



1

2

3

4

Document Number: DSP1004

Date: 2006-09-06

Version: 1.0.0a

5 **Base Server Profile**

6 **Document Type: Specification**

7 **Document Status: Preliminary Standard**

8 **Document Language: E**

Base Server Profile

9 Copyright Notice

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

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

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26 withdrawn or modified after publication, and shall be indemnified and held harmless by any party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

CONTENTS

30	Foreword	5
31	Introduction	6
32	1 Scope	7
33	2 Normative References.....	7
34	2.1 Approved References	7
35	2.2 References under Development	7
36	2.3 Other References.....	8
37	3 Terms and Definitions	8
38	4 Symbols and Abbreviated Terms	9
39	5 Synopsis.....	9
40	6 Description	10
41	6.1 Representation of System Power State.....	11
42	7 Implementation.....	12
43	7.1 Base Server System	12
44	7.2 Management of Base Server Components	13
45	7.3 State Management.....	13
46	7.4 Text Console Redirection (Optional).....	16
47	8 Methods.....	16
48	9 Use Cases	17
49	9.1 Object Diagrams	17
50	9.2 Determine the System Model and Serial Number	23
51	9.3 Power On a System	24
52	9.4 Power Off a System	24
53	9.5 Shutdown and Restart a System	24
54	9.6 Perform System Power Control	24
55	9.7 Determining the System Power State.....	24
56	9.8 Determine the Number of Processors in the System	25
57	9.9 Determine the Number of Processors that the System Can Hold	25
58	10 CIM Elements	25
59	10.1 CIM_ComputerSystem.....	26
60	10.2 CIM_ComputerSystemPackage	26
61	10.3 CIM_EnabledLogicalElementCapabilities.....	26
62	10.4 CIM_PhysicalPackage	26
63	10.5 CIM_RegisteredProfile.....	27
64	ANNEX A (Informative) Change Log	28
65	ANNEX B (informative) Acknowledgments.....	29
66		

67 Figures

68	Figure 1 – Base Server Profile: Class Diagram	11
69	Figure 2 – Profile Registration.....	17
70	Figure 3 – Power Management and Time Service	18
71	Figure 4 – Advanced Power Management.....	19
72	Figure 5 – Network Interfaces	20
73	Figure 6 – Logical Topology.....	21
74	Figure 7 – Physical Topology.....	22
75	Figure 8 – Logical to Physical Mapping	23
76		

Base Server Profile

77 Tables

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

88

89

Foreword

90 The *Base Server Profile* (DSP1004) was prepared by the Server Management Working Group of the
91 DMTF.

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

94

Introduction

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

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

101

Base Server Profile

102 1 Scope

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

108 2 Normative References

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

112 2.1 Approved References

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

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

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

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

117 *Advanced Configuration and Power Interface Specification* (ACPI Specification), revision 3.0,
118 www.acpi.info/Downloads/ACPIspec30.pdf

119 2.2 References under Development

120 DMTF DSP1012, *Boot Control Profile*

121 DMTF DSP1005, *CLP Service Profile*

122 DMTF DSP1052, *Computer System Profile*

123 DMTF [DSP1022](#), *CPU Profile*

124 DMTF [DSP1037](#), *DHCP Client Profile*

125 DMTF [DSP1038](#), *DNS Client Profile*

126 DMTF [DSP1014](#), *Ethernet Port Profile*

127 DMTF [DSP1013](#), *Fan Profile*

128 DMTF DSP1023, *Firmware Inventory Profile*

129 DMTF DSP1025, *Firmware Update Profile*

130 DMTF [DSP1036](#), *IP Interface Profile*

131 DMTF [DSP1011](#), *Physical Asset Profile*

132 DMTF [DSP1027](#), *Power State Management Profile*

Base Server Profile

- 133 DMTF [DSP1015](#), *Power Supply Profile*
- 134 DMTF [DSP1033](#), *Profile Registration Profile*
- 135 DMTF DSP1010, *Record Log Profile*
- 136 DMTF [DSP1009](#), *Sensors Profile*
- 137 DMTF [DSP1006](#), *SMASH Collections Profile*
- 138 DMTF [DSP1017](#), *SSH Service Profile*
- 139 DMTF DSP1026, *System Memory Profile*
- 140 DMTF [DSP1016](#), *Telnet Service Profile*
- 141 DMTF DSP1024, *Text Console Redirection Profile*

142 2.3 Other References

- 143 ISO/IEC Directives, Part 2, [Rules for the structure and drafting of International Standards](#)
- 144 OMG, [Unified Modeling Language \(UML\) from the Open Management Group \(OMG\)](#)

145 3 Terms and Definitions

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

148 3.1

149 **can**

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

151 3.2

152 **cannot**

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

154 3.3

155 **conditional**

156 indicates requirements to be followed strictly to conform to the document when the specified conditions
157 are met

158 3.4

159 **mandatory**

160 indicates requirements to be followed strictly to conform to the document and from which no deviation is
161 permitted

162 3.5

163 **may**

164 indicates a course of action permissible within the limits of the document

165 3.6

166 **need not**

167 indicates a course of action permissible within the limits of the document

- 168 **3.7**
 169 **optional**
 170 indicates a course of action permissible within the limits of the document
- 171 **3.8**
 172 **referencing profile**
 173 indicates a profile that owns the definition of this class and can include a reference to this profile in its
 174 “Referenced Profiles” table
- 175 **3.9**
 176 **shall**
 177 indicates requirements to be followed strictly to conform to the document and from which no deviation is
 178 permitted
- 179 **3.10**
 180 **shall not**
 181 indicates requirements to be followed strictly to conform to the document and from which no deviation is
 182 permitted
- 183 **3.11**
 184 **should**
 185 indicates that among several possibilities, one is recommended as particularly suitable, without
 186 mentioning or excluding others, or that a certain course of action is preferred but not necessarily required
- 187 **3.12**
 188 **should not**
 189 indicates that a certain possibility or course of action is deprecated but not prohibited
- 190 **3.13**
 191 **unspecified**
 192 indicates that this profile does not define any constraints for the referenced CIM element or operation

193 **4 Symbols and Abbreviated Terms**

194 The following abbreviations are used in this document.

- 195 **4.1**
 196 **ACPI**
 197 Advanced Configuration and Power Interface

198 **5 Synopsis**

- 199 **Profile Name:** *Base Server Profile*
 200 **Version:** 1.0.0a
 201 **Organization:** DMTF
 202 **CIM schema version:** 2.12
 203 **Specializes:** *DMTF Computer System Profile 1.0.0*
 204 **Central Class:** CIM_ComputerSystem
 205 **Scoping Class:** CIM_ComputerSystem

Base Server Profile

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

208 The Central Class of the *Base Server Profile* shall be *CIM_ComputerSystem*. The Central Instance shall
209 be an instance of *CIM_ComputerSystem*. The Scoping Class shall be *CIM_ComputerSystem*. The
210 Scoping Instance shall be the Central Instance. Table 1 lists profiles upon which this profile has a
211 dependency.

