



1

2

3

4

Document Number: DSP1018

Date: 2013-03-28

Version: 1.1.1

5 **Service Processor Profile**

6 **Document Type: Specification**

7 **Document Status: DMTF Standard**

8 **Document Language: en-US**

9

10 Copyright Notice

11 Copyright © 2006–2007, 2011, 2012, 2013 Distributed Management Task Force, Inc. (DMTF). All rights
12 reserved.

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

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27 withdrawn or modified after publication, and shall be indemnified and held harmless by any party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

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

CONTENTS

34	Foreword	4
35	Introduction.....	7
36	1 Scope	9
37	2 Normative References.....	9
38	2.1 Approved References	9
39	2.2 Other References.....	10
40	3 Terms and Definitions	10
41	4 Symbols and Abbreviated Terms	12
42	5 Synopsis	12
43	6 Description	13
44	7 Implementation.....	14
45	7.1 Representing a Service Processor	14
46	7.2 Modeling Service Processor Redundancy (Optional).....	17
47	7.3 Managing Service Processor Time (Optional)	18
48	7.4 User Account Management (Optional)	18
49	7.5 Boot Control Profile (Optional).....	18
50	7.6 CLP Service Profile (Optional).....	18
51	7.7 DHCP Client Profile (Optional)	18
52	7.8 DNS Client Profile (Optional)	18
53	7.9 Ethernet Port Profile (Optional).....	18
54	7.10 Software Inventory Profile (Optional).....	18
55	7.11 Software Update Profile (Optional).....	19
56	7.12 IP Interface Profile (Optional)	19
57	7.13 Physical Asset Profile (Optional)	19
58	7.14 Record Log Profile (Optional)	19
59	7.15 Sensors Profile (Optional).....	19
60	7.16 Power State Management Profile (Optional)	19
61	7.17 Shared Device Management Profile (Optional)	19
62	7.18 SMASH Collections Profile (Optional)	19
63	7.19 SSH Service Profile (Optional)	19
64	7.20 Telnet Service Profile (Optional).....	20
65	7.21 Text Console Redirection Profile (Optional)	20
66	7.22 PCI Device Profile (Optional).....	20
67	8 Methods.....	20
68	8.1 Method: CIM_ComputerSystem.RequestStateChange()	20
69	8.2 Method: CIM_RedundancySet.Failover()	21
70	8.3 Method: CIM_TimeService.ManageTime().....	22
71	8.4 Profile Conventions for Operations	23
72	8.5 CIM_ComputerSystem.....	23
73	8.6 CIM_HostedService	24
74	8.7 CIM_IsSpare	24
75	8.8 CIM_ElementCapabilities	25
76	8.9 CIM_EnabledLogicalElementCapabilities.....	25
77	8.10 CIM_MemberOfCollection	25
78	8.11 CIM_RedundancySet.....	25
79	8.12 CIM_TimeService	25
80	8.13 CIM_ServiceAffectsElement	26
81	9 Use Cases	26
82	9.1 Object Diagrams	26
83	9.2 Reset a Service Processor	29
84	9.3 Retrieve the Service Processor Redundancy Status.....	30
85	9.4 Determine Whether Manual Failover Is Supported	30

86	9.5	Force a Service Processor Failover.....	30
87	9.6	Determine Whether the ElementName Is Modifiable	30
88	9.7	Determining If State Management Is Supported	30
89	10	CIM Elements	31
90	10.1	CIM_ComputerSystem.....	31
91	10.2	CIM_ElementCapabilities	32
92	10.3	CIM_EnabledLogicalElementCapabilities.....	32
93	10.4	CIM_HostedService	32
94	10.5	CIM_IsSpare	33
95	10.6	CIM_MemberOfCollection	33
96	10.7	CIM_OwningCollectionElement.....	33
97	10.8	CIM_RedundancySet.....	34
98	10.9	CIM_RegisteredProfile.....	34
99	10.10	CIM_ServiceAffectsElement	34
100	10.11	CIM_TimeService	35
101	10.12	CIM_ManagementController.....	35
102		ANNEX A (informative) Change Log.....	36
103			

104 **Figures**

105	Figure 1 – Service Processor Profile: Class Diagram.....	14
106	Figure 2 – Base Server	26
107	Figure 3 – Modular System	27
108	Figure 4 – Service Processors before Failover.....	28
109	Figure 5 – Service Processors after Failover.....	29
110		

