	1.0	Optional	See 7.2.1.
DHCP Client	DMTF	1.0	Optional	See 7.4.3.
DNS Client	DMTF	1.0	Optional	See 7.4.4.
Ethernet Port	DMTF	1.0	Optional	See 7.4.1.
IP Interface	DMTF	1.0	Optional	See 7.4.2.
Record Log	DMTF	1.0	Optional	See 7.5.
Sensors	DMTF	1.0	Optional	See 7.2.3.
SMASH Collections	DMTF	1.0	Optional	See 7.8.
Software Inventory	DMTF	1.0	Optional	See 7.3.1.
Software Update	DMTF	1.0	Optional	See 7.3.2.
SSH Service	DMTF	1.0	Optional	See 7.6.2.
System Memory	DMTF	1.0	Optional	See 7.2.2.
Telnet Service	DMTF	1.0	Optional	See 7.6.3.

251 6 Description

252 The *Computer System Profile* is an autonomous profile that defines the minimum top-level object model
253 needed to model computer systems and related software. Other profiles add additional management
254 objects to this basic system model to provide system configuration, boot control, and other provisioning

255 capabilities. CIM_ComputerSystem represents the computer system. CIM_TimeService provides the
256 ability to manage the system time.

257 Figure 1 presents the class schema for the *Computer System Profile*. For simplicity, the prefix CIM_ has
258 been removed from the names of the classes.

259

260

Figure 1 – Computer System Profile: Class Diagram

261 **7 Implementation**

262 The *Computer System Profile* consists of definitions for the classes CIM_ComputerSystem and
263 CIM_TimeService, and their related EnabledLogicalElementCapabilities classes. Other related subsystem
264 classes such as CIM_LogicalDevice, CIM_Collection, and CIM_RecordLog are defined in their respective
265 profiles.

266 Requirements for propagating and formulating certain properties of the *Computer System Profile* classes
267 are discussed in this clause.

268 Methods are described in 8 (“Methods”), and properties are described in 10 (“CIM Elements”).

269 **7.1 Computer System**

270 The instrumentation shall create an instance of CIM_ComputerSystem to represent the system being
271 modeled.

272 **7.1.1 Identifying a Computer System**

273 Name/Value pairs contained in the CIM_ComputerSystem.OtherIdentifyingInfo and
 274 CIM_ComputerSystem.IdentifyingDescriptions properties should contain values that clients can use to
 275 correlate instances of CIM_ComputerSystem that represent the same underlying real-world system that
 276 the specialization of the *Computer System Profile* has been instrumented to represent. The following
 277 paragraphs detail the requirements when the OtherIdentifyingInfo and IdentifyingDescriptions properties
 278 are implemented.

279 When the OtherIdentifyingInfo property is implemented, the IdentifyingDescriptions property shall be
 280 implemented. The IdentifyingDescriptions property shall be formatted using the following algorithm:

281 < OrgID > : < LocalID >, where < OrgID > and < LocalID > are separated by a colon (:) and
 282 < OrgID > shall include a copyrighted, trademarked, or otherwise unique name that is owned by the
 283 business entity that is creating or defining the value or that is a registered ID assigned to the business
 284 entity by a recognized global authority. In addition, to ensure uniqueness, < OrgID > shall not contain
 285 a colon (:). When using this algorithm, the first colon to appear in the value shall appear between
 286 < OrgID > and < LocalID >. < LocalID > is chosen by the business entity and shall be used uniquely.

287 The values listed in the "IdentifyingDescriptions Value" column of Table 2 should be used as values for
 288 the IdentifyingDescriptions property. Every entry in Table 2 applicable for a given environment should be
 289 specified. An entry in Table 2 shall be used only if the value for the OtherIdentifyingInfo property is
 290 guaranteed to be globally unique across all underlying real-world systems.

291 **Table 2 – Predefined Identifiers for a Computer System**

IdentifyingDescriptions Value	OtherIdentifyingInfo Value
"CIM:GUID"	A globally unique identifier; see 7.1.1.1.
"CIM:MAC"	MAC address for one of the LAN interfaces of the system; see 7.1.1.2.
"CIM:Model:SerialNumber"	Model and serial number of the system; see 7.1.1.3.
"CIM:Tag"	Asset tag of the system; see 7.1.1.4.
"CIM:CorrelatableID"	An opaque identifier; see 7.1.1.5.

292 **7.1.1.1 CIM:GUID**

293 When the IdentifyingDescriptions property contains the value "CIM:GUID", the value of the corresponding
 294 array index of the OtherIdentifyingInfo property shall satisfy the following constraints:

- 295 • The value shall be a globally unique identifier for the system.
- 296 • The value shall match the pattern ("^[0..9A..F]{32}\$").
- 297 • The value should be the same value as the SMBIOS System UUID.

298 **7.1.1.2 CIM:MAC**

299 When the IdentifyingDescriptions property contains the value "CIM:MAC", the value of the corresponding
 300 array index of the OtherIdentifyingInfo property shall satisfy the following constraints:

- 301 • The value shall be the MAC address for one of the LAN interfaces of the system.
- 302 • The value shall be formatted as 12 contiguous uppercase hex digits (pattern
 303 "^[0123456789ABCDEF]{12}\$").
- 304 • When the [Ethernet Port Profile](#) is implemented, the value shall match the value of the
 305 PermanentAddress property of an instance of CIM_EthernetPort.

306 7.1.1.3 CIM:Model:SerialNumber

307 When the IdentifyingDescriptions property contains the value "CIM:Model:SerialNumber", the value of the
308 corresponding array index of the OtherIdentifyingInfo property shall be of the form < OrgID > : < LocalID >
309 : <Model Number> : <Serial Number>, where < OrgID > and < LocalID > are separated by a colon (:), and
310 where < OrgID > shall include a copyrighted, trademarked, or otherwise unique name that is owned by
311 the business entity that is creating or defining the value or that is a registered ID assigned to the business
312 entity by a recognized global authority. In addition, to ensure uniqueness, < OrgID > shall not contain a
313 colon (:). When using this algorithm, the first colon to appear in the value shall appear between < OrgID >
314 and < LocalID >. <LocalID> is chosen by the business entity and shall be used uniquely. <Model
315 Number> shall be the model number of the system, and <Serial Number> shall be the serial number of
316 the system.