212 **Table 1 – Referenced Profiles**

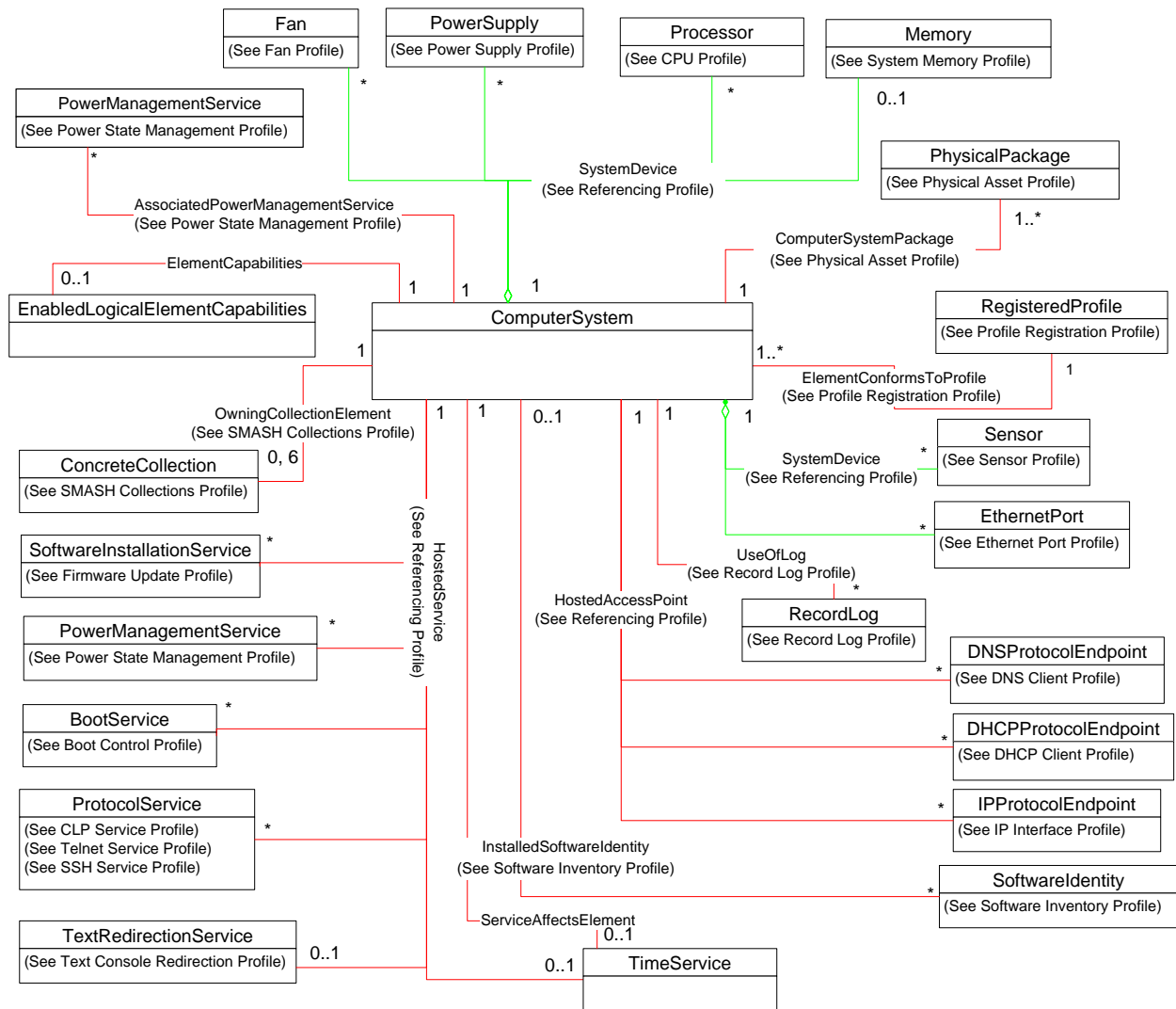
Profile Name	Organization	Version	Relationship	Behavior
<i>Computer System Profile</i>	DMTF	1.0	Specializes	None
<i>Fan Profile</i>	DMTF	1.0	Optional	See section 7.2.1.
<i>Physical Asset Profile</i>	DMTF	1.0	Mandatory	See section 7.1.2.
<i>Power State Management Profile</i>	DMTF	1.0	Optional	See section 7.3.3.
<i>Power Supply Profile</i>	DMTF	1.0	Optional	See section 7.2.2.
<i>Profile Registration Profile</i>	DMTF	1.0	Mandatory	None
<i>Text Console Redirection Profile</i>	DMTF	1.0	Optional	See section 7.4.

213 6 Description

214 The *Base Server Profile* is an autonomous profile that defines the minimum top-level object model
215 needed to model simple server hardware and related software. Other profiles add additional management
216 objects to this basic server model to provide system configuration, boot control, and other provisioning
217 capabilities. *CIM_ComputerSystem* represents the server system. *CIM_TimeService* provides the ability
218 to manage the system time.

219 Figure 1 presents the class schema for the *Base Server Profile*. For simplicity, the prefix *CIM_* has been
220 removed from the names of the classes.

221 Note that the behavioral constraints for many of the profiles identified in Figure 1 are inherited from the
222 specialized *Computer System Profile*. Therefore, although they are shown in Figure 1, they are not
223 referenced in this specification. Examples are the *IP Interface Profile*, *Ethernet Port Profile*, and *Record*
224 *Log Profile*.



225

226

Figure 1 – Base Server Profile: Class Diagram

227 **6.1 Representation of System Power State**

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

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

236 The power management behavior and system power states specified in the *Power State Management*
 237 *Profile* are a superset of the function and states that are represented using the EnabledState and
 238 RequestedState properties of CIM_ComputerSystem. That is, the EnabledState and RequestedState
 239 properties are sufficient to represent standard ACPI states S0 and S5. Implementing the *Power State Management*
 240 *Profile* provides the ability to represent additional ACPI states. For example, the equivalency between the
 241 EnabledState and PowerState values results from their mapping to identical ACPI states rather than that

Base Server Profile

242 they are defined in terms of each other. For the subset of values for the EnabledState and
243 RequestedState properties for which ACPI states are defined, there is a one-to-one correspondence with
244 a legal value for the PowerState and RequestedPowerState properties.

245 Defining the states expressible through the *Power State Management Profile* as a superset of those
246 states possible with the EnabledState and RequestedState properties is contrasted with the discarded
247 alternative of using the implementation of the *Power State Management Profile* to provide a refinement of
248 the interpretation of the EnabledState and RequestedState values. If this latter, discarded approach were
249 taken, multiple values of PowerState and RequestedPowerState would be mapped to the less granular
250 values for the EnabledState and RequestedState properties.

251 7 Implementation

252 The *Base Server Profile* consists of definitions for the CIM_ComputerSystem, CIM_PhysicalPackage, and
253 CIM_TimeService classes, and their related EnabledLogicalElementCapabilities classes. Other related
254 subsystem classes such as CIM_LogicalDevice, CIM_Collection, and CIM_RecordLog are defined in their
255 respective profiles.

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

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

262 7.1 Base Server System

263 The instrumentation shall create an instance of CIM_ComputerSystem to represent the system being
264 modeled.

265 7.1.1 Identifying a Base Server

266 This section details the constraints beyond those specified in the *Computer System Profile* for using the
267 IdentifyingDescriptions and OtherIdentifyingInfo properties to identify a computer system.