111 **Tables**

112	Table 1 – Referenced Profiles	12
113	Table 2 – CIM_ComputerSystem.EnabledState Value Description.....	15
114	Table 3 – CIM_ComputerSystem.RequestStateChange() Method: Return Code Values	21
115	Table 4 – CIM_ComputerSystem.RequestStateChange() Method: Parameters.....	21
116	Table 5 – CIM_RedundancySet.Failover() Method: Return Code Values	22
117	Table 6 – CIM_RedundancySet.Failover() Method: Parameters.....	22
118	Table 7 – CIM_TimeService.ManageTime() Method: Return Code Values	23
119	Table 8 – CIM_TimeService.ManageTime() Method: Parameters	23
120	Table 9 – Operations: CIM_ComputerSystem	23
121	Table 10 – Operations: CIM_HostedService	24
122	Table 11 – Operations: CIM_IsSpare	24
123	Table 12 – Operations: CIM_ElementCapabilities	25
124	Table 13 – Operations: CIM_MemberOfCollection	25
125	Table 14 – Operations: CIM_ServiceAffectsElement	26
126	Table 15 – CIM Elements: Service Processor Profile.....	31
127	Table 16 – Class: CIM_ComputerSystem.....	31
128	Table 17 – Class: CIM_ElementCapabilities.....	32
129	Table 18 – Class: CIM_EnabledLogicalElementCapabilities.....	32
130	Table 19 – Class: CIM_HostedService	32
131	Table 20 – Class: CIM_IsSpare	33

132 Table 21 – Class: CIM_MemberOfCollection..... 33

133 Table 22 – Class: CIM_OwningCollectionElement 33

134 Table 23 – Class: CIM_RedundancySet 34

135 Table 24 – Class: CIM_RegisteredProfile 34

136 Table 25 – Class: CIM_ServiceAffectsElement 34

137 Table 26 – Class: CIM_TimeService 35

138 Table 27 – Class: CIM_ManagementController 35

139

Foreword

140 The *Service Processor Profile* (DSP1018) was prepared by the Physical Platform Profiles Working Group
141 and the Server Management Working Group of the DMTF.

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

144 Acknowledgments

145 The authors wish to acknowledge the following people.

146 Editor:

- 147 • Aaron Merkin – IBM
- 148 • Jeff Hilland - HP

149 Contributors:

- 150 • Jon Hass – Dell
- 151 • Khachatur Papanyan – Dell
- 152 • Enoch Suen – Dell
- 153 • Jeff Hilland – HP
- 154 • Christina Shaw – HP
- 155 • Aaron Merkin – IBM
- 156 • Perry Vincent – Intel
- 157 • John Leung – Intel
- 158 • Hemal Shah – Broadcom
- 159 • Satheesh Thomas - AMI

160

161

Introduction

162 The information in this specification should be sufficient for a provider or consumer of this data to identify
163 unambiguously the classes, properties, methods, and values that shall be instantiated and manipulated to
164 represent and manage a service processor that is modeled using the DMTF Common Information Model
165 (CIM) core and extended model definitions.

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

169

Service Processor Profile

170 1 Scope

171 The *Service Processor Profile* is an autonomous profile for modeling service processors.

172 2 Normative References

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

176 2.1 Approved References

- 177 DMTF DSP0004, *CIM Infrastructure Specification 2.5*,
178 http://www.dmtf.org/standards/published_documents/DSP0004_2.5.pdf
- 179 DMTF DSP0200, *CIM Operations over HTTP 1.2*,
180 http://www.dmtf.org/sites/default/files/standards/documents/DSP0200_1.1.0.html
- 181 DMTF DSP1001, *Management Profile Specification Usage Guide 1.0*,
182 http://www.dmtf.org/standards/published_documents/DSP1001_1.0.pdf
- 183 DMTF DSP1004, *Base Server Profile 1.0*,
184 http://www.dmtf.org/standards/published_documents/DSP1004_1.0.pdf
- 185 DMTF DSP1005, *CLP Service Profile 1.0*,
186 http://www.dmtf.org/standards/published_documents/DSP1005_1.0.pdf
- 187 DMTF DSP1006, *SMASH Collections Profile 1.0*,
188 http://www.dmtf.org/standards/published_documents/DSP1006_1.0.pdf
- 189 DMTF DSP1008, *Modular System Profile 1.0*,
190 http://www.dmtf.org/standards/published_documents/DSP1008_1.0.pdf
- 191 DMTF DSP1009, *Sensors Profile 1.0*,
192 http://www.dmtf.org/standards/published_documents/DSP1009_1.0.pdf
- 193 DMTF DSP1010, *Record Log Profile 1.0*,
194 http://www.dmtf.org/standards/published_documents/DSP1010_1.0.pdf
- 195 DMTF DSP1011, *Physical Asset Profile 1.0*,
196 http://www.dmtf.org/standards/published_documents/DSP1011_1.0.pdf
- 197 DMTF DSP1012, *Boot Control Profile 1.0*,
198 http://www.dmtf.org/standards/published_documents/DSP1012_1.0.pdf
- 199 DMTF DSP1014, *Ethernet Port Profile 1.0*,
200 http://www.dmtf.org/standards/published_documents/DSP1014_1.0.pdf
- 201 DMTF DSP1016, *Telnet Service Profile 1.0*,
202 http://www.dmtf.org/standards/published_documents/DSP1016_1.0.pdf
- 203 DMTF DSP1017, *SSH Service Profile 1.0*,
204 http://www.dmtf.org/standards/published_documents/DSP1017_1.0.pdf