317 7.1.1.4 CIM:Tag

318 An asset tag is a unique identifier assigned to a computer system. Generally this value is assigned by an
319 administrator or a client application.

320 When the IdentifyingDescriptions property contains the value "CIM:Tag", the value of the corresponding
321 array index of the OtherIdentifyingInfo property shall be a uniquely identifying tag of the system. An
322 example is an asset tag.

323 7.1.1.5 CIM:CorrelatableID

324 When the IdentifyingDescriptions property contains the value "CIM:CorrelatableID", the value of the
325 corresponding array index of the OtherIdentifyingInfo property shall contain an opaque ID that can be
326 used to correlate instances of CIM_ComputerSystem across namespace implementations that represent
327 the same underlying real-world system. Underlying instrumentation shall guarantee that this value is the
328 same for any two or more instances of CIM_ComputerSystem that represent the same underlying real-
329 world system.

330 7.1.2 Modifying ElementName Is Supported

331 The CIM_ComputerSystem.ElementName property may support being modified by the ModifyInstance
332 operation. See 8.4.1. This behavior is conditional upon the existence of an instance of
333 CIM_EnabledLogicalElementCapabilities being associated with the CIM_ComputerSystem instance
334 where the CIM_EnabledLogicalElementCapabilities.ElementNameEditSupported property has the value
335 TRUE.

336 This clause describes the CIM elements and behavior requirements when an implementation supports
337 client modification of the CIM_ComputerSystem.ElementName property.

338 7.1.2.1 CIM_EnabledLogicalElementCapabilities

339 An instance of CIM_EnabledLogicalElementCapabilities shall be associated with the
340 CIM_ComputerSystem instance through an instance of CIM_ElementCapabilities.

341 7.1.2.1.1 CIM_EnabledLogicalElementCapabilities.ElementNameEditSupported

342 The ElementNameEditSupported property shall have a value of TRUE when the implementation supports
343 client modification of the CIM_ComputerSystem.ElementName property.

344 7.1.2.1.2 CIM_EnabledLogicalElement.MaxElementNameLen

345 The MaxElementNameLen property shall be implemented.

346 7.1.3 Modifying ElementName Is Not Supported

347 This clause describes the CIM elements and behaviors that shall be implemented when the
348 CIM_ComputerSystem.ElementName property does not support being modified by the ModifyInstance
349 operation.

350 7.1.3.1 CIM_EnabledLogicalElementCapabilities

351 An instance of CIM_EnabledLogicalElementCapabilities may be associated with the
352 CIM_ComputerSystem instance through an instance of CIM_ElementCapabilities.

353 7.1.3.1.1 CIM_EnabledLogicalElementCapabilities.ElementNameEditSupported

354 The ElementNameEditSupported property shall have a value of FALSE when the implementation does
355 not support client modification of the CIM_ComputerSystem.ElementName property.

356 7.1.3.1.2 CIM_EnabledLogicalElement.MaxElementNameLen

357 The MaxElementNameLen property may be implemented. The MaxElementNameLen property is
358 irrelevant in this context.

359 7.1.4 Managing System Time

360 A system can maintain an internal clock, which provides the system with the current time (for example, to
361 provide time stamps for log entries). The management of the current time of the system may be
362 supported. This behavior is optional. See 8.2 for requirements for the ManageTime() method.

363 7.1.4.1 Managing System Time Is Supported

364 If the management of the current time of the system is supported, it should be implemented in
365 conformance with this profile. If the management of the current time of the system is supported in
366 conformance with this profile, the requirements specified in this clause shall be met.

367 An instance of CIM_TimeService shall be associated with the Central Instance through the
368 CIM_HostedService association. The instance of CIM_TimeService shall also be associated with the
369 Central Instance through the CIM_ServiceAffectsElement association. Management of system time is
370 supported when the CIM_TimeService.ManageTime() method is supported for at least one value for the
371 GetRequest parameter.

372 7.1.4.2 Managing System Time Is Not Supported

373 When the management of system time is not supported, no instance of CIM_TimeService shall be
374 associated with the Central Instance through the CIM_ServiceAffectsElement association.

375 7.2 Management of Computer System Components

376 The following subclauses detail the requirements for management of components of the system.

377 7.2.1 Instrumentation of Processors

378 If the processors of the system are instrumented, the instrumentation should be conformant with the [CPU](#)
379 [Profile](#). If the processors of the system are instrumented in conformance with the [CPU Profile](#), the Central
380 Instance of the *Computer System Profile* shall be associated with the Central Instance of the [CPU Profile](#)
381 through the CIM_SystemDevice association.

382 7.2.2 Instrumentation of System Memory

383 If the memory of the system is modeled, the [System Memory Profile](#) should be implemented. If the
384 system memory is modeled in conformance with the [System Memory Profile](#), the Central Instance of the

385 *Computer System Profile* shall be associated with the Central Instance of the [System Memory Profile](#)
386 through the CIM_SystemDevice association.

387 7.2.3 Instrumentation of Sensors

388 A system can contain one or more sensors that monitor components within the system. If the sensors of
389 the system are instrumented, the instrumentation should be conformant with the [Sensors Profile](#). If the
390 sensors of the system are instrumented in conformance with the [Sensors Profile](#), the Central Instance of
391 the *Computer System Profile* shall be associated with the Central Instance of the [Sensors Profile](#) through
392 the CIM_SystemDevice association.

393 7.3 Software Asset Management

394 This clause describes behavioral requirements for the management of software asset information for the
395 system.

396 7.3.1 Software Inventory Support

397 The inventory of software installed on or for the system may be modeled. If the inventory of software
398 installed on or for the system is modeled, the [Software Inventory Profile](#) should be implemented. If the
399 inventory of software installed on or for the system is modeled in conformance with the [Software](#)
400 [Inventory Profile](#), at least one instance of CIM_SoftwareIdentity shall be associated with the Central
401 Instance of the *Computer System Profile* through the CIM_InstalledSoftwareIdentity association, or
402 exactly one instance of CIM_SystemSpecificCollection shall be implemented in accordance with the
403 requirements specified in the "Representing Available Software" clause of the [Software Inventory Profile](#)
404 and associated with the Central Instance of the *Computer System Profile* through the
405 CIM_HostedCollection association.