268 7.1.1.1 CIM:GUID

269 The value of the OtherIdentifyingInfo property shall match the value of the
270 CIM_ComputerSystemPackage.PlatformGUID property for an instance of CIM_ComputerSystemPackage
271 that references the Central Instance.

272 7.1.1.2 CIM:Model:SerialNumber

273 The value of the OtherIdentifyingInfo property shall match the value of the Model property of an instance
274 of CIM_PhysicalPackage, concatenated with a single colon (:), concatenated with the value of the
275 SerialNumber property of the same instance of CIM_PhysicalPackage.

276 7.1.1.3 CIM:Tag

277 The value of the OtherIdentifyingInfo property shall match the value of the Tag property of an instance of
278 CIM_PhysicalPackage.

279 7.1.2 Representing the Physical Packaging

280 The physical packaging for a system shall be modeled according to the requirements specified in the
 281 *Physical Asset Profile*. At least one instance of CIM_PhysicalPackage shall be associated with the
 282 Central Instance through the CIM_ComputerSystemPackage association.

283 7.2 Management of Base Server Components

284 The following subclauses detail the requirements for management of components of the system in
 285 addition to those specified in the *Computer System Profile*.

286 7.2.1 Instrumentation of Fans (Optional)

287 A system can contain one or more fans that provide cooling for the system. When the fans of the system
 288 are instrumented, the instrumentation shall be conformant with the *Fan Profile*, and the Central Instance
 289 of the *Base Server Profile* shall be associated with the Central Instance of the *Fan Profile* through the
 290 CIM_SystemDevice association.

291 7.2.2 Instrumentation of Power Supplies (Optional)

292 A system can contain one or more power supplies that provide power to the system. When the power
 293 supplies of the system are instrumented, the instrumentation shall be conformant with the *Power Supply*
 294 *Profile*, and the Central Instance of the *Base Server Profile* shall be associated with the Central Instance
 295 of the *Power Supply Profile* through the CIM_SystemDevice association.

296 7.3 State Management

297 This section details further constraints related to state management beyond those specified in the
 298 *Computer System Profile*.

299 7.3.1 State Management Supported

300 When the optional state management behavior specified in the *Computer System Profile* is supported, the
 301 following requirements shall be met.

302 7.3.1.1 CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported

303 The RequestedStatesSupported property shall contain one or more of the following values: 2 (Enabled), 3
 304 (Disabled), or 11 (Reset).

305 7.3.1.2 CIM_ComputerSystem.EnabledState

306 The EnabledState property shall have the value 2 (Enabled) or 3 (Disabled).

307 7.3.2 Correspondence of System States and ACPI States

308 The EnabledState property of CIM_ComputerSystem is defined in terms of ACPI values to provide
 309 meaningful context for the interpretation of values for a computer system realized in hardware. The
 310 mappings specified in Table 2 shall be used. It is not necessary for the underlying modeled system to
 311 support the ACPI specification.

312 **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

Base Server Profile

313 7.3.3 Power State Management (Optional)

314 The *Power State Management Profile* may be supported because the Central Instance either hosts an
315 instance of *CIM_PowerManagementService* or has the functionality of one available to it.

316 7.3.3.1 Power Management Available to System

317 Management of the power state of the system may be supported for the system. When the management
318 of the power state is supported, the *Power State Management Profile* shall be implemented and the
319 Central Instance of the *Base Server Profile* shall be associated with the Central Instance of the *Power
320 State Management Profile* through the *CIM_AssociatedPowerManagementService* association.

321 7.3.3.2 Power Management Hosted on System

322 The system may provide the ability to manage the power state of itself or other systems. When the
323 system provides this ability, the *Power State Management Profile* shall be implemented and the Central
324 Instance of the *Base Server Profile* shall be associated with the Central Instance of the *Power State
325 Management Profile* through the *CIM_HostedService* association.

326 7.3.4 Relationship between State Management and Power State Management 327 (Conditional)

328 The behavior in this section is conditional on the implementation of the behavior in section 7.3.3.1. When
329 the optional behavior specified in section 7.3.3.1 is supported, the state management behavior specified
330 in section "State Management Is Supported (Conditional)" of the *Computer System Profile* shall be
331 supported.

332 Management of the power state may be supported for a system. One reason for supporting power state
333 management is the need to provide more granular management beyond that available through state
334 management. To ensure consistent semantics for state management regardless of whether power state
335 management is supported, it is necessary to establish constraints on the interaction of power state
336 management and state management when power state management is supported. This section details
337 these constraints.

338 Note that the *CIM_ComputerSystem.RequestStateChange()* method defined in the *Computer System
339 Profile* causes the values for the *CIM_ComputerSystem.EnabledState* and
340 *CIM_ComputerSystem.RequestedState* properties to change. Due to the equivalence requirements
341 stated below, the possible invocation of the method will result in changes to the values of the
342 *CIM_AssociatedPowerManagementService.RequestedPowerState* and
343 *CIM_AssociatedPowerManagementService.PowerState* properties. Likewise, the
344 *CIM_PowerManagementService.RequestPowerStateChange()* method defined in the *Power State
345 Management Profile* will cause the *CIM_AssociatedPowerManagementService.RequestedPowerState*
346 and *CIM_AssociatedPowerManagementService.PowerState* properties to change. Due to the
347 equivalence requirements stated below, it is possible that this will result in changes to the values of the
348 *CIM_ComputerSystem.EnabledState* and *CIM_ComputerSystem.RequestedState* properties.

349 7.3.4.1 Relationship between EnabledState and PowerState

350 Table 3 details the equivalency requirements for values of the *CIM_ComputerSystem.EnabledState*
351 property and the *CIM_AssociatedPowerManagementService.PowerState* property for the instance of
352 *CIM_AssociatedPowerManagementService* that references the *CIM_ComputerSystem* instance. When
353 the *CIM_AssociatedPowerManagementService.PowerState* property has the value listed in the first
354 column, the *CIM_ComputerSystem.EnabledState* property shall have the value listed in the second
355 column. The set of power states that can be represented by the *PowerState* property is a superset of
356 those power states that are expressible through the *EnabledState* property. Power states expressible
357 through the *PowerState* property that are not expressible through the *EnabledState* property are mapped
358 to 5 (Not Applicable).

359

Table 3 – EnabledState and PowerState Values

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

360 7.3.4.2 Relationship between RequestedState and RequestedPowerState

361 Table 4 details equivalency requirements for the values of the CIM_ComputerSystem.RequestedState
 362 property and the CIM_AssociatedPowerManagementService.RequestedPowerState property for the
 363 instance of CIM_AssociatedPowerManagementService that references the CIM_ComputerSystem
 364 instance. When the CIM_AssociatedPowerManagementService.RequestedPowerState property has the
 365 value listed in the first column, the CIM_ComputerSystem.RequestedState property shall have the value
 366 listed in the second column. The set of power states that can be represented by the
 367 RequestedPowerState property is a superset of those power states that are expressible through the
 368 RequestedState property. Power states expressible through the RequestedPowerState property that are
 369 not expressible through the RequestedState property are mapped to 12 (Not Applicable).

