



1

2

3

4

**Document Number: DSP1004**

**Date: 2009-06-16**

**Version: 1.0.0**

5 **Base Server Profile**

6 **Document Type: Specification**

7 **Document Status: DMTF Standard**

8 **Document Language: E**

## 9 Copyright Notice

10 Copyright © 2006, 2009 Distributed Management Task Force, Inc. (DMTF). All rights reserved.

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

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25 withdrawn or modified after publication, and shall be indemnified and held harmless by any party  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

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

31

32

# CONTENTS

33 Foreword ..... 5

34 Introduction ..... 6

35 1 Scope ..... 7

36 2 Normative References..... 7

37 2.1 Approved References ..... 7

38 2.2 Other References..... 8

39 3 Terms and Definitions ..... 8

40 4 Symbols and Abbreviated Terms ..... 10

41 5 Synopsis ..... 10

42 6 Description ..... 10

43 6.1 Representation of System Power State..... 11

44 7 Implementation..... 12

45 7.1 Base Server System ..... 12

46 7.2 Management of Base Server Components ..... 13

47 7.3 State Management..... 13

48 7.4 Text Console Redirection (Optional)..... 16

49 8 Methods..... 17

50 9 Use Cases ..... 17

51 9.1 Object Diagrams ..... 17

52 9.2 Determine the System Model and Serial Number ..... 23

53 9.3 Power on a System..... 24

54 9.4 Power off a System..... 24

55 9.5 Shutdown and Restart a System ..... 24

56 9.6 Perform System Power Control ..... 24

57 9.7 Determining the System Power State..... 24

58 9.8 Determine the Number of Processors in the System ..... 25

59 9.9 Determine the Number of Processors that the System Can Hold ..... 25

60 10 CIM Elements ..... 25

61 10.1 CIM\_ComputerSystem..... 26

62 10.2 CIM\_ComputerSystemPackage ..... 26

63 10.3 CIM\_EnabledLogicalElementCapabilities..... 26

64 10.4 CIM\_PhysicalPackage ..... 26

65 10.5 CIM\_RegisteredProfile..... 27

66 ANNEX A (informative) Change Log..... 28

67

## 68 Figures

69 Figure 1 – Base Server Profile: Class Diagram ..... 11

70 Figure 2 – Profile Registration..... 17

71 Figure 3 – Power Management and Time Service ..... 18

72 Figure 4 – Advanced Power Management..... 19

73 Figure 5 – Network Interfaces..... 20

74 Figure 6 – Logical Topology..... 21

75 Figure 7 – Physical Topology..... 22

76 Figure 8 – Logical to Physical Mapping ..... 23

77

78 **Tables**

79	Table 1 – Referenced Profiles .....	10
80	Table 2 – EnabledState and ACPI State Equivalence .....	13
81	Table 3 – PowerState and EnabledState Values (Required Equivalence) .....	15
82	Table 4 – EnabledState and PowerState Values (Recommended Equivalence) .....	15
83	Table 5 – RequestedState and RequestedPowerState Values .....	15
84	Table 6 – RequestedStatesSupported and PowerStatesSupported Values .....	16
85	Table 7 – CIM Elements: Base Server Profile .....	25
86	Table 8 – Class: CIM_ComputerSystem .....	26
87	Table 9 – Class: CIM_ComputerSystemPackage .....	26
88	Table 10 – Class: CIM_EnabledLogicalElementCapabilities .....	26
89	Table 11 – Class: CIM_RegisteredProfile .....	27
90		

91

## Foreword

92 The *Base Server Profile* (DSP1004) was prepared by the Server Management Working Group and the  
93 Physical Platform Profiles Working Group of the DMTF.

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

96

## 97 **Acknowledgments**

98 The authors wish to acknowledge the following people.

### 99 **Editors:**

- 100 • Christina Shaw – HP
- 101 • Aaron Merkin – IBM

### 102 **Contributors:**

- 103 • Jon Hass – Dell
- 104 • Khachatur Papanyan – Dell
- 105 • Jeff Hilland – HP
- 106 • Christina Shaw – HP
- 107 • Aaron Merkin – IBM
- 108 • Perry Vincent – Intel
- 109 • John Leung – Intel

110

111

## Introduction

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

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

118

# Base Server Profile

## 119 1 Scope

120 The *Base Server Profile* is the autonomous profile that defines the classes used to describe basic server  
121 hardware and its related software. The scope of this profile is limited to simple servers that are directly  
122 realized in physical components. The profiles referenced by the *Base Server Profile* extend the  
123 management capabilities by adding the capability to represent server configuration, boot control,  
124 provisioning, and hardware.

## 125 2 Normative References

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

### 129 2.1 Approved References

- 130 DMTF DSP0004, *CIM Infrastructure Specification 2.5*,  
131 [http://www.dmtf.org/standards/published\\_documents/DSP0004\\_2.5.pdf](http://www.dmtf.org/standards/published_documents/DSP0004_2.5.pdf)
- 132 DMTF DSP0200, *CIM Operations over HTTP 1.3*,  
133 [http://www.dmtf.org/standards/published\\_documents/DSP0200\\_1.3.pdf](http://www.dmtf.org/standards/published_documents/DSP0200_1.3.pdf)
- 134 DMTF DSP1001, *Management Profile Specification Usage Guide 1.0*,  
135 [http://www.dmtf.org/standards/published\\_documents/DSP1001\\_1.0.pdf](http://www.dmtf.org/standards/published_documents/DSP1001_1.0.pdf)
- 136 DMTF DSP1005, *CLP Service Profile 1.0*,  
137 [http://www.dmtf.org/standards/published\\_documents/DSP1005\\_1.0.pdf](http://www.dmtf.org/standards/published_documents/DSP1005_1.0.pdf)
- 138 DMTF DSP1006, *SMASH Collections Profile 1.0*,  
139 [http://www.dmtf.org/standards/published\\_documents/DSP1006\\_1.0.pdf](http://www.dmtf.org/standards/published_documents/DSP1006_1.0.pdf)
- 140 DMTF DSP1009, *Sensors Profile 1.0*,  
141 [http://www.dmtf.org/standards/published\\_documents/DSP1009\\_1.0.pdf](http://www.dmtf.org/standards/published_documents/DSP1009_1.0.pdf)
- 142 DMTF DSP1010, *Record Log Profile 1.0*,  
143 [http://www.dmtf.org/standards/published\\_documents/DSP1010\\_1.0.pdf](http://www.dmtf.org/standards/published_documents/DSP1010_1.0.pdf)
- 144 DMTF DSP1011, *Physical Asset Profile 1.0*,  
145 [http://www.dmtf.org/standards/published\\_documents/DSP1011\\_1.0.pdf](http://www.dmtf.org/standards/published_documents/DSP1011_1.0.pdf)
- 146 DMTF DSP1012, *Boot Control Profile 1.0*,  
147 [http://www.dmtf.org/standards/published\\_documents/DSP1012\\_1.0.pdf](http://www.dmtf.org/standards/published_documents/DSP1012_1.0.pdf)
- 148 DMTF DSP1013, *Fan Profile 1.0*, [http://www.dmtf.org/standards/published\\_documents/DSP1013\\_1.0.pdf](http://www.dmtf.org/standards/published_documents/DSP1013_1.0.pdf)
- 149 DMTF DSP1014, *Ethernet Port Profile 1.0*,  
150 [http://www.dmtf.org/standards/published\\_documents/DSP1014\\_1.0.pdf](http://www.dmtf.org/standards/published_documents/DSP1014_1.0.pdf)
- 151 DMTF DSP1015, *Power Supply Profile 1.0*,  
152 [http://www.dmtf.org/standards/published\\_documents/DSP1015\\_1.0.pdf](http://www.dmtf.org/standards/published_documents/DSP1015_1.0.pdf)
- 153 DMTF DSP1016, *Telnet Service Profile 1.0*,  
154 [http://www.dmtf.org/standards/published\\_documents/DSP1016\\_1.0.pdf](http://www.dmtf.org/standards/published_documents/DSP1016_1.0.pdf)