406 7.3.2 Software Update Support

407 Management of software updates for the system or components contained in the system may be
408 supported. If the management of software updates for a component installed in the system is supported,
409 the [Software Update Profile](#) should be implemented. If the management of software updates for a
410 component installed in the system is supported in conformance with the [Software Update Profile](#), the
411 instance of a subclass of CIM_ManagedElement that represents the component shall be associated with
412 the Central Instance of the [Software Update Profile](#) through the CIM_ServiceAffectsElement association.

413 If the management of software updates for the system is supported in conformance with the [Software](#)
414 [Update Profile](#), the Central Instance of the *Computer System Profile* shall be associated with the Central
415 Instance of the [Software Update Profile](#) through the CIM_ServiceAffectsElement association.

416 If the system provides the ability to perform software updates for itself or other systems in conformance
417 with the [Software Update Profile](#), the Central Instance of the *Computer System Profile* shall be associated
418 with the Central Instance of the [Software Update Profile](#) through the CIM_HostedService association.

419 7.4 Network Interface Management

420 This clause describes the requirements for the management of network interfaces of the system.

421 7.4.1 Ethernet Interface Management

422 If the Ethernet interfaces of the system are instrumented, the instrumentation should be conformant with
423 the [Ethernet Port Profile](#). If the Ethernet Interfaces of the system are instrumented in conformance with
424 the [Ethernet Port Profile](#), at least one instance of CIM_EthernetPort shall be associated with the Central
425 Instance of the *Computer System Profile* through the CIM_SystemDevice association.

426 7.4.2 IP Interface Management

427 If the management of one or more IP interfaces of the system is supported, the [IP Interface Profile](#) should
428 be implemented. If the management of one or more IP interfaces of the system is supported in
429 conformance with the [IP Interface Profile](#), the Central Instance of the *Computer System Profile* shall be
430 associated with the Central Instance of the [IP Interface Profile](#) through the CIM_HostedAccessPoint
431 association.

432 If the system provides the optional behavior of managing alternate configurations for the IP interface in
433 conformance with the [IP Interface Profile](#), the instance of CIM_IPConfigurationService specified by the [IP
434 Interface Profile](#) shall be associated with the Central Instance of the *Computer System Profile* through the
435 CIM_HostedService association.

436 7.4.3 DHCP Client Management

437 If the DHCP client of the system is modeled, the [DHCP Client Profile](#) should be implemented. If the DHCP
438 client of the system is modeled in conformance with the [DHCP Client Profile](#), at least one instance of
439 CIM_DHCPProtocolEndpoint shall be associated with the Central Instance of the *Computer System
440 Profile* through the CIM_HostedAccessPoint association.

441 7.4.4 DNS Client Management

442 If the DNS client of the system is modeled, the [DNS Client Profile](#) should be implemented. If the DNS
443 client of the system is modeled in conformance with the [DNS Client Profile](#), at least one instance of
444 CIM_DNSProtocolEndpoint shall be associated with the Central Instance of the *Computer System Profile*
445 through the CIM_HostedAccessPoint association.

446 7.5 Record Logs

447 Error and event information about a system can be recorded in one or more record logs. If a record log
448 that contains information about the system is instrumented, the [Record Log Profile](#) should be
449 implemented. If a record log that contains information about a system is instrumented in conformance
450 with the [Record Log Profile](#), the Central Instance of the *Computer System Profile* shall be associated with
451 the Central Instance of the [Record Log Profile](#) through the CIM_UseOfLog association.

452 7.6 Management of Protocol Services

453 This clause describes behavioral requirements for the management of protocol services hosted on the
454 system.

455 7.6.1 Hosting a CLP Service

456 The system may host one or more CLP services. If the system hosts at least one CLP service, the [CLP
457 Service Profile](#) should be implemented. If a CLP service that is hosted by the system is modeled in
458 conformance with the [CLP Service Profile](#), the Central Instance of the *Computer System Profile* shall be
459 associated with the Central Instance of the [CLP Service Profile](#) through the CIM_HostedService
460 association.

461 7.6.2 Hosting an SSH Service

462 The system may host one or more SSH services. If the system hosts at least one SSH service, the [SSH
463 Service Profile](#) should be implemented. If a SSH service that is hosted by the system is modeled in
464 conformance with the [SSH Service Profile](#), the Central Instance of the *Computer System Profile* shall be
465 associated with the Central Instance of the [SSH Service Profile](#) through the CIM_HostedService
466 association.

467 7.6.3 Hosting a Telnet Service

468 The system may host one or more telnet services. If the system hosts at least one telnet service, the
469 [Telnet Service Profile](#) should be implemented. If a telnet service that is hosted by the system is modeled
470 in conformance with the [Telnet Service Profile](#), the Central Instance of the *Computer System Profile* shall
471 be associated with the Central Instance of the [Telnet Service Profile](#) through the CIM_HostedService
472 association.

473 7.7 System Lifecycle Management

474 The following subclauses detail requirements related to lifecycle management of the system.

475 7.7.1 System State Management

476 This clause details the requirements for representing and managing the state of a computer system.

477 7.7.1.1 Representing Current System State

478 The current state and last requested state for a computer system may be modeled using the
479 EnabledState and RequestedState properties of CIM_ComputerSystem. This behavior is optional.

480 When modeling system state is supported, the CIM_ComputerSystem.EnabledState property shall have a
481 value corresponding to the current state of the system and shall not have the value 12 (Not Applicable).
482 The CIM_ComputerSystem.RequestedState property shall not have the value 5 (Not Applicable). The
483 system state can change; therefore, the values of the RequestedState and EnabledState properties may
484 still change even when the optional behavior in 7.7.1.2 is not implemented.

485 When modeling system state is not supported, the CIM_ComputerSystem.EnabledState property shall
486 have the value 12 (Not Applicable) and the CIM_ComputerSystem.RequestedState property shall have
487 the value 5 (Not Applicable).

488 7.7.1.2 Client State Management Is Supported

489 The EnabledState and RequestedState properties and the RequestStateChange() method of
490 CIM_ComputerSystem are used to perform basic lifecycle and state management of abstract systems.
491 Common lifecycle states and state changes (for example, enable, disable, and reset) can be managed
492 using these CIM elements. Specializations of this profile define semantics for each state and state
493 change specific to the management domain targeted by the specializing profile.