370

Table 4 – 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)

371 7.3.4.3 Relationship between RequestedStatesSupported and PowerStatesSupported

372 Table 5 details equivalency requirements for values of the following properties:

- 373 • the CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported property for the
 374 instance of CIM_EnabledLogicalElementCapabilities that is associated with the
 375 CIM_ComputerSystem instance

Base Server Profile

- the CIM_PowerManagementCapabilities.PowerStatesSupported property for the instance of CIM_PowerManagementCapabilities that is associated through CIM_ElementCapabilities with the instance of CIM_PowerManagementService that is associated with the CIM_ComputerSystem instance through the CIM_AssociatedPowerManagementService association

When the CIM_PowerManagementCapabilities.PowerStatesSupported property contains the value listed in the first column, the CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported property shall contain the value listed in the second column. The RequestedStatesSupported property may contain additional values that correspond to supported states. The PowerStatesSupported property may contain other values; however, corresponding values for the RequestedStatesSupported property are not defined.

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

Table 5 – RequestedStatesSupported and PowerStatesSupported Values

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

7.4 Text Console Redirection (Optional)

This section details requirements for the implementation of the *Text Console Redirection Profile*.

7.4.1 Text Console Redirection Available to the System

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

The *Text Console Redirection Profile* shall be implemented. The Central Instance of the *Base Server Profile* shall be associated with the CIM_TextRedirectionSAP instance of the *Text Console Redirection Profile* through the CIM_SAPAvailableForElement association. The Central Instance of the *Base Server Profile* shall be associated with the Central Instance of the *Text Console Redirection Profile* through the CIM_ServiceAffectsElement association.

7.4.2 Text Console Redirection Provided by the System

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

The *Text Console Redirection Profile* shall be implemented. The Central Instance of the *Base Server Profile* shall be associated with the Central Instance of the *Text Console Redirection Profile* through the CIM_HostedService association. The Central Instance of the *Base Server Profile* shall be associated with one or more instances of CIM_TextRedirectionSAP implemented conformant with the *Text Console Redirection Profile* through the CIM_HostedAccessPoint association.

8 Methods

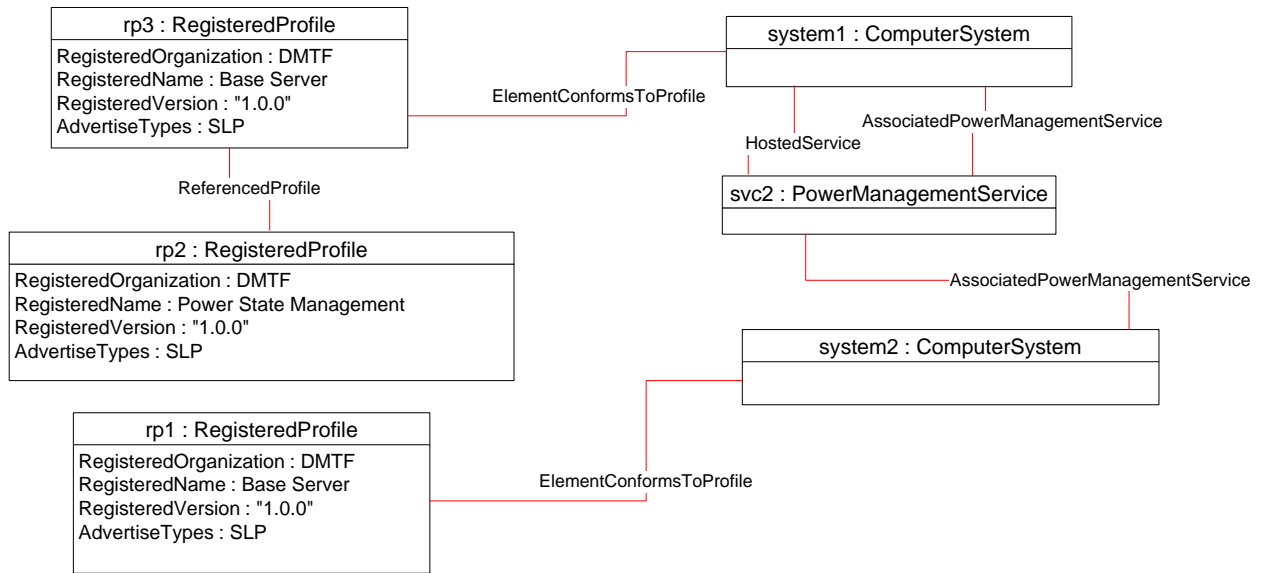
All intrinsic and extrinsic methods are supported as defined in the *Computer System Profile*.

413 **9 Use Cases**

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

415 **9.1 Object Diagrams**

416 Figure 2 shows two systems conformant with the *Base Server Profile*. rp3 and rp1 both advertise the
 417 instrumentation of the *Base Server Profile*. rp2 advertises the existence of the *Power State Management*
 418 *Profile* and is associated with rp3, which is an instance of CIM_RegisteredProfile that advertises the *Base*
 419 *Server Profile*. system1 provides power control over itself and system2. The ability to provide power
 420 control is modeled by svc2. The *Power State Management Profile* is advertised as supported on system1
 421 because that is where the functionality is hosted.