- 205 DMTF DSP1021, *Shared Device Management Profile 1.0*,
206 http://www.dmtf.org/standards/published_documents/DSP1021_1.0.pdf
- 207 DMTF DSP1023, *Software Inventory Profile 1.0*,
208 http://www.dmtf.org/standards/published_documents/DSP1023_1.0.pdf
- 209 DMTF DSP1024, *Text Console Redirection Profile 1.0*,
210 http://www.dmtf.org/standards/published_documents/DSP1024_1.0.pdf
- 211 DMTF DSP1025, *Software Update Profile 1.0*,
212 http://www.dmtf.org/standards/published_documents/DSP1025_1.0.pdf
- 213 DMTF DSP1027, *Power State Management Profile 1.0*,
214 http://www.dmtf.org/standards/published_documents/DSP1027_1.0.pdf
- 215 DMTF DSP1033, *Profile Registration Profile 1.0*,
216 http://www.dmtf.org/standards/published_documents/DSP1033_1.0.pdf
- 217 DMTF DSP1034, *Simple Identity Management Profile 1.0*,
218 http://www.dmtf.org/standards/published_documents/DSP1034_1.0.pdf
- 219 DMTF DSP1036, *IP Interface Profile 1.0*,
220 http://www.dmtf.org/standards/published_documents/DSP1036_1.0.pdf
- 221 DMTF DSP1037, *DHCP Client Profile 1.0*,
222 http://www.dmtf.org/standards/published_documents/DSP1037_1.0.pdf
- 223 DMTF DSP1038, *DNS Client Profile 1.0*,
224 http://www.dmtf.org/standards/published_documents/DSP1038_1.0.pdf
- 225 DMTF DSP1039, *Role Based Authorization Profile 1.0*,
226 http://www.dmtf.org/standards/published_documents/DSP1039_1.0.pdf
- 227 DMTF DSP1075, *PCI Device Profile 1.0*,
228 http://www.dmtf.org/standards/published_documents/DSP1075_1.0.pdf

229 2.2 Other References

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

232 3 Terms and Definitions

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

235 3.1

236 **can**

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

238 3.2

239 **cannot**

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

241 3.3

242 **conditional**

243 indicates requirements to be followed strictly to conform to the document when the specified conditions
244 are met

- 245 **3.4**
246 **mandatory**
247 indicates requirements to be followed strictly to conform to the document and from which no deviation is
248 permitted
- 249 **3.5**
250 **may**
251 indicates a course of action permissible within the limits of the document
- 252 **3.6**
253 **need not**
254 indicates a course of action permissible within the limits of the document
- 255 **3.7**
256 **optional**
257 indicates a course of action permissible within the limits of the document
- 258 **3.8**
259 **referencing profile**
260 indicates a profile that owns the definition of this class and can include a reference to this profile in its
261 "Referenced Profiles" table
- 262 **3.9**
263 **shall**
264 indicates requirements to be followed strictly to conform to the document and from which no deviation is
265 permitted
- 266 **3.10**
267 **shall not**
268 indicates requirements to be followed strictly to conform to the document and from which no deviation is
269 permitted
- 270 **3.11**
271 **should**
272 indicates that among several possibilities, one is recommended as particularly suitable, without
273 mentioning or excluding others, or that a certain course of action is preferred but not necessarily required
- 274 **3.12**
275 **should not**
276 indicates that a certain possibility or course of action is deprecated but not prohibited
- 277 **3.13**
278 **unspecified**
279 indicates that this profile does not define any constraints for the referenced CIM element or operation
- 280 **3.14**
281 **service processor**
282 a specialized device dedicated to management
- 283 **3.15**
284 **standby service processor**
285 an instance of CIM_ComputerSystem that represents a standby service processor of a redundancy set