- 155 DMTF DSP1017, *SSH Service Profile 1.0*,  
156 [http://www.dmtf.org/standards/published\\_documents/DSP1017\\_1.0.pdf](http://www.dmtf.org/standards/published_documents/DSP1017_1.0.pdf)
- 157 DMTF DSP1022, *CPU Profile 1.0*,  
158 [http://www.dmtf.org/standards/published\\_documents/DSP1022\\_1.0.pdf](http://www.dmtf.org/standards/published_documents/DSP1022_1.0.pdf)
- 159 DMTF DSP1023, *Firmware Inventory Profile 1.0*,  
160 [http://www.dmtf.org/standards/published\\_documents/DSP1023\\_1.0.pdf](http://www.dmtf.org/standards/published_documents/DSP1023_1.0.pdf)
- 161 DMTF DSP1024, *Text Console Redirection Profile 1.0*,  
162 [http://www.dmtf.org/standards/published\\_documents/DSP1024\\_1.0.pdf](http://www.dmtf.org/standards/published_documents/DSP1024_1.0.pdf)
- 163 DMTF DSP1025, *Firmware Update Profile 1.0*,  
164 [http://www.dmtf.org/standards/published\\_documents/DSP1025\\_1.0.pdf](http://www.dmtf.org/standards/published_documents/DSP1025_1.0.pdf)
- 165 DMTF DSP1026, *System Memory Profile 1.0*,  
166 [http://www.dmtf.org/standards/published\\_documents/DSP1026\\_1.0.pdf](http://www.dmtf.org/standards/published_documents/DSP1026_1.0.pdf)
- 167 DMTF DSP1027, *Power State Management Profile 1.0*,  
168 [http://www.dmtf.org/standards/published\\_documents/DSP1027\\_1.0.pdf](http://www.dmtf.org/standards/published_documents/DSP1027_1.0.pdf)
- 169 DMTF DSP1033, *Profile Registration Profile 1.0*,  
170 [http://www.dmtf.org/standards/published\\_documents/DSP1033\\_1.0.pdf](http://www.dmtf.org/standards/published_documents/DSP1033_1.0.pdf)
- 171 DMTF DSP1036, *IP Interface Profile 1.0*,  
172 [http://www.dmtf.org/standards/published\\_documents/DSP1036\\_1.0.pdf](http://www.dmtf.org/standards/published_documents/DSP1036_1.0.pdf)
- 173 DMTF DSP1037, *DHCP Client Profile 1.0*,  
174 [http://www.dmtf.org/standards/published\\_documents/DSP1037\\_1.0.pdf](http://www.dmtf.org/standards/published_documents/DSP1037_1.0.pdf)
- 175 DMTF DSP1038, *DNS Client Profile 1.0*,  
176 [http://www.dmtf.org/standards/published\\_documents/DSP1038\\_1.0.pdf](http://www.dmtf.org/standards/published_documents/DSP1038_1.0.pdf)
- 177 DMTF DSP1052, *Computer System Profile 1.0*,  
178 [http://www.dmtf.org/standards/published\\_documents/DSP1052\\_1.0.pdf](http://www.dmtf.org/standards/published_documents/DSP1052_1.0.pdf)
- 179 *Advanced Configuration and Power Interface Specification (ACPI Specification)*, revision 3.0,  
180 [www.acpi.info/Downloads/ACPIspec30.pdf](http://www.acpi.info/Downloads/ACPIspec30.pdf)

## 181 **2.2 Other References**

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

## 184 **3 Terms and Definitions**

- 185 For the purposes of this document, the following terms and definitions apply. For the purposes of this  
186 document, the terms and definitions given in [DSP1033](#) and [DSP1001](#) also apply.

### 187 **3.1**

#### 188 **can**

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

### 190 **3.2**

#### 191 **cannot**

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



- 193 **3.3**  
194 **conditional**  
195 indicates requirements to be followed strictly to conform to the document when the specified conditions  
196 are met
- 197 **3.4**  
198 **mandatory**  
199 indicates requirements to be followed strictly to conform to the document and from which no deviation is  
200 permitted
- 201 **3.5**  
202 **may**  
203 indicates a course of action permissible within the limits of the document
- 204 **3.6**  
205 **need not**  
206 indicates a course of action permissible within the limits of the document
- 207 **3.7**  
208 **optional**  
209 indicates a course of action permissible within the limits of the document
- 210 **3.8**  
211 **referencing profile**  
212 indicates a profile that owns the definition of this class and can include a reference to this profile in its  
213 "Referenced Profiles" table
- 214 **3.9**  
215 **shall**  
216 indicates requirements to be followed strictly to conform to the document and from which no deviation is  
217 permitted
- 218 **3.10**  
219 **shall not**  
220 indicates requirements to be followed strictly to conform to the document and from which no deviation is  
221 permitted
- 222 **3.11**  
223 **should**  
224 indicates that among several possibilities, one is recommended as particularly suitable, without  
225 mentioning or excluding others, or that a certain course of action is preferred but not necessarily required
- 226 **3.12**  
227 **should not**  
228 indicates that a certain possibility or course of action is deprecated but not prohibited
- 229 **3.13**  
230 **unspecified**  
231 indicates that this profile does not define any constraints for the referenced CIM element or operation

## 232 4 Symbols and Abbreviated Terms

233 The following abbreviations are used in this document.

### 234 4.1

#### 235 ACPI

236 Advanced Configuration and Power Interface

## 237 5 Synopsis

238 **Profile Name:** Base Server

239 **Version:** 1.0.0

240 **Organization:** DMTF

241 **CIM schema version:** 2.13

242 **Specializes:** DMTF *Computer System Profile* 1.0

243 **Central Class:** CIM\_ComputerSystem

244 **Scoping Class:** CIM\_ComputerSystem

245 The *Base Server Profile* is an autonomous profile that provides the capability to manage simple server  
246 hardware and related software.

247 The Central Class of the *Base Server Profile* shall be CIM\_ComputerSystem. The Central Instance shall  
248 be an instance of CIM\_ComputerSystem. The Scoping Class shall be CIM\_ComputerSystem. The  
249 Scoping Instance shall be the Central Instance. Table 1 lists profiles upon which this profile has a  
250 dependency.

251 **Table 1 – Referenced Profiles**

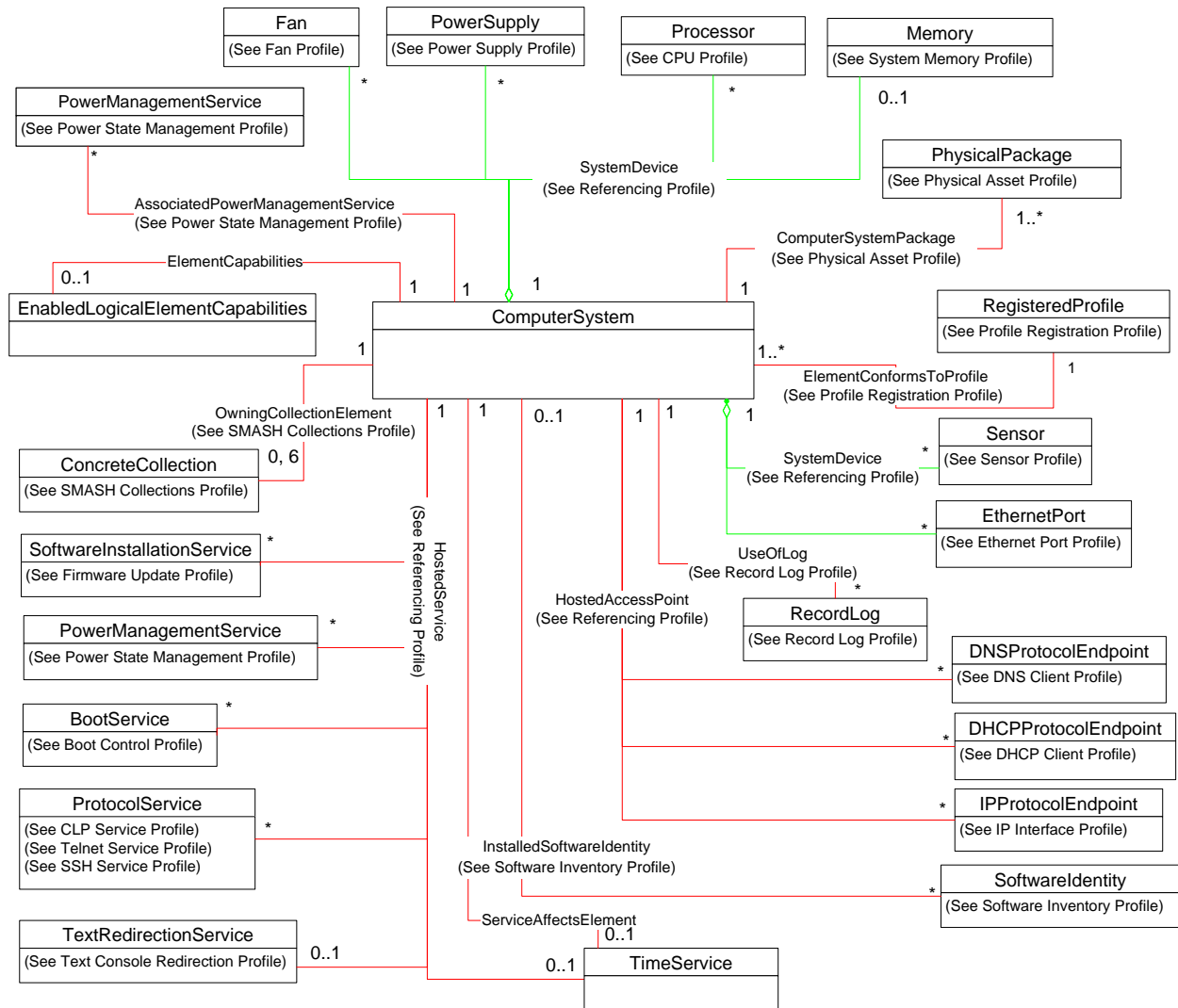
Profile Name	Organization	Version	Relationship	Behavior
<a href="#">Computer System</a>	DMTF	1.0	Specializes	None
<a href="#">Fan Profile</a>	DMTF	1.0	Optional	See section 7.2.1.
<a href="#">Physical Asset</a>	DMTF	1.0	Mandatory	See section 7.1.2.
<a href="#">Power State Management</a>	DMTF	1.0	Optional	See section 7.3.2.
<a href="#">Power Supply</a>	DMTF	1.0	Optional	See section 7.2.2.
<a href="#">Profile Registration</a>	DMTF	1.0	Mandatory	None
<a href="#">Text Console Redirection</a>	DMTF	1.0	Optional	See section 7.4.