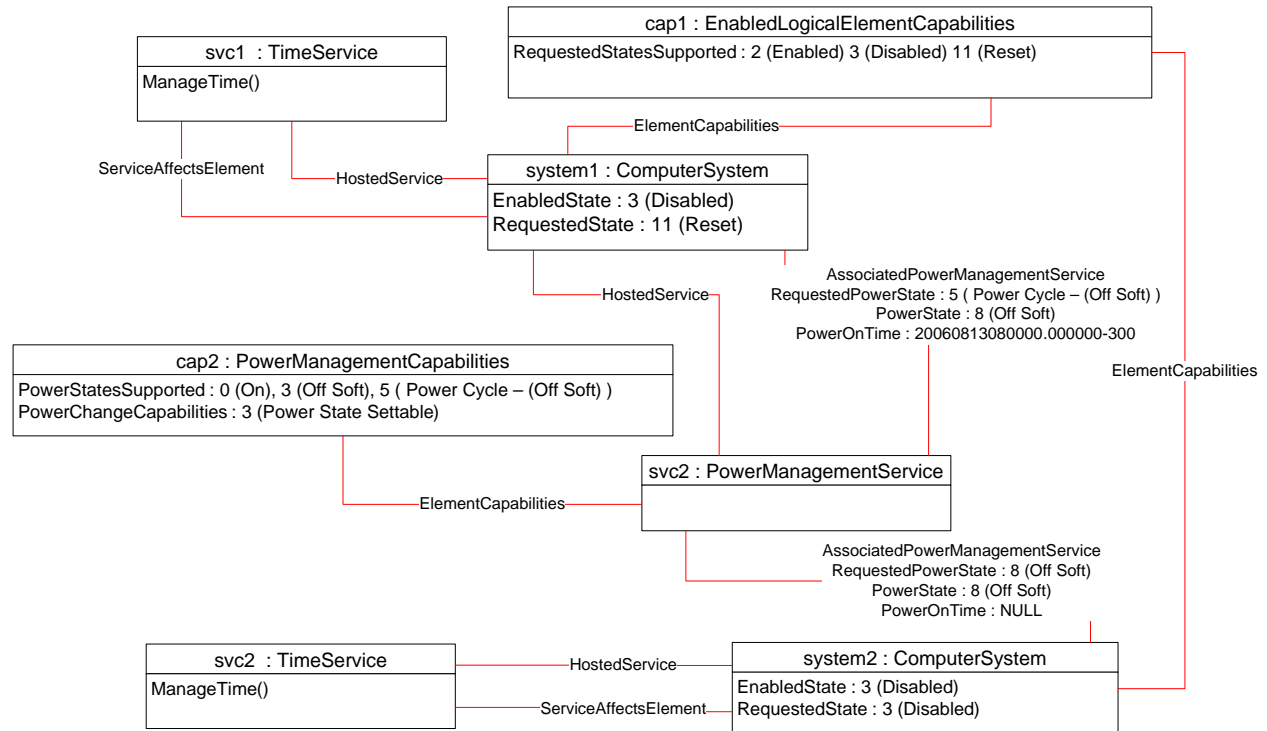
422

423

Figure 2 – Profile Registration

Base Server Profile

424 Figure 3 shows the power management functionality available for system1 and system2. Each system
 425 hosts an instance of CIM_TimeService for managing the system's time. system1 has been configured to
 426 power on at 8 A.M. EST on August 13, 2006, as indicated by the value of the PowerOnTime property of
 427 the instance of CIM_AssociatedPowerManagementService that references system1. This value is relative
 428 to the system time as returned by a call to the ManageTime() method of svc1. Note that state
 429 management is supported with functional equivalence to the supported power state management.

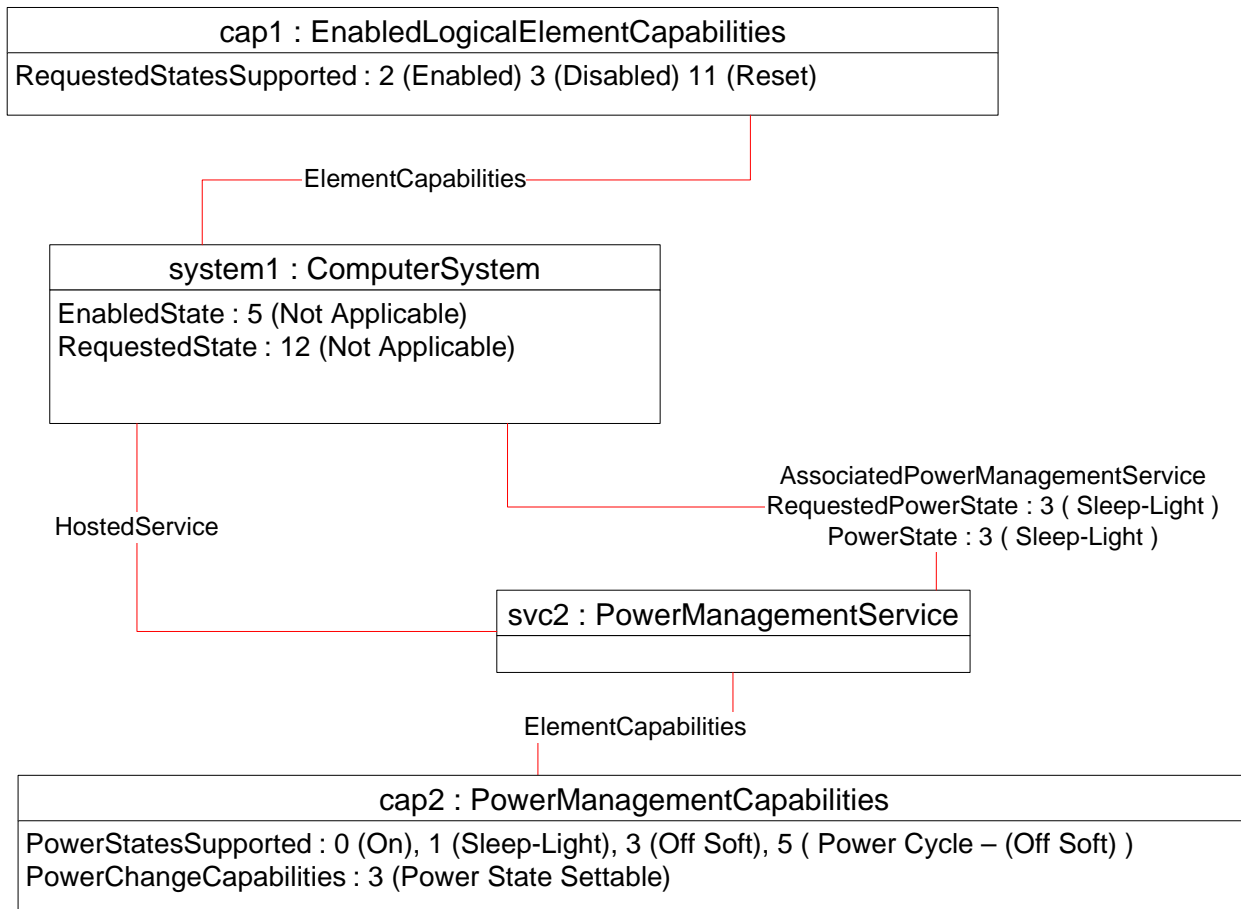


430

431

Figure 3 – Power Management and Time Service

432 Figure 4 shows a system where the ability to put the system into a sleep light power state is supported.
 433 The sleep light state is an extended power state that is not expressible through the
 434 CIM_ComputerSystem.EnabledState property. Thus the CIM_ComputerSystem.EnabledState property
 435 has the value 5 (Not Applicable). The actual power state of the system is expressed through the
 436 CIM_AssociatedPowerManagementService.PowerState property.