286 4 Symbols and Abbreviated Terms

287 None.

288 5 Synopsis

289 **Profile Name:** Service Processor

290 **Version:** 1.1.1

291 **Organization:** DMTF

292 **CIM Schema Version:** 2.20

293 **Central Class:** CIM_ComputerSystem

294 **Scoping Class:** CIM_ComputerSystem

295 Table 1 identifies profiles on which this profile has a dependency.

296

Table 1 – Referenced Profiles

Profile Name	Organization	Version	Relationship	Behavior
Simple Identity Management	DMTF	1.0	Optional	See 7.3.
Boot Control	DMTF	1.0	Optional	See 7.5.
CLP Service	DMTF	1.0	Optional	See 7.6.
DHCP Client	DMTF	1.0	Optional	See 7.7.
DNS Client	DMTF	1.0	Optional	See 7.8.
Ethernet Port	DMTF	1.0	Optional	See 7.9.
Software Inventory	DMTF	1.0	Optional	See 7.10.
Software Update	DMTF	1.0	Optional	See 7.11.
IP Interface	DMTF	1.0	Optional	See 7.12.
Physical Asset	DMTF	1.0	Optional	See 7.13.
Profile Registration	DMTF	1.0	Mandatory	None
Record Log	DMTF	1.0	Optional	See 7.14.
Role Based Authorization	DMTF	1.0	Optional	See 7.3.
Sensors	DMTF	1.0	Optional	See 7.15.
Power State Management	DMTF	1.0	Optional	See 7.16.
Shared Device Management	DMTF	1.0	Optional	See 7.17.
SMASH Collections	DMTF	1.0	Optional	See 7.18.
SSH Service	DMTF	1.0	Optional	See 7.19.
Telnet Service	DMTF	1.0	Optional	See 7.20.
Text Console Redirection	DMTF	1.0	Optional	See 7.21.
PCI Device	DMTF	1.0	Optional	See 7.22.