494 When management of the state of a system is supported, exactly one instance of
495 CIM_EnabledLogicalElementCapabilities shall be associated with the CIM_ComputerSystem instance
496 through an instance of CIM_ElementCapabilities.

497 Even when client state management is supported, the values of the RequestedState and EnabledState
498 properties may still change implicitly to reflect state changes and requests that were not initiated by a
499 client of the instrumentation.

500 Support for managing the state of the system is optional behavior. This clause describes the CIM
501 elements and behaviors that shall be implemented when this behavior is supported.

502 7.7.1.2.1 CIM_EnabledLogicalElementCapabilities

503 When state management is supported, exactly one instance of CIM_EnabledLogicalElementCapabilities
504 shall be associated with the CIM_ComputerSystem instance through an instance of
505 CIM_ElementCapabilities.

506 7.7.1.2.1.1 CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported

507 The RequestedStatesSupported property may contain zero or more values.

508 7.7.1.2.2 CIM_ComputerSystem.RequestedState

509 When the CIM_ComputerSystem.RequestStateChange() method is successfully invoked, the value of the
510 RequestedState property shall be the value of the RequestedState parameter. If the method is not
511 successfully invoked, the value of the RequestedState property is indeterminate.

512 The CIM_ComputerSystem.RequestedState property shall have one of the values specified in the
513 CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported property or a value of 5 (No
514 Change). A value of 5 (No Change) shall indicate that the instrumentation is not aware of a request to
515 change the state of the managed system.

516 7.7.1.2.3 CIM_ComputerSystem.EnabledState

517 When the RequestedState parameter has a value of 2 (Enabled) or 3 (Disabled) and the
518 CIM_ComputerSystem.RequestStateChange() method completes successfully, the value of the
519 EnabledState property may equal the value of the CIM_ComputerSystem.RequestedState property.

520 If the method does not complete successfully, the value of the EnabledState property is indeterminate.

521 7.7.1.3 Client State Management Is Not Supported

522 This clause describes the CIM elements and behaviors that shall be implemented when client state
523 management is not supported.

524 7.7.1.3.1 CIM_EnabledLogicalElementCapabilities

525 When client state management is not supported, exactly one instance of
526 CIM_EnabledLogicalElementCapabilities may be associated with the CIM_ComputerSystem instance
527 through an instance of CIM_ElementCapabilities.

528 7.7.1.3.1.1 CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported

529 The CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported property shall not contain any
530 values.

531 7.7.2 Boot Control

532 This clause describes the behavioral requirements for modeling and managing the boot process and
533 configuration of the managed system.

534 7.7.2.1 Boot Configuration Management Is Not Supported

535 When management of boot configurations and the boot process is not supported for the system, the
536 managed system may initiate its boot process when it is enabled.

537 7.7.2.2 Boot Configuration Management Is Supported

538 Management of boot configurations and the boot process may be supported for the system. This clause
539 describes the requirements when the management of boot configurations and the boot process is
540 supported.

541 If the instrumentation of the boot configurations and the boot process is supported, the instrumentation
542 should be conformant with the [Boot Control Profile](#). If the instrumentation of the boot configurations and
543 the boot process is in conformance with the [Boot Control Profile](#), the Central Instance of the *Computer
544 System Profile* shall be associated with the Central Instance of the [Boot Control Profile](#) through the
545 CIM_ServiceAffectsElement association.

546 7.7.2.3 Hosting a Boot Service

547 The system may provide the ability to manage the boot configurations and control the boot process of
 548 itself or other systems. If the system provides this ability, the [Boot Control Profile](#) should be implemented.
 549 If the modeling of ability to manage the boot configurations and control the boot process of itself or other
 550 systems is in conformance with the [Boot Control Profile](#), the Central Instance of the *Computer System*
 551 *Profile* shall be associated with the Central Instance of the [Boot Control Profile](#) through the
 552 CIM_HostedService association.

553 7.8 SMASH Collections Profile

554 The [SMASH Collections Profile](#) may be implemented. If the [SMASH Collections Profile](#) is implemented,
 555 each instance of CIM_ConcreteCollection defined by the [SMASH Collections Profile](#) shall be associated
 556 with the Central Instance the *Computer System Profile* through the CIM_OwningCollectionElement
 557 association.

558 8 Methods

559 This clause details the requirements for supporting intrinsic operations and extrinsic methods for the CIM
 560 elements defined by this profile.

561 8.1 CIM_ComputerSystem.RequestStateChange()

562 Invoking the CIM_ComputerSystem.RequestStateChange() method changes the element's state to the
 563 value specified in the RequestedState parameter. The values 2 (Enabled) and 3 (Disabled) of the
 564 RequestedState parameter correspond to enabling or disabling the system. A value of 2 (Enabled) shall
 565 correspond to a request to enable the system. A value of 3 (Disabled) shall correspond to a request to
 566 disable the system. A value of 11 (Reset) shall be equivalent to invoking the method with a value of 3
 567 (Disabled), waiting for the operation to complete, and then invoking the method with a value of 2
 568 (Enabled).

569 See clause 7.7.1.2.2 for information about the effect of this method on the RequestedState property.

570 The method shall be considered successful if the (initiated) state of the system upon completion of the
 571 method corresponds to the desired state indicated by the RequestedState parameter. An actual change
 572 in state does not need to occur for the method to be considered successful; the resultant state only needs
 573 to be equal to the requested state. When the method completes successfully, the return value shall be
 574 zero.

575 See clause 7.7.1.2.3 for information about the effect of this method on the EnabledState property.

576 Detailed requirements of the RequestStateChange() method are specified in Table 3 and Table 4.

577 No standard messages are defined.

578 Invoking the CIM_ComputerSystem.RequestStateChange() method multiple times could result in earlier
 579 requests being overwritten or lost.