## 252 6 Description

253 The *Base Server Profile* is an autonomous profile that defines the minimum top-level object model  
254 needed to model simple server hardware and related software. Other profiles add additional management  
255 objects to this basic server model to provide system configuration, boot control, and other provisioning  
256 capabilities. CIM\_ComputerSystem represents the server system. CIM\_TimeService provides the ability  
257 to manage the system time.

258 Figure 1 presents the class schema for the *Base Server Profile*. For simplicity, the prefix CIM\_ has been  
259 removed from the names of the classes.

260 Note that the behavioral constraints for many of the profiles identified in Figure 1 are inherited from the  
 261 specialized [Computer System Profile](#). Therefore, although they are shown in Figure 1, they are not  
 262 referenced in this specification. Examples are the [IP Interface Profile](#), [Ethernet Port Profile](#), and [Record  
 263 Log Profile](#).



264

265

Figure 1 – Base Server Profile: Class Diagram

266 **6.1 Representation of System Power State**

267 Normative requirements for the representation of system power state are expressed in section 7.3. The  
 268 following informative text provides background on the approach taken to modeling system power state.

269 The *Base Server Profile* identifies two complementary approaches to representing the power state of a  
 270 base server: simple on/off management through the RequestedState and EnabledState properties, and  
 271 the RequestStateChange() method. Definitions are given for the 2 (Enabled) and 3 (Disabled) values for  
 272 the EnabledState property in terms of industry standard ACPI definitions. Alternately, if an implementation  
 273 wants to support more granular or complex power-management behavior, the [Power State Management  
 274 Profile](#) can be implemented.

275 The power-management behavior and system power states specified in the [Power State Management](#)  
276 [Profile](#) are a superset of the function and states that are represented using the EnabledState and  
277 RequestedState properties of CIM\_ComputerSystem. That is, the EnabledState and RequestedState  
278 properties are sufficient to represent ACPI states S0 and S5. Implementing the [Power State Management](#)  
279 [Profile](#) provides the ability to represent additional ACPI states. For example, the equivalency between the  
280 EnabledState and PowerState values results from their mapping to identical ACPI states rather than that  
281 they are defined in terms of each other. For the subset of values for the EnabledState and  
282 RequestedState properties for which ACPI states are defined, there is a one-to-one correspondence with  
283 a legal value for the PowerState and RequestedPowerState properties.

284 Defining the states expressible through the [Power State Management Profile](#) as a superset of those  
285 states possible with the EnabledState and RequestedState properties is contrasted with the discarded  
286 alternative of using the implementation of the [Power State Management Profile](#) to provide a refinement of  
287 the interpretation of the EnabledState and RequestedState values. If this latter, discarded approach were  
288 taken, multiple values of PowerState and RequestedPowerState would be mapped to the less granular  
289 values for the EnabledState and RequestedState properties.

## 290 **7 Implementation**

291 The *Base Server Profile* consists of definitions for the CIM\_ComputerSystem, CIM\_PhysicalPackage, and  
292 CIM\_TimeService classes, and their related EnabledLogicalElementCapabilities classes. Other related  
293 subsystem classes such as CIM\_LogicalDevice, CIM\_Collection, and CIM\_RecordLog are defined in their  
294 respective profiles.

295 Requirements for propagating and formulating certain properties of the *Base Server Profile* classes are  
296 discussed in this section. The *Base Server Profile* is divided into two areas of functionality: the logical  
297 aspects of the server system and its physical aspects. This profile defines how to model the system's  
298 logical aspects, and the *Physical Asset Profile* defines how to model its physical aspects.

299 Methods are described in section 8 ("Methods"), and properties are described in section 10 ("CIM  
300 Elements").

### 301 **7.1 Base Server System**

302 The instrumentation shall create an instance of CIM\_ComputerSystem to represent the system being  
303 modeled.

#### 304 **7.1.1 Identifying a Base Server**

305 This section details the constraints beyond those specified in the [Computer System Profile](#) for using the  
306 IdentifyingDescriptions and OtherIdentifyingInfo properties to identify a computer system.

##### 307 **7.1.1.1 CIM:GUID**

308 The value of the OtherIdentifyingInfo property shall match the value of the  
309 CIM\_ComputerSystemPackage.PlatformGUID property for an instance of CIM\_ComputerSystemPackage  
310 that references the Central Instance.

##### 311 **7.1.1.2 CIM:Model:SerialNumber**

312 The value of the OtherIdentifyingInfo property shall match the value of the Model property of an instance  
313 of CIM\_PhysicalPackage, concatenated with a single colon (:), concatenated with the value of the  
314 SerialNumber property of the same instance of CIM\_PhysicalPackage.

315 **7.1.1.3 CIM:Tag**

316 The value of the OtherIdentifyingInfo property shall match the value of the Tag property of an instance of  
317 CIM\_PhysicalPackage.

318 **7.1.2 Representing the Physical Packaging**

319 The physical packaging for a system shall be modeled according to the requirements specified in the  
320 [Physical Asset Profile](#). At least one instance of CIM\_PhysicalPackage shall be associated with the  
321 Central Instance through the CIM\_ComputerSystemPackage association.

322 **7.2 Management of Base Server Components**

323 The following subclauses detail the requirements for management of components of the system in  
324 addition to those specified in the [Computer System Profile](#).

325 **7.2.1 Instrumentation of Fans (Optional)**

326 A system can contain one or more fans that provide cooling for the system. When the fans of the system  
327 are instrumented, the instrumentation shall be conformant with the [Fan Profile](#), and the Central Instance  
328 of the *Base Server Profile* shall be associated with the Central Instance of the [Fan Profile](#) through the  
329 CIM\_SystemDevice association.

330 **7.2.2 Instrumentation of Power Supplies (Optional)**

331 A system can contain one or more power supplies that provide power to the system. When the power  
332 supplies of the system are instrumented, the instrumentation shall be conformant with the [Power Supply](#)  
333 [Profile](#), and the Central Instance of the *Base Server Profile* shall be associated with the Central Instance  
334 of the [Power Supply Profile](#) through the CIM\_SystemDevice association.

335 **7.3 State Management**

336 This section details further constraints related to state management beyond those specified in the  
337 [Computer System Profile](#).

338 **7.3.1 Correspondence of System States and ACPI States**

339 The EnabledState property of CIM\_ComputerSystem is defined in terms of ACPI values to provide  
340 meaningful context for the interpretation of values for a computer system realized in hardware. The  
341 mappings specified in Table 2 shall be used. It is not necessary for the underlying modeled system to  
342 support the ACPI specification.

343 **Table 2 – EnabledState and ACPI State Equivalence**

CIM_ComputerSystem.EnabledState Value	Corresponding ACPI State
2 (Enabled)	G0 or S0 Working
3 (Disabled)	G2 or S5
9 (Quiesce)	G1, S1, S2, S3, or S4

344 **7.3.2 Power State Management**

345 The [Power State Management Profile](#) may be supported because the Central Instance either hosts an  
346 instance of CIM\_PowerManagementService or has the functionality of one available to it.

### 347 7.3.2.1 Power Management Available to System

348 Management of the power state of the system may be supported for the system. When the management  
349 of the power state is supported, the [Power State Management Profile](#) shall be implemented and the  
350 Central Instance of the *Base Server Profile* shall be associated with the Central Instance of the [Power  
351 State Management Profile](#) through the CIM\_AssociatedPowerManagementService association.

### 352 7.3.2.2 Power Management Hosted on System

353 The system may provide the ability to manage the power state of itself or other systems. When the  
354 system provides this ability, the [Power State Management Profile](#) shall be implemented and the Central  
355 Instance of the *Base Server Profile* shall be associated with the Central Instance of the [Power State  
356 Management Profile](#) through the CIM\_HostedService association.