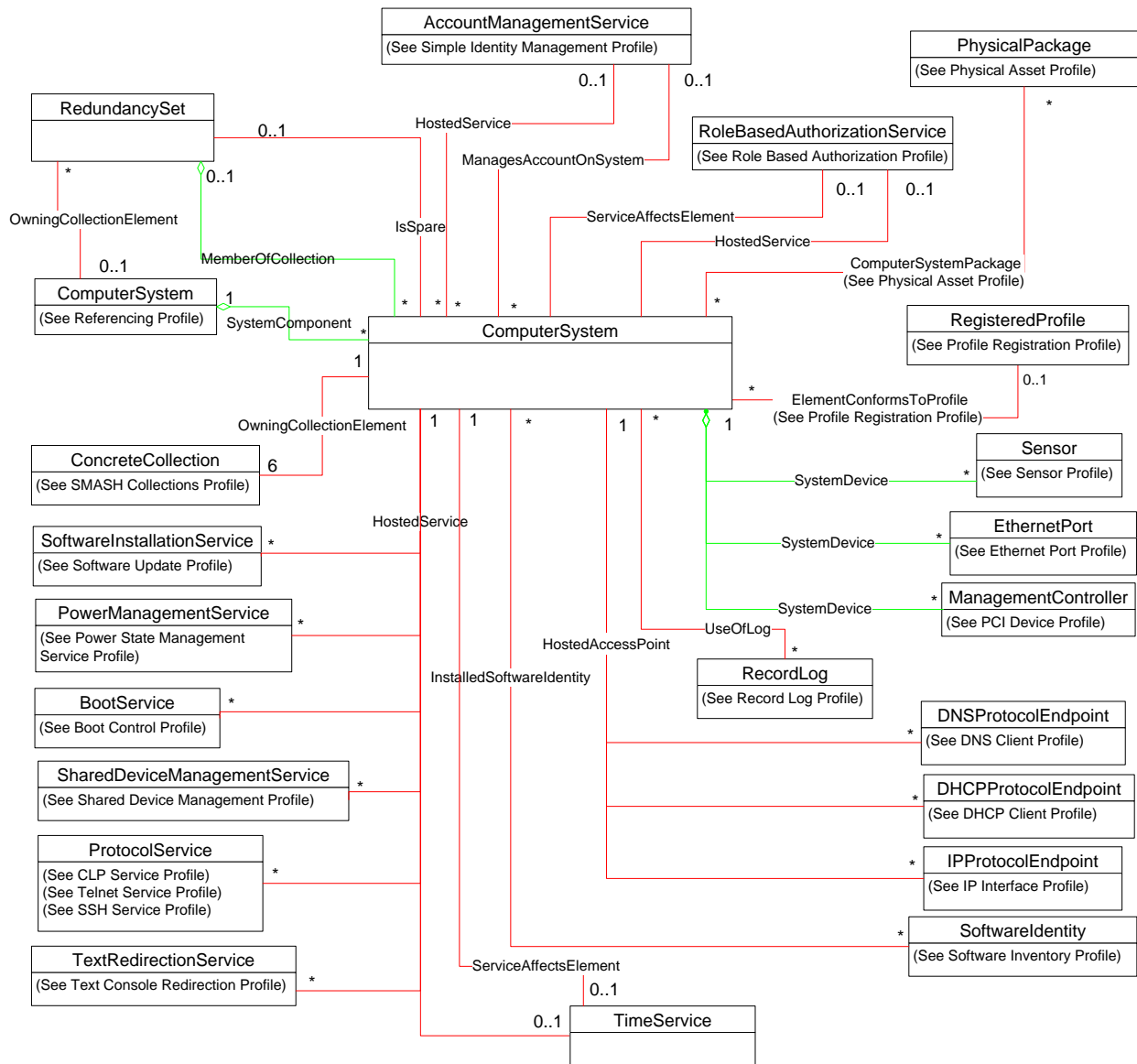
297 **6 Description**

298 The *Service Processor Profile* describes the management and configuration of a service processor for a
299 computer system. The computer system may be contained in a single chassis or comprise a more
300 complex modular system with multiple chassis or a blade system. This description includes modeling
301 redundant service processors.

302 Some examples of the service processors are:

- 303 • management processor (MP)
- 304 • service processor (SP)
- 305 • baseboard management controller (BMC)
- 306 • chassis manager

307 Figure 1 represents the class schema for the *Service Processor Profile*. For simplicity, the prefix CIM_
308 has been removed from the names of the classes.



309

310

Figure 1 – Service Processor Profile: Class Diagram

311 7 Implementation

312 This section details the requirements related to the arrangement of instances and their properties for
 313 implementations of this profile. All required methods and operations are described in clause 8. Required
 314 CIM elements are described in clause 10.

315 7.1 Representing a Service Processor

316 A service processor shall be represented with an instance of CIM_ComputerSystem.

317 **7.1.1 CIM_ComputerSystem.EnabledState**

318 Table 2 describes the mapping between the values of the CIM_ComputerSystem.EnabledState property
 319 and the corresponding description of the state of the service processor. The EnabledState property shall
 320 match the values that are specified in Table 2. When the RequestStateChange() method executes but
 321 does not complete successfully, and the service processor is in an indeterminate state, the EnabledState
 322 property shall have value of 5 (Not Applicable). The value of the EnabledState property may also change
 323 as a result of change to the service processor's enabled state by non-CIM implementation.

324 **Table 2 – CIM_ComputerSystem.EnabledState Value Description**

Value	Description	Extended Description
2	Enabled	The service processor shall be enabled.
3	Disabled	The service processor shall be disabled.
5	Not Applicable	The service processor state is indeterminate, or service processor state management is not supported.
6	Enabled but Offline	The service processor shall be enabled but inactive (used in redundant configuration; see 7.2.4).

325 **7.1.2 Service Processor State Management Is Supported — Conditional**

326 Support for managing the state of the service processor is optional behavior. This section describes the
 327 CIM elements and behaviors that shall be implemented when this behavior is supported.

328 **7.1.2.1 CIM_EnabledLogicalElementCapabilities**

329 When state management is supported, exactly one instance of CIM_EnabledLogicalElementCapabilities
 330 shall be associated with the CIM_ComputerSystem instance that represents a service processor through
 331 an instance of CIM_ElementCapabilities.

332 **7.1.2.1.1 CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported**

333 The RequestedStatesSupported property may contain zero or more of the following values: 2 (Enabled),
 334 3 (Disabled), 6 (Offline), or 11 (Reset).

335 **7.1.2.2 CIM_ComputerSystem.RequestedState**

336 When the CIM_ComputerSystem.RequestStateChange() method is successfully invoked, the value of the
 337 RequestedState property shall be the value of the RequestedState parameter. If the method is not
 338 successfully invoked, the value of the RequestedState property is indeterminate.