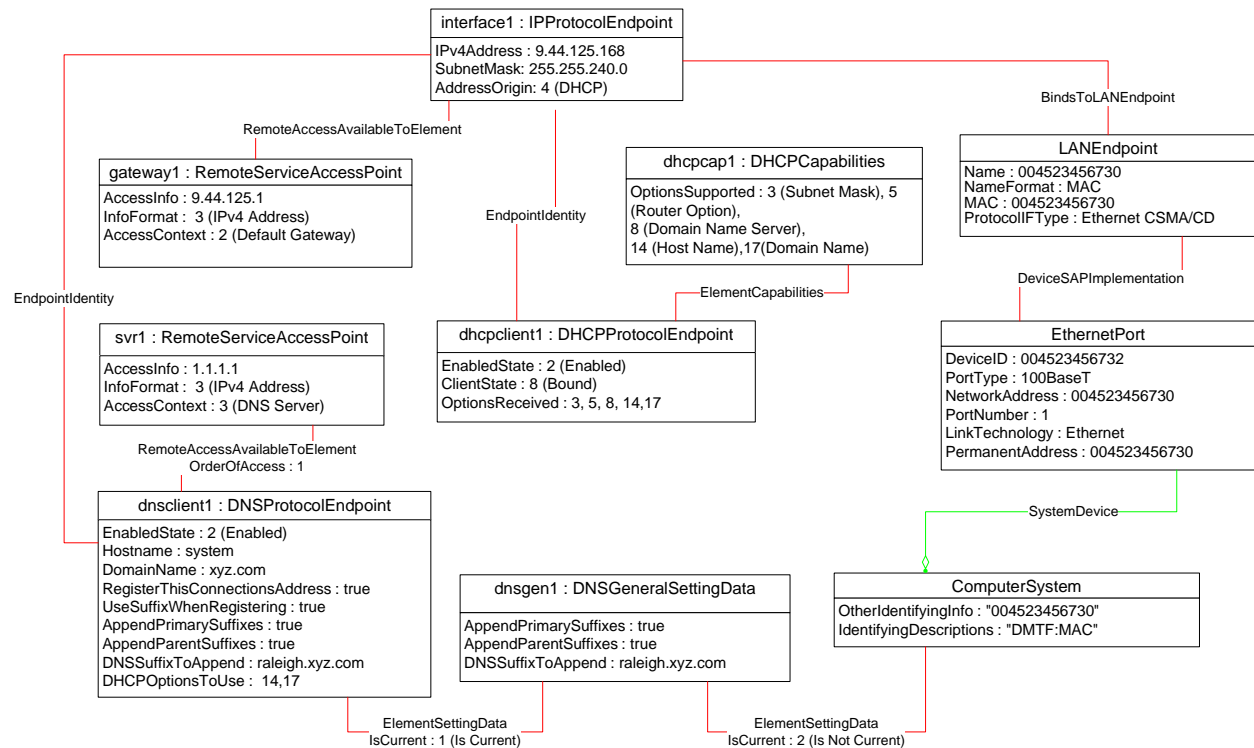
437

438

Figure 4 – Advanced Power Management

Base Server Profile

439 Figure 5 illustrates the network interfaces of the system. The *Ethernet Port Profile*, *IP Interface Profile*,
 440 *DHCP Client Profile*, and *DNS Client Profile* are implemented. The system has a single network interface.



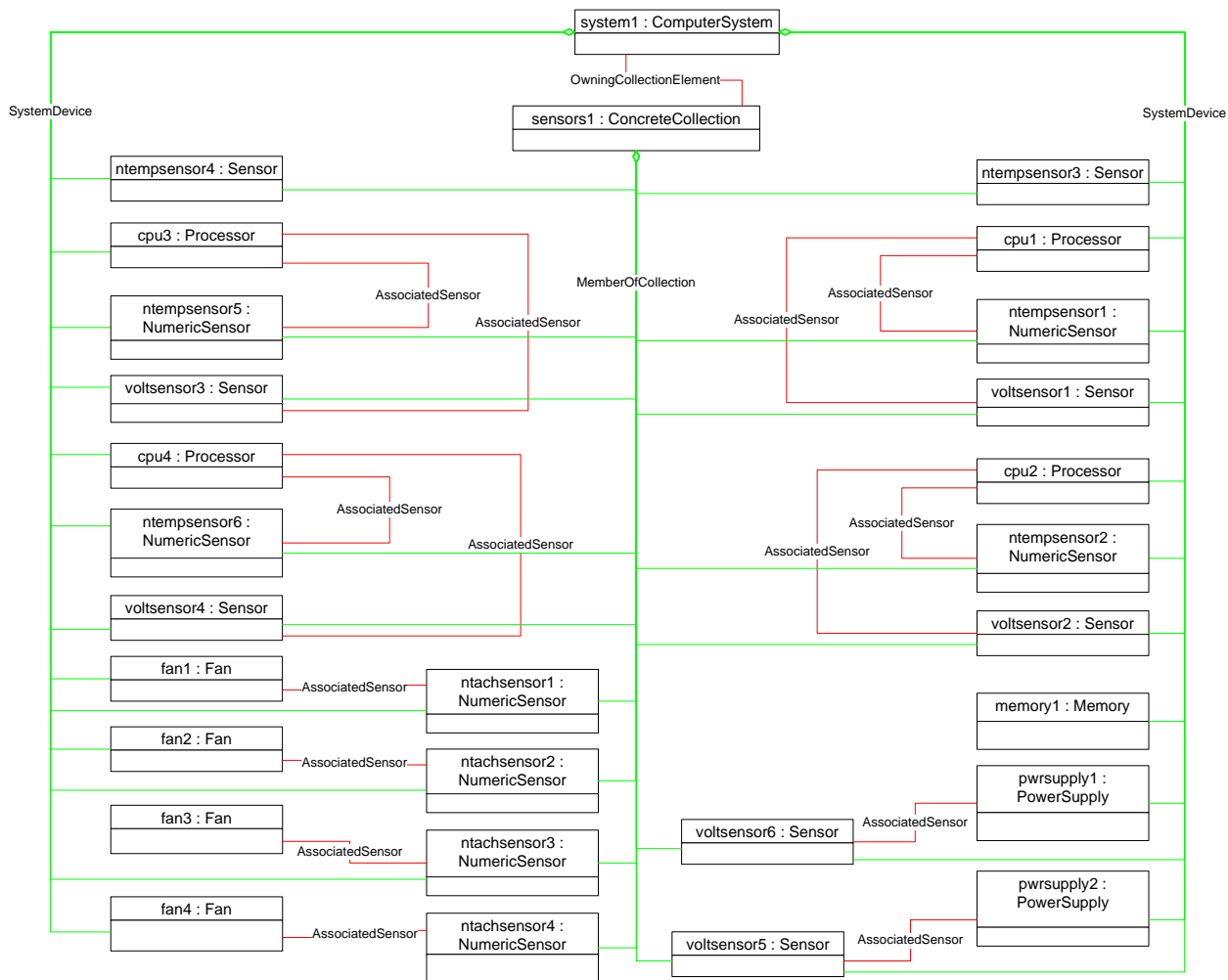
441

442

Figure 5 – Network Interfaces

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

445 Figure 6 illustrates the logical hierarchy of components contained in the system. The optional *CPU Profile*,
 446 *Fan Profile*, *Power Supply Profile*, *Sensors Profile*, *System Memory Profile*, and *SMASH Collections*
 447 *Profile* have been implemented. The system has four processors; each processor has a dedicated
 448 voltage sensor and a dedicated temperature sensor. The total system memory available is modeled. The
 449 system has two power supplies installed; each power supply has a dedicated voltage sensor. Four fans
 450 are installed in the system; each fan has a dedicated tachometer associated with it.