## 357 7.3.3 Relationship between State Management and Power State Management

358 The behavior in this section is conditional on the implementation of the behavior in section 7.3.2.1. When  
359 the optional behavior specified in section 7.3.2.1 is supported, the state management behavior specified  
360 in section "State Management Is Supported (Conditional)" of the [Computer System Profile](#) shall be  
361 supported.

362 Management of the power state may be supported for a system. One reason for supporting power state  
363 management is the need to provide more granular management beyond that available through state  
364 management. To ensure consistent semantics for state management regardless of whether power state  
365 management is supported, it is necessary to establish constraints on the interaction of power state  
366 management and state management when power state management is supported. This section details  
367 these constraints.

368 Note that the CIM\_ComputerSystem.RequestStateChange() method defined in the [Computer System  
369 Profile](#) causes the values for the CIM\_ComputerSystem.EnabledState and  
370 CIM\_ComputerSystem.RequestedState properties to change. Due to the equivalence requirements  
371 stated below, the possible invocation of the method will result in changes to the values of the  
372 CIM\_AssociatedPowerManagementService.RequestedPowerState and  
373 CIM\_AssociatedPowerManagementService.PowerState properties. Likewise, the  
374 CIM\_PowerManagementService.RequestPowerStateChange() method defined in the [Power State  
375 Management Profile](#) will cause the CIM\_AssociatedPowerManagementService.RequestedPowerState  
376 and CIM\_AssociatedPowerManagementService.PowerState properties to change. Due to the  
377 equivalence requirements stated below, it is possible that this will result in changes to the values of the  
378 CIM\_ComputerSystem.EnabledState and CIM\_ComputerSystem.RequestedState properties.

### 379 7.3.3.1 Relationship between EnabledState and PowerState

380 Table 3 and Table 4 detail the equivalency requirements for values of the  
381 CIM\_ComputerSystem.EnabledState property and the  
382 CIM\_AssociatedPowerManagementService.PowerState property for the instance of  
383 CIM\_AssociatedPowerManagementService that references the CIM\_ComputerSystem instance. When  
384 the CIM\_AssociatedPowerManagementService.PowerState property has the value listed in the first  
385 column, the CIM\_ComputerSystem.EnabledState property shall have the value listed in the second  
386 column. When the CIM\_AssociatedPowerManagementService.PowerState property has the value listed  
387 in the first column of Table 4, the CIM\_ComputerSystem.EnabledState property should have the value  
388 listed in the second column. The set of power states that can be represented by the PowerState property  
389 is a superset of those power states that are expressible through the EnabledState property. Power states  
390 expressible through the PowerState property that are not expressible through the EnabledState property  
391 are mapped to 5 (Not Applicable).

392

**Table 3 – PowerState and EnabledState Values (Required Equivalence)**

PowerState Value	Corresponding EnabledState Value
2 (On)	2 (Enabled)
8 (Off – Soft)	3 (Disabled)
12 (Off – Soft Graceful)	3 (Disabled)

393

**Table 4 – EnabledState and PowerState Values (Recommended Equivalence)**

PowerState Value	Corresponding EnabledState Value
3 (Sleep-Light)	9 (Quiesce)
4 (Sleep-Deep)	9 (Quiesce)
5 (Power Cycle (Off-Soft))	5 (Not Applicable)
6 (Off – Hard)	3 (Disabled)
7 (Hibernate (Off-Soft))	9 (Quiesce)
9 (Power Cycle (Off – Hard))	5 (Not Applicable)
10 (Master Bus Reset)	5 (Not Applicable)
11 (Diagnostic Interrupt (NMI))	5 (Not Applicable)
13 (Off – Hard Graceful)	3 (Disabled)
14 (Master Bus Reset Graceful)	5 (Not Applicable)
15 (Power Cycle (Off – Soft) Graceful)	5 (Not Applicable)
16 (Power Cycle (Off – Hard) Graceful)	5 (Not Applicable)

394

**7.3.3.2 Relationship between RequestedState and RequestedPowerState**

395

Table 5 details equivalency requirements for the values of the CIM\_ComputerSystem.RequestedState property and the CIM\_AssociatedPowerManagementService.RequestedPowerState property for the instance of CIM\_AssociatedPowerManagementService that references the CIM\_ComputerSystem instance. When the CIM\_AssociatedPowerManagementService.RequestedPowerState property has the value listed in the first column, the CIM\_ComputerSystem.RequestedState property shall have the value listed in the second column. The set of power states that can be represented by the RequestedPowerState property is a superset of those power states that are expressible through the RequestedState property. Power states expressible through the RequestedPowerState property that are not expressible through the RequestedState property are mapped to 12 (Not Applicable).

404

**Table 5 – RequestedState and RequestedPowerState Values**

RequestedPowerState Value	Corresponding RequestedState Value
2 (On)	2 (Enabled)
3 (Sleep-Light)	12 (Not Applicable)
4 (Sleep-Deep)	12 (Not Applicable)
5 (Power Cycle (Off-Soft))	11 (Reset)
6 (Power Cycle (Off-Hard))	12 (Not Applicable)
7 (Hibernate (Off-Soft))	12 (Not Applicable)
8 (Off – Hard)	12 (Not Applicable)
9 (Off – Soft)	3 (Disabled)
10 (Master Bus Reset)	12 (Not Applicable)
11 (Diagnostic Interrupt (NMI))	12 (Not Applicable)

### 405 7.3.3.3 Relationship between RequestedStatesSupported and PowerStatesSupported

406 Table 6 details equivalency requirements for values of the following properties:

- 407 • the CIM\_EnabledLogicalElementCapabilities.RequestedStatesSupported property for the  
408 instance of CIM\_EnabledLogicalElementCapabilities that is associated with the  
409 CIM\_ComputerSystem instance
- 410 • the CIM\_PowerManagementCapabilities.PowerStatesSupported property for the instance of  
411 CIM\_PowerManagementCapabilities that is associated through CIM\_ElementCapabilities with  
412 the instance of CIM\_PowerManagementService that is associated with the  
413 CIM\_ComputerSystem instance through the CIM\_AssociatedPowerManagementService  
414 association

415 When the CIM\_PowerManagementCapabilities.PowerStatesSupported property contains the value listed  
416 in the first column, the CIM\_EnabledLogicalElementCapabilities.RequestedStatesSupported property  
417 shall contain the value listed in the second column. The RequestedStatesSupported property may contain  
418 additional values that correspond to supported states. The PowerStatesSupported property may contain  
419 other values; however, corresponding values for the RequestedStatesSupported property are not defined.

420 The purpose of the PowerStatesSupported and RequestedStatesSupported properties is to indicate the  
421 power state changes that can be initiated through the RequestPowerStateChange() method and the  
422 RequestStateChange() method, respectively. The absence of a value from the array indicates the  
423 absence of support for that power state change. For those power state changes that can be initiated  
424 through the RequestPowerStateChange() method and not through the RequestStateChange() method,  
425 no mapping is defined because the absence of a value in the RequestedStatesSupported property  
426 implicitly indicates a lack of support for initiating the corresponding power state change.

427 **Table 6 – RequestedStatesSupported and PowerStatesSupported Values**

PowerStatesSupported Value	RequestedStatesSupported Value
0 (On)	2 (Enabled)
4 (Power Cycle (Off-Soft))	11 (Reset)
3 (Off – Soft)	3 (Disabled)

## 428 7.4 Text Console Redirection (Optional)

429 This section details requirements for the implementation of the [Text Console Redirection Profile](#).

### 430 7.4.1 Text Console Redirection Available to the System

431 Redirection of a text console may be supported for the system. When the redirection of a text console is  
432 supported, the requirements specified in this section shall be met.

433 The [Text Console Redirection Profile](#) shall be implemented. The Central Instance of the *Base Server*  
434 *Profile* shall be associated with the CIM\_TextRedirectionSAP instance of the [Text Console Redirection](#)  
435 [Profile](#) through the CIM\_SAPAvailableForElement association. The Central Instance of the *Base Server*  
436 *Profile* shall be associated with the Central Instance of the [Text Console Redirection Profile](#) through the  
437 CIM\_ServiceAffectsElement association.

### 438 7.4.2 Text Console Redirection Provided by the System

439 The system may provide support for the redirection of a text console for itself or other systems. When the  
440 system provides this support, the requirements specified in this section shall be met.



441 The [Text Console Redirection Profile](#) shall be implemented. The Central Instance of the *Base Server Profile*  
 442 shall be associated with the Central Instance of the [Text Console Redirection Profile](#) through the  
 443 CIM\_HostedService association. The Central Instance of the *Base Server Profile* shall be associated with  
 444 one or more instances of CIM\_TextRedirectionSAP implemented conformant with the [Text Console](#)  
 445 [Redirection Profile](#) through the CIM\_HostedAccessPoint association.