580 **Table 3 – CIM_ComputerSystem.RequestStateChange() Method: Return Code Values**

Value	Description
0	Request was successfully executed.
1	Method is unsupported in the implementation.
2	Error occurred
0x1000	Job started: REF returned to started CIM_ConcreteJob

581

Table 4 – CIM_ComputerSystem.RequestStateChange() Method: Parameters

Qualifiers	Name	Type	Description/Values
IN, REQ	RequestedState	uint16	Valid state values : 2 (Enabled) 3 (Disabled) 4 (Shutdown) 9 (Quiesce) 11 (Reset)
OUT	Job	CIM_ConcreteJob REF	Returned if job started
IN	TimeoutPeriod	datetime	Client specified the maximum amount of time the transition to a new state is supposed to take: 0 or NULL – No time requirements <interval> – Maximum time allowed

582 **8.1.1 CIM_ComputerSystem.RequestStateChange() Conditional Support**

583 When the CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported property contains at least
584 one value, the CIM_ComputerSystem.RequestStateChange() method shall be implemented and
585 supported. The CIM_ComputerSystem.RequestStateChange() method shall not return a value of 1 (Not
586 Supported).

587 **8.2 CIM_TimeService.ManageTime()**

588 The CIM_TimeService.ManageTime() method is used to query or modify the system time. When the
589 GetRequest parameter has a value of TRUE, the TimeData parameter shall be ignored. When the
590 ManagedElement parameter is not a reference to the Central Instance, the method shall return a value of
591 2. When the method is not supported for the specified value of GetRequest, the method shall return a
592 value of 2.

593 When the GetRequest parameter is TRUE and the method completes successfully, the value of the
594 TimeData parameter shall be an absolute date-time and shall not be an interval. When the value of the
595 GetRequest parameter is FALSE, and the TimeData parameter is expressed as an interval, the method
596 shall return a value of 2.

597 CIM_TimeService.ManageTime() method's detailed requirements shall be as specified in Table 5 and
598 Table 6.

599 No standard messages are defined for this method.

600 **Table 5 – CIM_TimeService.ManageTime() Method: Return Code Values**

Value	Description
0	Request was successfully executed.
1	Method is not supported in the implementation.
2	Error occurred

601

Table 6 – CIM_TimeService.ManageTime() Method: Parameters

Qualifiers	Name	Type	Description/Values
IN	GetRequest	Boolean	Indicates whether the request is to get (TRUE) or set (FALSE) the time for the specified element
IN / OUT	TimeData	datetime	On input, this is the desired value for the system time. On output, this is the system time.
IN	ManagedElement	CIM_Managed Element	Reference to the Central Instance

602 8.3 Profile Conventions for Operations

603 This profile specification defines operations in terms of [DSP0200](#).

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

606 The default list of operations is as follows:

- 607 • Associators()
- 608 • AssociatorNames()
- 609 • EnumerateInstances()
- 610 • EnumerateInstanceNames()
- 611 • GetInstance()
- 612 • References()
- 613 • ReferenceNames()

614 8.4 CIM_ComputerSystem

615 Table 7 lists implementation requirements for operations. If implemented, these operations shall be
616 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 7, all operations in
617 the default list in 8.3 shall be implemented as defined in [DSP0200](#).

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

619 **Table 7 – Operations: CIM_ComputerSystem**

Operation	Requirement	Messages
ModifyInstance	Optional. See 8.4.1.	None

620 8.4.1 CIM_ComputerSystem — ModifyInstance

621 This clause details the requirements for the ModifyInstance operation applied to an instance of
622 CIM_ComputerSystem. The ModifyInstance operation may be supported.

623 The ModifyInstance operation shall be supported and the CIM_ComputerSystem.ElementName property
624 shall be modifiable when an instance of CIM_EnabledLogicalElementCapabilities is associated with the
625 CIM_ComputerSystem instance and the ElementNameEditSupported property of the
626 CIM_EnabledLogicalElementCapabilities instance associated with the CIM_ComputerSystem instance
627 has a value of TRUE (see 8.4.1.1).

628 **8.4.1.1 CIM_ComputerSystem.ElementName**

629 When an instance of CIM_EnabledLogicalElementCapabilities is associated with the
 630 CIM_ComputerSystem instance and the ElementNameEditSupported property of the
 631 CIM_EnabledLogicalElementCapabilities instance associated with the CIM_ComputerSystem instance
 632 has a value of TRUE, the implementation shall allow the ModifyInstance operation to change the value of
 633 the ElementName property of the CIM_ComputerSystem instance. The ModifyInstance operation shall
 634 enforce the length restriction specified in the MaxElementNameLen property of the
 635 CIM_EnabledLogicalElementCapabilities instance.

636 When an instance of CIM_EnabledLogicalElementCapabilities is associated with the
 637 CIM_ComputerSystem instance and the ElementNameEditSupported property of the
 638 CIM_EnabledLogicalElementCapabilities has a value of FALSE or no instance of
 639 CIM_EnabledLogicalElementCapabilities is associated with the CIM_ComputerSystem instance, the
 640 implementation shall not allow the ModifyInstance operation to change the value of the ElementName
 641 property of the CIM_ComputerSystem instance.

642 **8.5 CIM_ElementCapabilities**

643 Table 8 lists implementation requirements for operations. If implemented, these operations shall be
 644 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 8, all operations in
 645 the default list in 8.3 shall be implemented as defined in [DSP0200](#).

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

647 **Table 8 – Operations: CIM_ElementCapabilities**

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

648 **8.6 CIM_EnabledLogicalElementCapabilities**

649 All operations in the default list in 8.3 shall be implemented as defined in [DSP0200](#).

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

651 **8.7 CIM_HostedService**

652 Table 9 lists implementation requirements for operations. If implemented, these operations shall be
 653 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 9, all operations in
 654 the default list in 8.3 shall be implemented as defined in [DSP0200](#).

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

656

Table 9 – Operations: CIM_HostedService

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

657 8.8 CIM_ServiceAffectsElement

658 Table 10 lists implementation requirements for operations. If implemented, these operations shall be
 659 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 10, all operations
 660 in the default list in 8.3 shall be implemented as defined in [DSP0200](#).