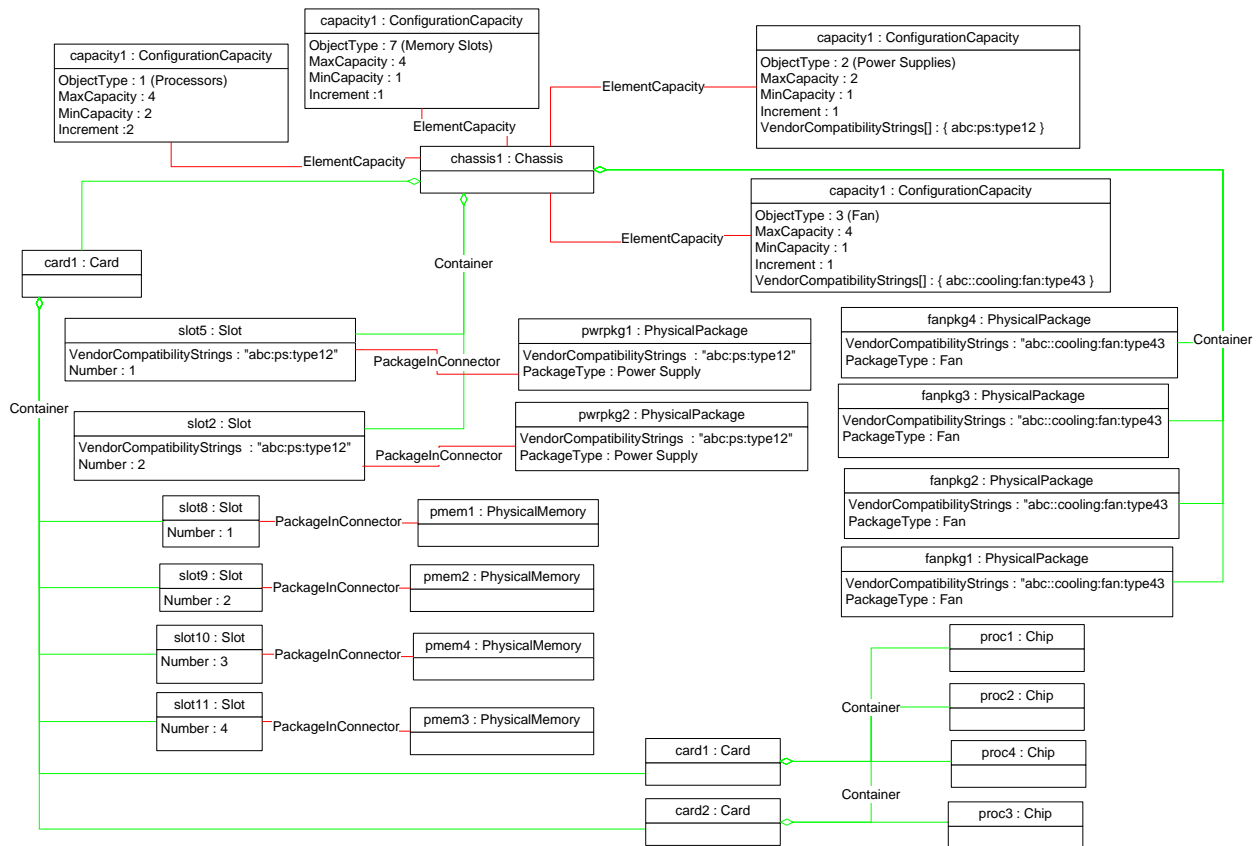
451

452

Figure 6 – Logical Topology

Base Server Profile

453 Figure 7 shows the physical containment hierarchy for the managed system. The *Physical Asset Profile*
 454 has been implemented. The location of the fans within the system is not modeled; instead the fans are
 455 modeled as being directly contained in the main system chassis. The slots or bays in the main chassis
 456 that can contain a power supply are separately modeled (slot5 and slot2). The optional slot and package
 457 compatibility behavior of the *Physical Asset Profile* has been implemented for the power supply slots. The
 458 system memory is installed in four slots on the main system board (card1). The processors (proc1–proc4)
 459 are installed in pairs on separate cards on the main system card. The capacity of the system for
 460 processors, fans, power supplies, and memory is indicated through instances of
 461 CIM_ConfigurationCapacity.

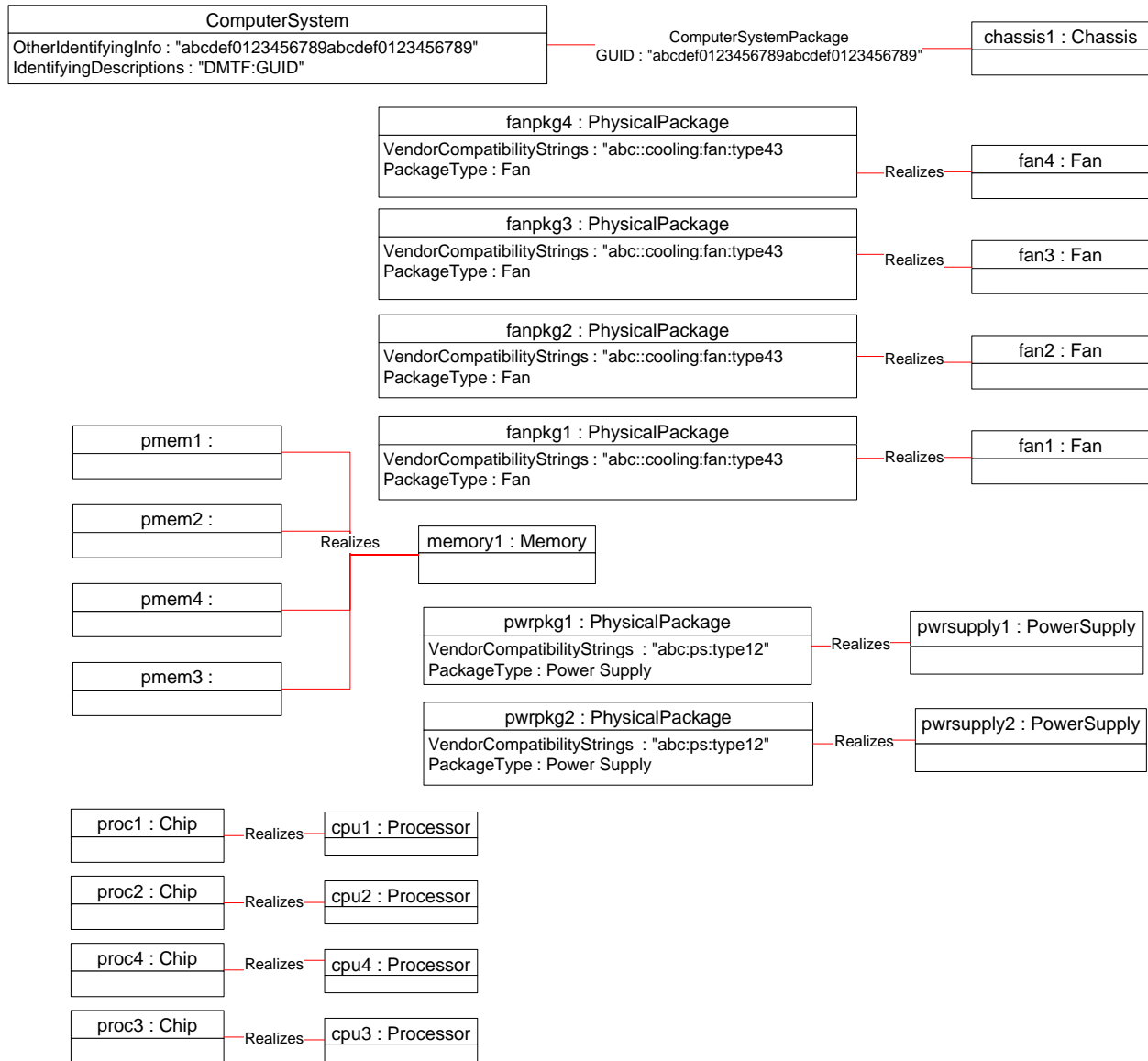


462

463

Figure 7 – Physical Topology

464 Figure 8 shows the relationship between the logical components and their underlying physical packaging.
 465 Each fan, power supply, and processor has a dedicated package. The system memory is realized with
 466 four physical components. The system itself is packaged in a single chassis. To keep the diagram
 467 uncluttered, the CIM_SystemDevice associations have been elided.