446 **8 Methods**

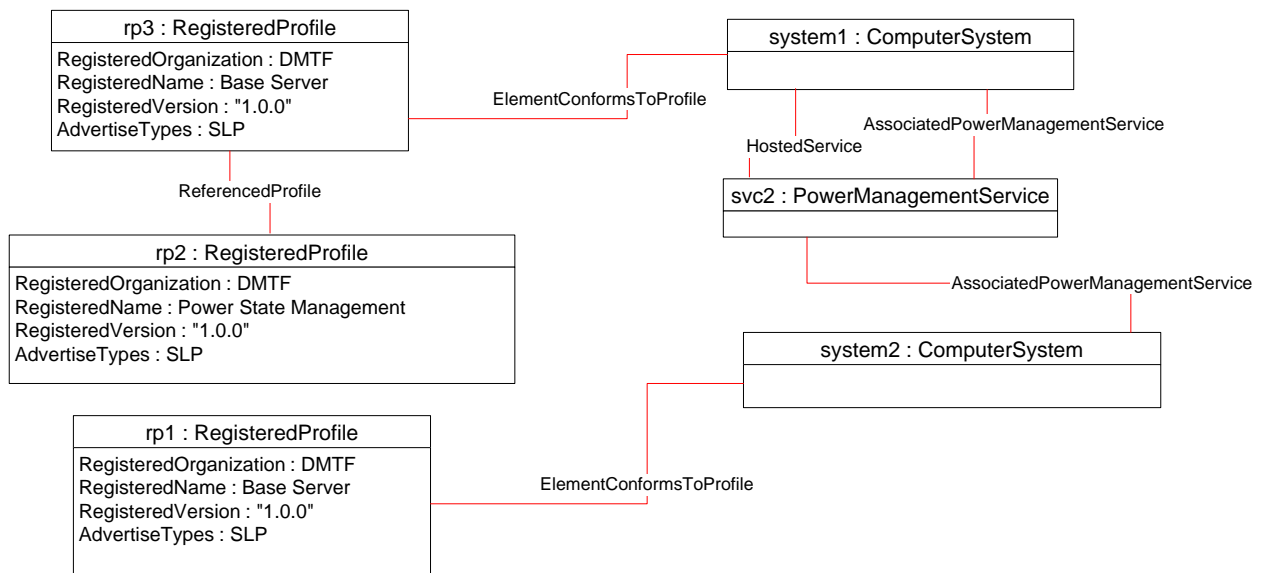
447 All intrinsic and extrinsic methods are supported as defined in the [Computer System Profile](#).

448 **9 Use Cases**

449 The following use cases are based on the implementation conforming to the DMTF *Base Server Profile*.

450 **9.1 Object Diagrams**

451 Figure 2 shows two systems conformant with the *Base Server Profile*. rp3 and rp1 both advertise the  
 452 instrumentation of the *Base Server Profile*. rp2 advertises the existence of the [Power State Management](#)  
 453 [Profile](#) and is associated with rp3, which is an instance of CIM\_RegisteredProfile that advertises the *Base*  
 454 *Server Profile*. system1 provides power control over itself and system2. The ability to provide power  
 455 control is modeled by svc2. The [Power State Management Profile](#) is advertised as supported on system1  
 456 because that is where the functionality is hosted.

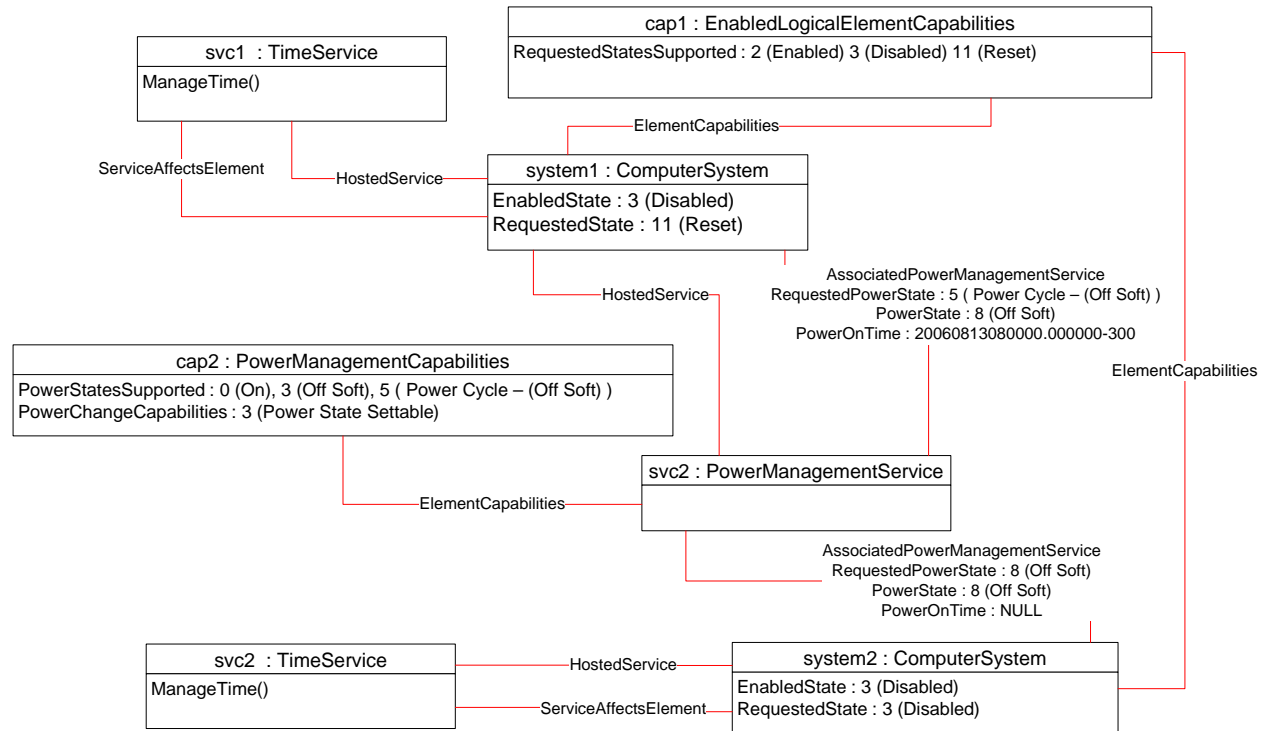


457

458

**Figure 2 – Profile Registration**

459 Figure 3 shows the power management functionality available for system1 and system2. Each system  
 460 hosts an instance of CIM\_TimeService for managing the system's time. system1 has been configured to  
 461 power on at 8 A.M. EST on August 13, 2006, as indicated by the value of the PowerOnTime property of  
 462 the instance of CIM\_AssociatedPowerManagementService that references system1. This value is relative  
 463 to the system time as returned by a call to the ManageTime() method of svc1. Note that state  
 464 management is supported with functional equivalence to the supported power state management.