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

662

Table 10 – Operations: CIM_ServiceAffectsElement

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

663 8.9 CIM_TimeService

664 All operations in the default list in 8.3 shall be implemented as defined in [DSP0200](#).

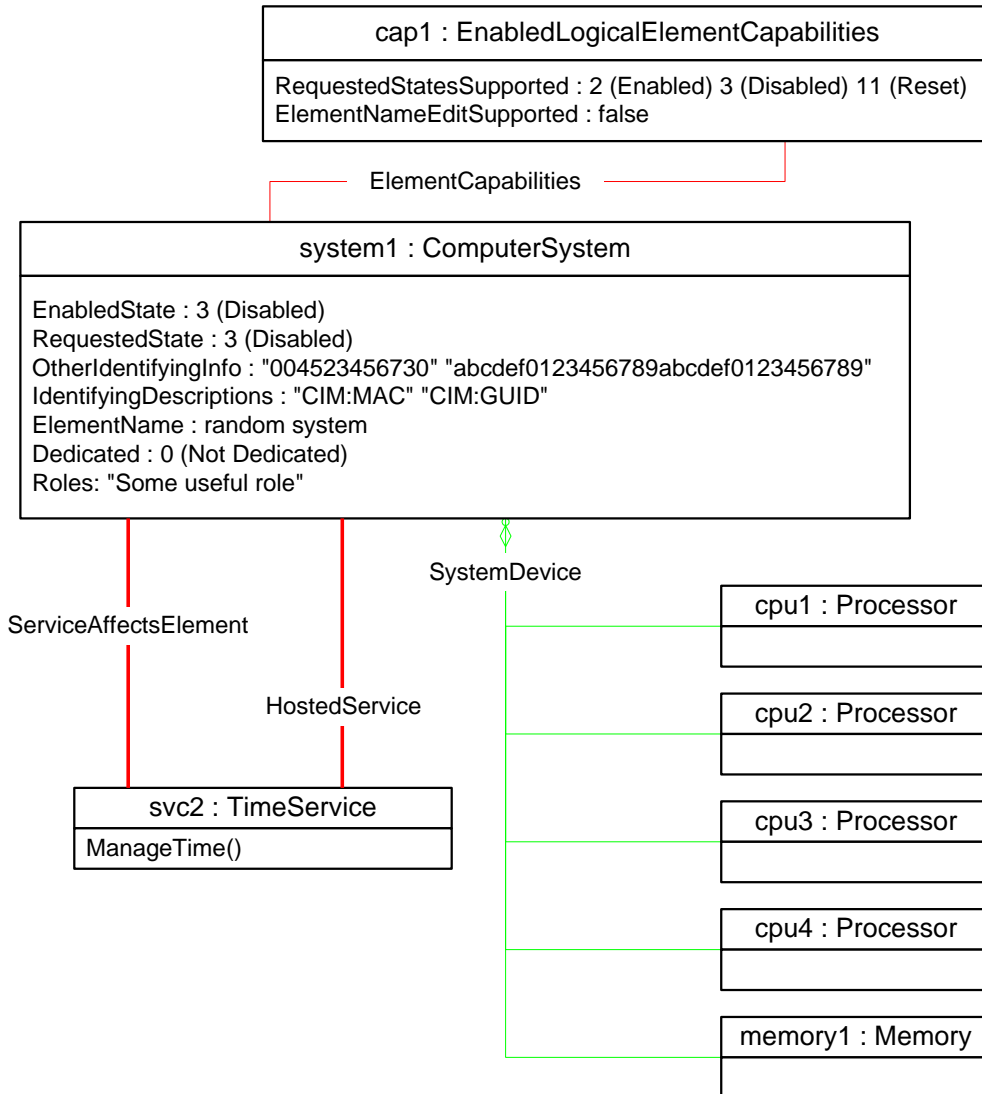
665 NOTE: Related profiles may define additional requirements on operations for the profile class.

666 9 Use Cases

667 The following use cases and object diagrams illustrate use of the *Computer System Profile*. They are for
 668 informational purposes only and do not introduce behavioral requirements for implementations of the
 669 profile.

670 9.1 Object Diagrams

671 The object diagram in Figure 2 shows an abstract system in which the optional state management and
 672 time management behaviors are supported as well as the [CPU Profile](#) and [System Memory Profile](#).

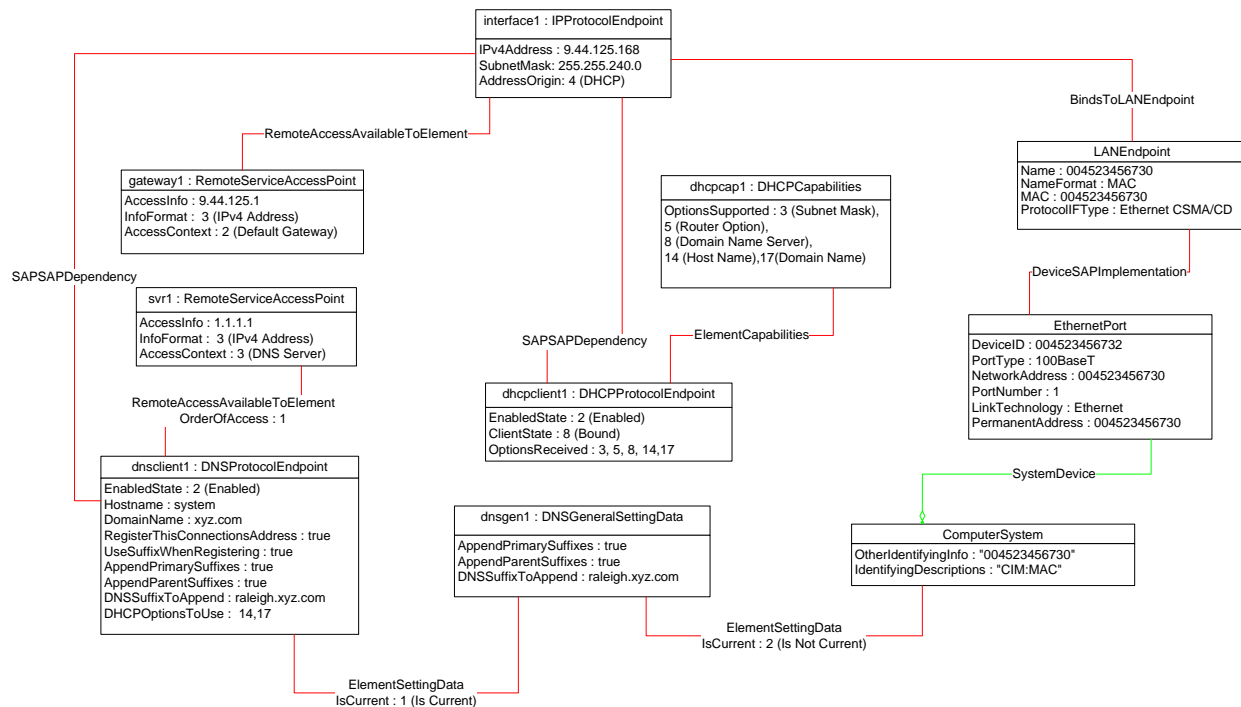


673

674

Figure 2 – Logical Topology

675 Figure 3 is an object diagram illustrating the network interfaces of the system: [Ethernet Port Profile](#), [IP](#)
 676 [Interface Profile](#), [DHCP Client Profile](#), and [DNS Client Profile](#). The system has a single network interface.