468

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

470 **9.2 Determine the System Model and Serial Number**

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

- 473 1) Find an instance of CIM_PhysicalPackage that is associated with the Central Instance through
 474 the CIM_ComputerSystemPackage association.
 475 2) Query the Model and SerialNumber properties of the instance.

Base Server Profile

476 9.3 Power On a System

477 A client can power on a system as follows:

- 478 1) Look for an instance of CIM_EnabledLogicalElementCapabilities that is associated with the
479 Central Instance through the CIM_ElementCapabilities association.
- 480 2) Verify that the CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported property
481 contains the value 2 (Enabled).
- 482 3) Invoke the RequestStateChange() method on the target instance, specifying 2 (Enabled) for the
483 RequestedState parameter.

484 9.4 Power Off a System

485 A client can power off a system as follows:

- 486 1) Look for an instance of CIM_EnabledLogicalElementCapabilities that is associated with the
487 Central Instance through the CIM_ElementCapabilities association.
- 488 2) Verify that the CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported property
489 contains the value 3 (Disabled).
- 490 3) Invoke the RequestStateChange() method on the target instance, specifying 3 (Disabled) for
491 the RequestedState parameter.

492 9.5 Shutdown and Restart a System

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

- 494 1) Look for an instance of CIM_EnabledLogicalElementCapabilities that is associated with the
495 Central Instance through the CIM_ElementCapabilities association.
- 496 2) Verify that the CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported property
497 contains the value 11 (Reset).
- 498 3) Invoke the RequestStateChange() parameter on the target instance, specifying 11 (Reset) for
499 the RequestedState parameter.

500 9.6 Perform System Power Control

501 A client might need to perform power control that is more granular than the functionality available through
502 state management. This is done through power state management. A client can determine whether power
503 state management is available for the system by searching for an instance of
504 CIM_PowerManagementService that is associated with the Central Instance through the
505 CIM_AssociatedPowerManagementService association. The specific use cases for performing power
506 state management are documented in the *Power State Management Profile*.

507 9.7 Determining the System Power State

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

- 509 1) Query the CIM_ComputerSystem.EnabledState property.

510 If the property has the value 2 (Enabled), the system is currently in ACPI state S0 (or equivalent
511 if non-ACPI system). If the property has the value 3 (Disabled), the system is currently in ACPI
512 state S0 (or equivalent if non-ACPI system).
- 513 2) If the CIM_ComputerSystem.EnabledState property has the value 5 (Not Applicable), find the
514 instance of CIM_AssociatedPowerManagementService that references the
515 CIM_ComputerSystem instance.

516 3) Query the value of the CIM_AssociatedPowerManagementService.PowerState property. The
 517 *Power State Management Profile* details the equivalent ACPI states for each value.

518 **9.8 Determine the Number of Processors in the System**

519 When the optional *CPU Profile* is implemented, the client can determine the number of processors in the
 520 system by querying for instances of CIM_Processor that are associated with the Central Instance through
 521 the CIM_SystemDevice association.

522 The client can also use these same steps to find the fans and power supplies installed in the system,
 523 substituting the *Fan Profile* and CIM_Fan, and the *Power Supply Profile* and CIM_PowerSupply
 524 appropriately.

525 **9.9 Determine the Number of Processors that the System Can Hold**

526 When the optional configuration capacity behavior from the *Physical Asset Profile* is implemented for
 527 processors for the system, a client can determine the number of processors that the system can hold as
 528 follows:

- 529 1) Find the instances of CIM_PhysicalPackage that are associated with the Central Instance
 530 through the CIM_ComputerSystemPackage association.
- 531 2) For each instance of CIM_PhysicalPackage, find the instances of CIM_ConfigurationCapacity
 532 that are associated with the CIM_PhysicalPackage instance through the CIM_ElementCapacity
 533 association.
- 534 3) For each instance of CIM_ConfigurationCapacity, if the ObjectType property has the value 1
 535 (Processors), query the MaximumCapacity property and add the value to the total number of
 536 processors that the system can hold.

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

540 **10 CIM Elements**

541 Table 6 shows the instances of CIM Elements for this profile. Instances of the CIM Elements shall be
 542 implemented as described in Table 6. Sections 7 (“Implementation”) and 8 (“Methods”) may impose
 543 additional requirements on these elements.

544 **Table 6 – CIM Elements: Base Server Profile**

Element Name	Requirement	Description
Classes		
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.
Indications		
None defined in this profile		

Base Server Profile

545 10.1 CIM_ComputerSystem

546 An instance of CIM_ComputerSystem is used to represent the system. Table 7 contains the requirements
547 for elements of this class.

548 **Table 7 – Class: CIM_ComputerSystem**

Elements	Requirement	Notes
EnabledState	Mandatory	See sections 7.3.4.1 and 7.3.1.1.
RequestedState	Mandatory	See section 7.3.4.2.
Dedicated	Mandatory	

549 10.2 CIM_ComputerSystemPackage

550 One or more instances of CIM_ComputerSystemPackage associate the CIM_ComputerSystem instance
551 with the CIM_PhysicalPackage instances in which it resides. The constraints specified in Table 8 are in
552 addition to those specified in the *Physical Asset Profile*.

553 **Table 8 – 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..*

554 10.3 CIM_EnabledLogicalElementCapabilities

555 CIM_EnabledLogicalElementCapabilities indicates support for managing the state of the system. Table 9
556 contains the requirements for elements of this class.

557 **Table 9 – Class: CIM_EnabledLogicalElementCapabilities**

Elements	Requirement	Notes
RequestedStatesSupported	Mandatory	See section 7.3.4.3.

558 10.4 CIM_PhysicalPackage

559 One or more instances of CIM_PhysicalPackage represent the physical packaging of the computer
560 system. Other than the existence of at least one instance of CIM_PhysicalPackage, this profile does not
561 specify any constraints for CIM_PhysicalPackage beyond those specified in the *Physical Asset Profile*.

562 **10.5 CIM_RegisteredProfile**

563 CIM_RegisteredProfile identifies the *Base Server Profile* in order for a client to determine whether an
564 instance of CIM_ComputerSystem is conformant with this profile. The CIM_RegisteredProfile class is
565 defined by the *Profile Registration Profile*. With the exception of the mandatory values specified for the
566 properties in Table 10, the behavior of the CIM_RegisteredProfile instance is in accordance with the
567 *Profile Registration Profile*.

568 **Table 10 – Class: CIM_RegisteredProfile**

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

569
570
571
572

**ANNEX A
(Informative)

Change Log**

Version	Date	Description

573

574
575
576
577

ANNEX B (informative)

Acknowledgments

578 The authors wish to acknowledge the following people.

579 **Editors:**

- 580 • Christina Shaw – HP
- 581 • Aaron Merkin – IBM

582 **Contributors:**

- 583 • Jon Hass – Dell
- 584 • Khachatur Papanyan – Dell
- 585 • Jeff Hilland – HP
- 586 • Christina Shaw – HP
- 587 • Aaron Merkin – IBM
- 588 • Perry Vincent – Intel
- 589 • John Leung – Intel
- 590