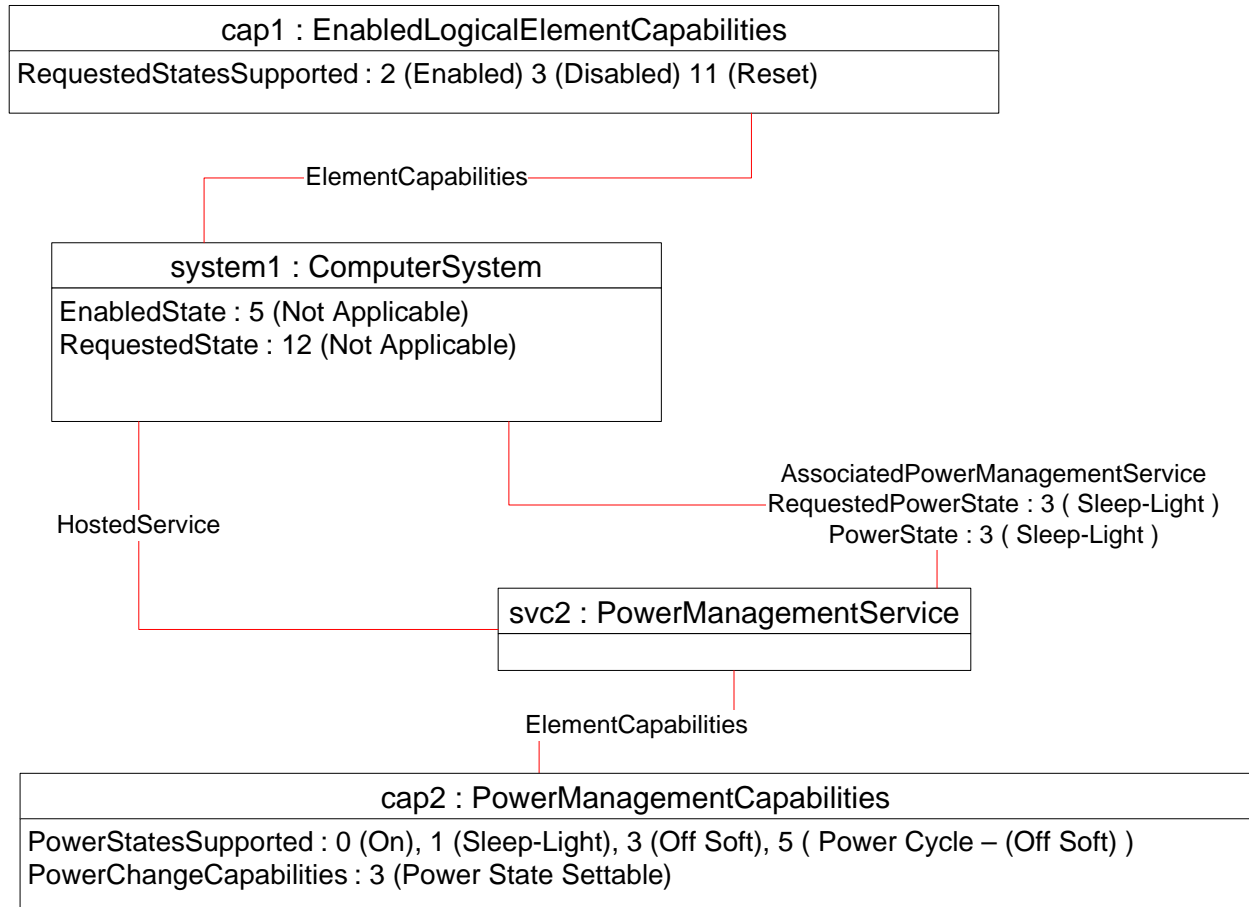


465

466

**Figure 3 – Power Management and Time Service**

467 Figure 4 shows a system where the ability to put the system into a sleep light power state is supported.  
 468 The sleep light state is an extended power state that is not expressible through the  
 469 CIM\_ComputerSystem.EnabledState property. Thus the CIM\_ComputerSystem.EnabledState property  
 470 has the value 5 (Not Applicable). The actual power state of the system is expressed through the  
 471 CIM\_AssociatedPowerManagementService.PowerState property.

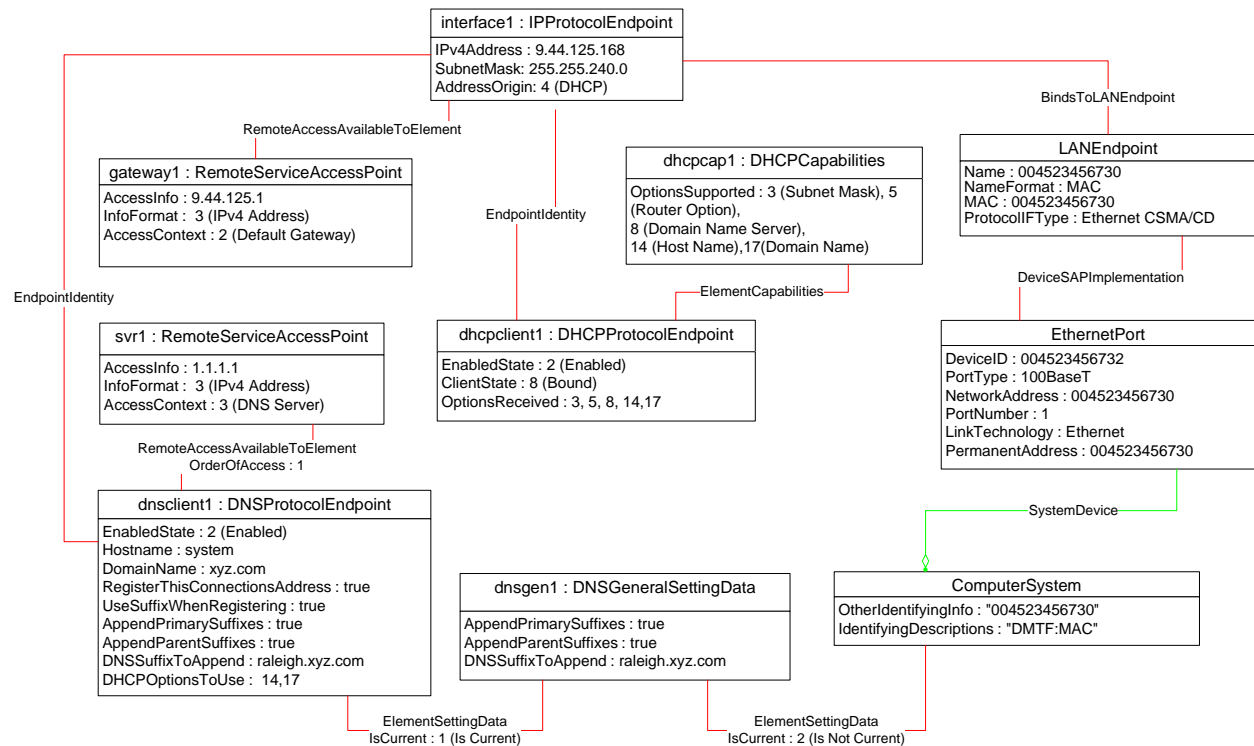


472

473

**Figure 4 – Advanced Power Management**

474 Figure 5 illustrates the network interfaces of the system. The [Ethernet Port Profile](#), [IP Interface Profile](#),  
 475 [DHCP Client Profile](#), and [DNS Client Profile](#) are implemented. The system has a single network interface.



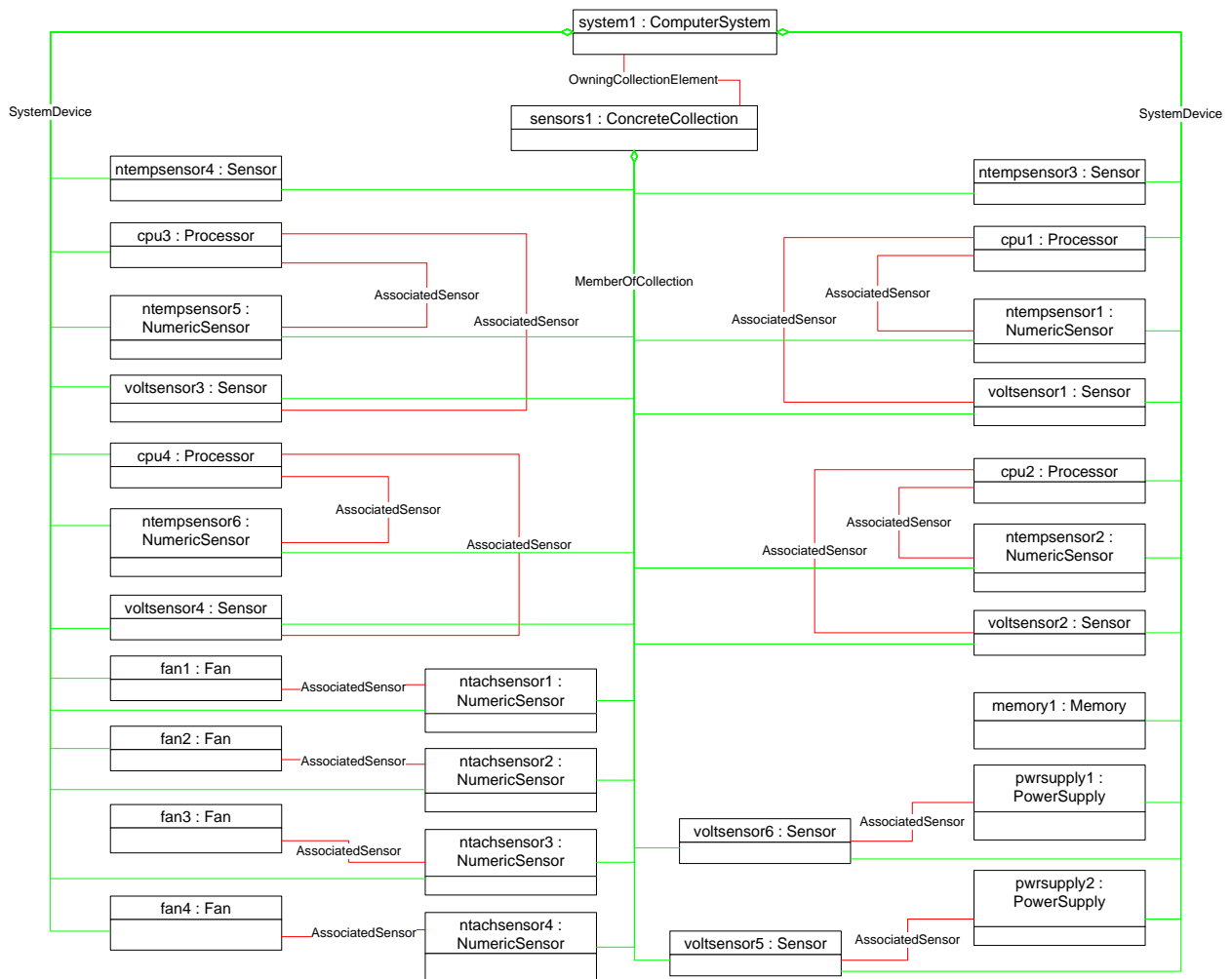
476

477

Figure 5 – Network Interfaces

478 Figure 6, Figure 7, and Figure 8 illustrate the logical and physical containment hierarchy of a single  
 479 system.