677

678

Figure 3 – Network Interfaces

679 9.2 Find a Dedicated System

680 Computer systems can have dedicated purposes or assigned roles. A client can find a system with a
 681 dedicated purpose or role by querying the value of the CIM_ComputerSystem.Dedicated or
 682 CIM_ComputerSystem.Roles property.

683 9.3 Correlate Instrumented Systems

684 For a given real system modeled with a specialization of the *Computer System Profile*, multiple
 685 implementations of the profile can exist to model the same real system within a single namespace, across
 686 namespaces, and across implementations. Across all of the namespaces to which the client has access
 687 to, starting with a single instance of CIM_ComputerSystem A that represents the real-world system, a
 688 client can find all of the other implementations of a specialization of the *Computer System Profile* that
 689 represent the same real system, as follows:

- 690 1) Form a set of identification pairs consisting of each pair of names and values from the
 691 IdentifyingDescriptions and OtherIdentifyingInfo properties of instance A.
- 692 2) For each CIM OM, query the Interop namespace to determine if the specialization of the
 693 *Computer System Profile* is advertised as instrumented.
- 694 3) If the specialization of the *Computer System Profile* has been instrumented, for the instance of
 695 CIM_RegisteredProfile that advertised it, find all instances of CIM_ComputerSystem associated
 696 through the CIM_ElementConformsToProfile association.
- 697 4) For each instance of CIM_ComputerSystem found in step 3), query the IdentifyingDescriptions
 698 and OtherIdentifyingInfo properties to determine if a name/value pair matches a name/value
 699 pair in the set of identification pairs found in step 1) for instance A.

- 700 5) If there is a match, then the instance of CIM_ComputerSystem from step 4) is instrumented for
701 the same real-world system as instance A. For each name/value pair for the instance, if it is not
702 already in the set of identification pairs known by the client for the system, add it to the set.
- 703 6) If a new identification pair was added in step 5), go back to step 4) and retest each instance of
704 CIM_ComputerSystem.

705 9.4 Enable a System

706 A client can enable a system as follows:

- 707 1) Look for an instance of CIM_EnabledLogicalElementCapabilities associated with the target
708 instance through the CIM_ElementCapabilities association.
- 709 2) Verify that the CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported property
710 contains the value 2 (Enabled).
- 711 3) Invoke the RequestStateChange() method on the target instance, specifying 2 (Enabled) for the
712 RequestedState parameter.

713 9.5 Disable a System

714 A client can disable a system as follows:

- 715 1) Look for an instance of CIM_EnabledLogicalElementCapabilities associated with the Central
716 Instance through the CIM_ElementCapabilities association.
- 717 2) Verify that the CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported property
718 contains the value 3 (Disabled).
- 719 3) Invoke the RequestStateChange() method on the target instance, specifying 3 (Disabled) for
720 the RequestedState parameter.

721 9.6 Reset a System

722 A client can reset a system as follows:

- 723 1) Look for an instance of CIM_EnabledLogicalElementCapabilities associated with the target
724 instance through the CIM_ElementCapabilities association.
- 725 2) Verify that the CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported property
726 contains the value 11 (Reset).
- 727 3) Invoke the RequestStateChange() method on the target instance, specifying 11 (Reset) for the
728 RequestedState parameter.

729 9.7 Manage the System Boot Configuration

730 A client can verify that an instance of CIM_RegisteredProfile for the [Boot Control Profile](#) exists using
731 either the central class or scoping class methodology as described in [Profile Registration Profile](#). If it
732 exists, a client can determine whether management of the system boot configuration is supported by
733 searching for an instance of CIM_BootService that is conformant with the [Boot Control Profile](#) and
734 associated with the Central Instance of the *Computer System Profile* through the
735 CIM_ServiceAffectsElement association. The specific use cases for managing the system boot
736 configuration are documented in the [Boot Control Profile](#).

737 9.8 Determine the Number of Processors in the System

738 A client can verify that an instance of CIM_RegisteredProfile for the [CPU Profile](#) exists using either the
739 central class or scoping class methodology as described in [Profile Registration Profile](#). If it exists, then the
740 CPU profile is implemented. When the optional [CPU Profile](#) is implemented, the client can determine the
741 number of processors in the system by querying for instances of CIM_Processor that are conformant with

742 the [CPU Profile](#) and associated with the Central Instance of the *Computer System Profile* through the
743 CIM_SystemDevice association.

744 **9.9 Determine If Time Management Is Supported**

745 To determine if time management is supported, the client can look for an instance of CIM_TimeService
746 associated with the target instance through the CIM_ServiceAffectsElement association.

747 **9.10 Get Time for System**

748 A client can determine the system time by first using the steps in 9.9 to determine if time management is
749 supported and find the associated instance of CIM_TimeService. The client can then invoke the
750 CIM_TimeService.ManageTime() method, specifying a value of TRUE for the value of the GetRequest
751 parameter and a reference to the target instance for the value of the ManagedElement parameter.

752 **9.11 Set Time for System**

753 A client can determine the system time by first using the steps in 9.9 to determine if time management is
754 supported and find the associated instance of CIM_TimeService. The client can then invoke the
755 CIM_TimeService.ManageTime() method, specifying a value of FALSE for the value of the GetRequest
756 parameter, the desired time for the value of the TimeData parameter, and a reference to the target
757 instance for the value of the ManagedElement parameter.