480 Figure 6 illustrates the logical hierarchy of components contained in the system. The optional [CPU Profile](#),  
 481 [Fan Profile](#), [Power Supply Profile](#), [Sensors Profile](#), [System Memory Profile](#), and [SMASH Collections](#)  
 482 [Profile](#) have been implemented. The system has four processors; each processor has a dedicated  
 483 voltage sensor and a dedicated temperature sensor. The total system memory available is modeled. The  
 484 system has two power supplies installed; each power supply has a dedicated voltage sensor. Four fans  
 485 are installed in the system; each fan has a dedicated tachometer associated with it.

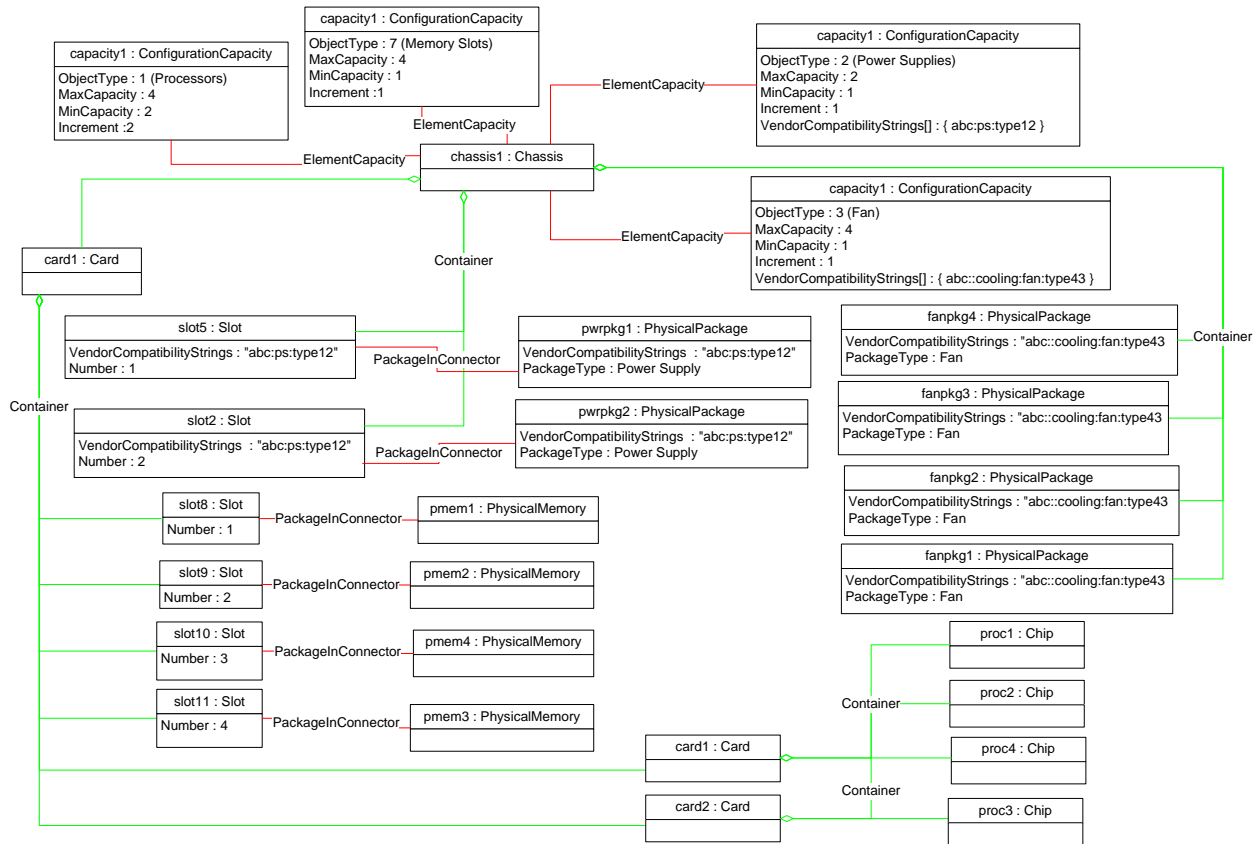


486

487

**Figure 6 – Logical Topology**

488 Figure 7 shows the physical containment hierarchy for the managed system. The [Physical Asset Profile](#)  
 489 has been implemented. The location of the fans within the system is not modeled; instead the fans are  
 490 modeled as being directly contained in the main system chassis. The slots or bays in the main chassis  
 491 that can contain a power supply are separately modeled (slot5 and slot2). The optional slot and package  
 492 compatibility behavior of the [Physical Asset Profile](#) has been implemented for the power supply slots. The  
 493 system memory is installed in four slots on the main system board (card1). The processors (proc1–proc4)  
 494 are installed in pairs on separate cards on the main system card. The capacity of the system for  
 495 processors, fans, power supplies, and memory is indicated through instances of  
 496 CIM\_ConfigurationCapacity.

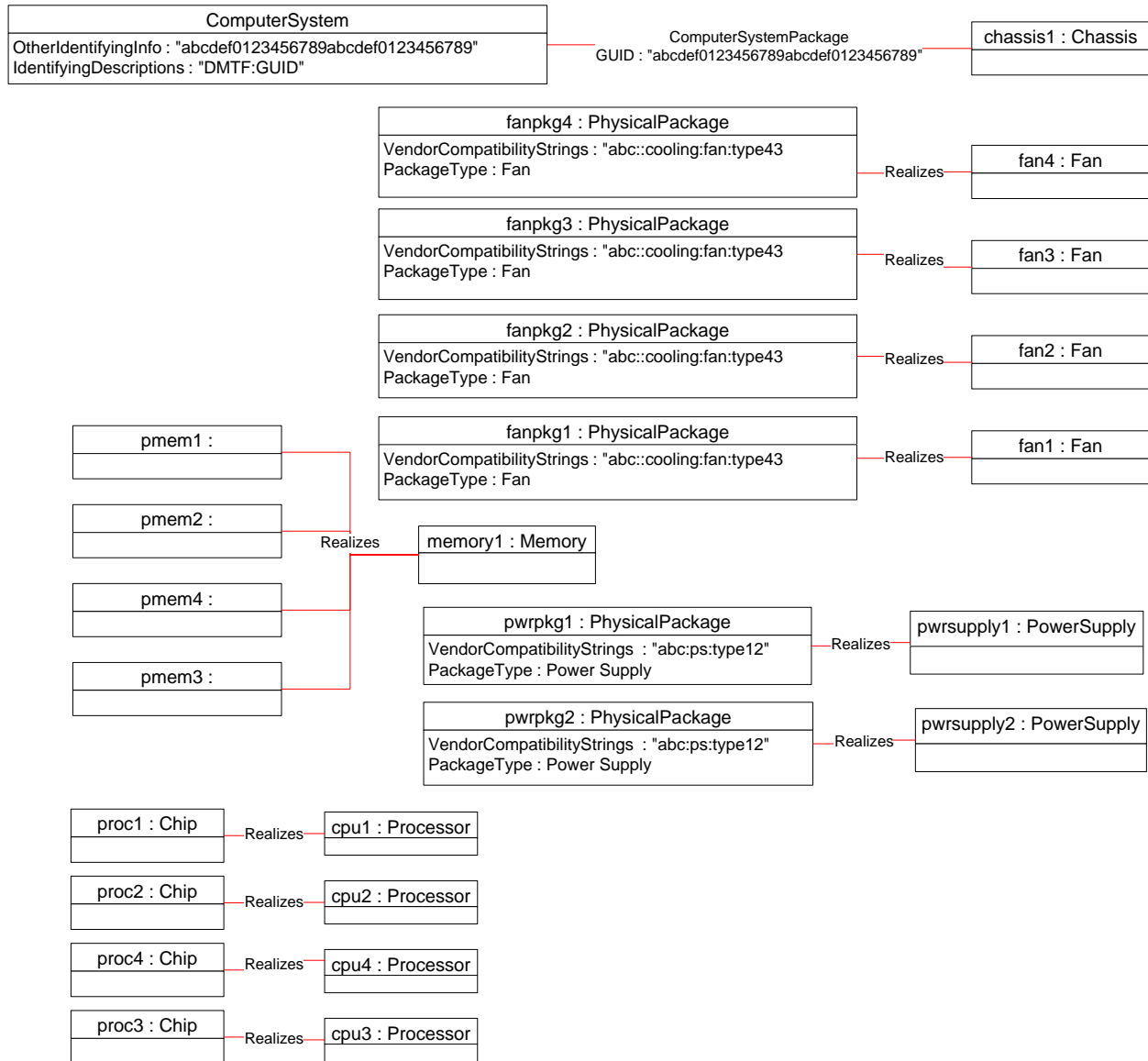


497

498

Figure 7 – Physical Topology

499 Figure 8 shows the relationship between the logical components and their underlying physical packaging.  
 500 Each fan, power supply, and processor has a dedicated package. The system memory is realized with  
 501 four physical components. The system itself is packaged in a single chassis. To keep the diagram  
 502 uncluttered, the CIM\_SystemDevice associations have been elided.



503

504 **Figure 8 – Logical to Physical Mapping**

505 **9.2 Determine the System Model and Serial Number**

506 When the optional asset management of the *Physical Asset Profile* has been implemented for the system,  
 507 a client can determine the system model and serial number as follows:

- 508 1) Find an instance of CIM\_PhysicalPackage that is associated with the Central Instance through  
 509 the CIM\_ComputerSystemPackage association.
- 510 2) Query the Model and SerialNumber properties of the instance.

### 511 9.3 Power on a System

512 A client can power on a system as follows:

- 513 1) Look for an instance of CIM\_EnabledLogicalElementCapabilities that is associated with the  
514 Central Instance through the CIM\_ElementCapabilities association.
- 515 2) Verify that the CIM\_EnabledLogicalElementCapabilities.RequestedStatesSupported property  
516 contains the value 2 (Enabled).
- 517 3) Invoke the RequestStateChange() method on the target instance, specifying 2 (Enabled) for the  
518 RequestedState parameter.

### 519 9.4 Power off a System

520 A client can power off a system as follows:

- 521 1) Look for an instance of CIM\_EnabledLogicalElementCapabilities that is associated with the  
522 Central Instance through the CIM\_ElementCapabilities association.
- 523 2) Verify that the CIM\_EnabledLogicalElementCapabilities.RequestedStatesSupported property  
524 contains the value 3 (Disabled).
- 525 3) Invoke the RequestStateChange() method on the target instance, specifying 3 (Disabled) for  
526 the RequestedState parameter.

### 527 9.5 Shutdown and Restart a System

528 A client can shut down and restart a system as follows:

- 529 1) Look for an instance of CIM\_EnabledLogicalElementCapabilities that is associated with the  
530 Central Instance through the CIM\_ElementCapabilities association.
- 531 2) Verify that the CIM\_EnabledLogicalElementCapabilities.RequestedStatesSupported property  
532 contains the value 11 (Reset).
- 533 3) Invoke the RequestStateChange() parameter on the target instance, specifying 11 (Reset) for  
534 the RequestedState parameter.

### 535 9.6 Perform System Power Control

536 A client might need to perform power control that is more granular than the functionality available through  
537 state management. This is done through power state management. A client can determine whether power  
538 state management is available for the system by searching for an instance of  
539 CIM\_PowerManagementService that is associated with the Central Instance through the  
540 CIM\_AssociatedPowerManagementService association. The specific use cases for performing power  
541 state management are documented in the [Power State Management Profile](#).

### 542 9.7 Determining the System Power State

543 A client can determine the power state of the system as follows:

- 544 1) Query the CIM\_ComputerSystem.EnabledState property.  
  
545 If the property has the value 2 (Enabled), the system is currently in ACPI state S0 (or equivalent  
546 if non-ACPI system). If the property has the value 3 (Disabled), the system is currently in ACPI  
547 state S0 (or equivalent if non-ACPI system).
- 548 2) If the CIM\_ComputerSystem.EnabledState property has the value 5 (Not Applicable), find the  
549 instance of CIM\_AssociatedPowerManagementService that references the  
550 CIM\_ComputerSystem instance.



551 3) Query the value of the CIM\_AssociatedPowerManagementService.PowerState property. The  
 552 [Power State Management Profile](#) details the equivalent ACPI states for each value.

553 **9.8 Determine the Number of Processors in the System**

554 When the optional [CPU Profile](#) is implemented, the client can determine the number of processors in the  
 555 system by querying for instances of CIM\_Processor that are associated with the Central Instance through  
 556 the CIM\_SystemDevice association.

557 The client can also use these same steps to find the fans and power supplies installed in the system,  
 558 substituting the [Fan Profile](#) and CIM\_Fan, and the [Power Supply Profile](#) and CIM\_PowerSupply  
 559 appropriately.

560 **9.9 Determine the Number of Processors that the System Can Hold**

561 When the optional configuration capacity behavior from the [Physical Asset Profile](#) is implemented for  
 562 processors for the system, a client can determine the number of processors that the system can hold as  
 563 follows:

- 564 1) Find the instances of CIM\_PhysicalPackage that are associated with the Central Instance  
 565 through the CIM\_ComputerSystemPackage association.
- 566 2) For each instance of CIM\_PhysicalPackage, find the instances of CIM\_ConfigurationCapacity  
 567 that are associated with the CIM\_PhysicalPackage instance through the CIM\_ElementCapacity  
 568 association.
- 569 3) For each instance of CIM\_ConfigurationCapacity, if the ObjectType property has the value 1  
 570 (Processors), query the MaximumCapacity property and add the value to the total number of  
 571 processors that the system can hold.

572 The client can also apply these steps to find the total amount of physical memory and the total number of  
 573 fans and power supplies that the system can hold when the configuration capacity has been instrumented  
 574 for objects of that type by substituting the appropriate value for 1 (Processors) in step 3.

575 **10 CIM Elements**

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

579 **Table 7 – CIM Elements: Base Server Profile**

Element Name	Requirement	Description
<b>Classes</b>		
CIM_ComputerSystem	Mandatory	See section 10.1.
CIM_ComputerSystemPackage	Mandatory	See section 10.2.
CIM_EnabledLogicalElementCapabilities	Optional	See section 10.3.
CIM_PhysicalPackage	Mandatory	See section 10.4.
CIM_RegisteredProfile	Mandatory	See section 10.5.
<b>Indications</b>		
None defined in this profile		

## 580 10.1 CIM\_ComputerSystem

581 An instance of CIM\_ComputerSystem is used to represent the system. Table 8 contains the requirements  
582 for elements of this class.

583 **Table 8 – Class: CIM\_ComputerSystem**

Elements	Requirement	Notes
EnabledState	Mandatory	See sections 7.3.3.1 and 7.3.1.
RequestedState	Mandatory	See section 7.3.3.2.
Dedicated	Mandatory	

## 584 10.2 CIM\_ComputerSystemPackage

585 One or more instances of CIM\_ComputerSystemPackage associate the CIM\_ComputerSystem instance  
586 with the CIM\_PhysicalPackage instances in which it resides. The constraints specified in Table 9 are in  
587 addition to those specified in the [Physical Asset Profile](#).

588 **Table 9 – Class: CIM\_ComputerSystemPackage**

Elements	Requirement	Notes
Dependent	Mandatory	This property shall be a reference to the Central Instance. Cardinality 1
Antecedent	Mandatory	This property shall be a reference to CIM_PhysicalPackage. Cardinality 1..*

## 589 10.3 CIM\_EnabledLogicalElementCapabilities

590 CIM\_EnabledLogicalElementCapabilities indicates support for managing the state of the system.  
591 Table 10 contains the requirements for elements of this class.

592 **Table 10 – Class: CIM\_EnabledLogicalElementCapabilities**

Elements	Requirement	Notes
RequestedStatesSupported	Mandatory	See section 7.3.3.3.

## 593 10.4 CIM\_PhysicalPackage

594 One or more instances of CIM\_PhysicalPackage represent the physical packaging of the computer  
595 system. Other than the existence of at least one instance of CIM\_PhysicalPackage, this profile does not  
596 specify any constraints for CIM\_PhysicalPackage beyond those specified in the [Physical Asset Profile](#).

597 **10.5 CIM\_RegisteredProfile**

598 CIM\_RegisteredProfile identifies the *Base Server Profile* in order for a client to determine whether an  
 599 instance of CIM\_ComputerSystem is conformant with this profile. The CIM\_RegisteredProfile class is  
 600 defined by the [Profile Registration Profile](#). With the exception of the mandatory values specified for the  
 601 properties in Table 11, the behavior of the CIM\_RegisteredProfile instance is in accordance with the  
 602 [Profile Registration Profile](#).

603 **Table 11 – Class: CIM\_RegisteredProfile**

Elements	Requirement	Notes
RegisteredName	Mandatory	This property shall have a value of "Base Server".
RegisteredVersion	Mandatory	This property shall have a value of "1.0.0".
RegisteredOrganization	Mandatory	This property shall have a value of 2 (DMTF).

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

607

608  
609  
610  
611

**ANNEX A**  
(informative)  
**Change Log**

Version	Date	Description
1.0.0	2009-06-16	DMTF Standard Release

612