758 **9.12 Determining If ElementName Can Be Modified**

759 For a given instance of CIM_ComputerSystem, a client can determine whether the ElementName
760 property can be modified as follows:

- 761 1) Find the CIM_EnabledLogicalElementCapabilities instance that is associated with the target
762 instance.
- 763 2) If an instance of CIM_EnabledLogicalElementCapabilities is not found, client cannot modify the
764 ElementName property.
- 765 3) Query the value of the ElementNameEditSupported property of the
766 CIM_EnabledLogicalElementCapabilities instance. If the value is TRUE, the client can modify
767 the ElementName property of the target instance.

768 **9.13 Determining If State Management Is Supported**

769 For a given instance of CIM_ComputerSystem, a client can determine whether state management is
770 supported as follows:

- 771 1) Find the CIM_EnabledLogicalElementCapabilities instance that is associated with the target
772 instance.
- 773 2) If an instance of CIM_EnabledLogicalElementCapabilities is not found, state management is not
774 supported.
- 775 3) Query the value of the RequestedStatesSupported property. If at least one value is specified,
776 state management is supported.

777 **10 CIM Elements**

778 Table 11 shows the instances of CIM Elements for this profile. Instances of the CIM elements shall be
 779 implemented as described in Table 11. Clauses 7 (“Implementation”) and 8 (“Methods”) may impose
 780 additional requirements on these elements.

781 **Table 11 – CIM Elements: Computer System Profile**

Element Name	Requirement	Description
Classes		
CIM_ComputerSystem	Mandatory	See 10.1.
CIM_ElementCapabilities	Optional	See 10.2.
CIM_EnabledLogicalElementCapabilities	Optional	See 10.3.
CIM_HostedService	Optional	See 10.4.
CIM_ServiceAffectsElement	Optional	See 10.5.
CIM_TimeService	Optional	See 10.6.
Indications		
None defined in this profile		

782 **10.1 CIM_ComputerSystem**

783 An instance of CIM_ComputerSystem is used to represent the system. Table 12 contains the
 784 requirements for elements of this class.

785 **Table 12 – Class: CIM_ComputerSystem**

Elements	Requirement	Description
Name	Mandatory	Key
CreationClassName	Mandatory	Key
OtherIdentifyingInfo	Optional	See 7.1.1.
IdentifyingDescriptions	Optional	See 7.1.1.
EnabledState	Mandatory	See 7.7.1.
RequestedState	Mandatory	See 7.7.1.2.2.
OperationalStatus	Mandatory	None
HealthState	Mandatory	None
ElementName	Mandatory	See 7.1.2 and 7.1.3.
RequestStateChange()	Conditional	See 8.1.

786 **10.2 CIM_ElementCapabilities**

787 CIM_ElementCapabilities associates an instance of CIM_EnabledLogicalElementCapabilities with an
 788 instance of CIM_ComputerSystem. Table 13 contains the requirements for elements of this class.

789 **Table 13 – Class: CIM_ElementCapabilities**

Elements	Requirement	Notes
ManagedElement	Mandatory	This property shall be a reference to an instance of CIM_ComputerSystem. Cardinality 1..*
Capabilities	Mandatory	This property shall be a reference to the instance of CIM_EnabledLogicalElementCapabilities. Cardinality 0..1

790 **10.3 CIM_EnabledLogicalElementCapabilities**

791 CIM_EnabledLogicalElementCapabilities indicates support for managing the state of the system.
 792 Table 14 contains the requirements for elements of this class.

793 **Table 14 – Class: CIM_EnabledLogicalElementCapabilities**

Elements	Requirement	Notes
InstanceID	Mandatory	Key
RequestedStatesSupported	Mandatory	See 7.7.1.2.1.1 and 7.7.1.3.1.1.
ElementNameEditSupported	Mandatory	See 7.1.2.1.1 and 7.1.3.1.1.
MaxElementNameLen	Conditional	See 7.1.2.1.2 and 7.1.3.1.2.

794 **10.4 CIM_HostedService**

795 CIM_HostedService relates the CIM_TimeService to its scoping CIM_ComputerSystem instance.
 796 Table 15 contains the requirements for elements of this class.

797 **Table 15 – Class: CIM_HostedService**

Elements	Requirement	Notes
Antecedent	Mandatory	This property shall reference the Central Instance. Cardinality 1
Dependent	Mandatory	This property shall reference CIM_TimeService. Cardinality 0..1

798 **10.5 CIM_ServiceAffectsElement**

799 CIM_ServiceAffectsElement associates the CIM_TimeService instance with the Central Instance.
 800 Table 16 contains the requirements for elements of this class.

801 **Table 16 – Class: CIM_ServiceAffectsElement**

Elements	Requirement	Notes
AffectedElement	Mandatory	This property shall be a reference to the Central Instance. Cardinality 1
AffectingElement	Mandatory	This property shall be a reference to an instance of CIM_TimeService. Cardinality 0..1
ElementEffects	Mandatory	Matches 5 (Manages)

802 **10.6 CIM_TimeService**

803 CIM_TimeService manages the current time on the system. Table 17 contains the requirements for
 804 elements of this class.

805 **Table 17 – Class: CIM_TimeService**

Elements	Requirement	Notes
SystemCreationClassName	Mandatory	Key
SystemName	Mandatory	Key
CreationClassName	Mandatory	Key
Name	Mandatory	Key
ElementName	Mandatory	Pattern (“.*”). See clauses 7 and 8.
ManageTime()	Mandatory	See 8.2.

806
807
808
809**ANNEX A
(Informative)****Change Log**

Version	Date	Description
1.0.0	2008-12-08	
1.0.1	2010-04-22	Released as DMTF Standard. This errata release ensures that other profiles can reference the ComputerSystem profile and corrects a wrong association used in a diagram. Experimental Qualifiers have been removed for classes and profiles that have gone Final or been released as DMTF Standard.
1.0.2	2013-01-24	This errata addresses semantics of EnabledState and RequestedState properties. Experimental Qualifiers have been removed for CLP Service Profile and for Software Update Profile, which have gone Final or have been released as DMTF Standard.
1.0.3	2014-05-22	This errata adds additional supported RequestedState values and recommended mapping of CIM:GUID identifying information to SMBIOS UUID.

810