339 The CIM_ComputerSystem.RequestedState property shall have one of the values specified in the
 340 CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported property or a value of 5 (No
 341 Change).

342 **7.1.2.3 CIM_ComputerSystem.EnabledState**

343 When the RequestedState parameter has a value of 2 (Enabled) or 3 (Disabled) and the
 344 CIM_ComputerSystem.RequestStateChange() method completes successfully, the value of the
 345 EnabledState property shall equal the value of the CIM_ComputerSystem.RequestedState property.

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

347 **7.1.3 Service Processor State Management Is Not Supported**

348 This section describes the CIM elements and behaviors that shall be implemented when management of
 349 the service processor state is not supported.

350 **7.1.3.1 CIM_EnabledLogicalElementCapabilities**

351 When state management is not supported, exactly one instance of
352 CIM_EnabledLogicalElementCapabilities may be associated with the CIM_ComputerSystem instance that
353 represents a service processor through an instance of CIM_ElementCapabilities.

354 **7.1.3.1.1 CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported**

355 The CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported property shall not contain any
356 values.

357 **7.1.3.2 CIM_ComputerSystem.RequestedState**

358 The RequestedState property shall have the value 12 (Not Applicable).

359 **7.1.4 Modifying ElementName Is Supported — Conditional**

360 The CIM_ComputerSystem.ElementName property may support being modified by the ModifyInstance
361 operation. See 8.5.1. This behavior is conditional. This section describes the CIM elements and behavior
362 requirements when an implementation supports client modification of the
363 CIM_ComputerSystem.ElementName property.

364 **7.1.4.1 CIM_EnabledLogicalElementCapabilities**

365 An instance of CIM_EnabledLogicalElementCapabilities shall be associated with the
366 CIM_ComputerSystem instance through an instance of CIM_ElementCapabilities.

367 **7.1.4.1.1 CIM_EnabledLogicalElementCapabilities.ElementNameEditSupported**

368 The ElementNameEditSupported property shall have a value of TRUE.

369 **7.1.4.1.2 CIM_EnabledLogicalElement.MaxElementNameLen**

370 The MaxElementNameLen property shall be implemented.

371 **7.1.5 Modifying ElementName Is Not Supported**

372 This section describes the CIM elements and behaviors that shall be implemented when the
373 CIM_ComputerSystem.ElementName does not support being modified by the ModifyInstance operation.

374 **7.1.5.1 CIM_EnabledLogicalElementCapabilities**

375 An instance of CIM_EnabledLogicalElementCapabilities may be associated with the
376 CIM_ComputerSystem instance through an instance of CIM_ElementCapabilities.

377 **7.1.5.1.1 CIM_EnabledLogicalElementCapabilities.ElementNameEditSupported**

378 The ElementNameEditSupported shall have a value of FALSE.

379 **7.1.5.1.2 CIM_EnabledLogicalElement.MaxElementNameLen**

380 The MaxElementNameLen property may be implemented. The MaxElementNameLen property is
381 irrelevant in this context.

382 **7.1.6 Representing the Physical Packaging (Optional)**

383 Support for representing the physical packaging of the service processor is optional. The physical
384 packaging may be modeled using one or more instances of CIM_PhysicalElement in accordance with the
385 [Physical Asset Profile](#).

386 7.2 Modeling Service Processor Redundancy (Optional)

387 Modeling of service processor redundancy is optional. When service processor redundancy is supported,
388 the requirements in this section apply.

389 At least one instance of CIM_RedundancySet shall exist.

390 7.2.1 Relationship between Redundancy Set and Redundant Service Processors

391 Each CIM_ComputerSystem instance that represents a service processor participating in the redundancy
392 shall be associated with the CIM_RedundancySet instance through the CIM_MemberOfCollection
393 association. Each instance of CIM_ComputerSystem that is associated with the CIM_RedundancySet
394 instance through the CIM_MemberOfCollection association shall be associated with the same instance of
395 CIM_ComputerSystem through the CIM_SystemComponent association where the value of the
396 CIM_SystemComponent.PartComponent property is the instance of CIM_ComputerSystem that is
397 associated with the CIM_RedundancySet.

398 7.2.2 Relationship between Redundancy Set and Containing System

399 When the CIM_ComputerSystem instance that represents a service processor is associated with another
400 CIM_ComputerSystem instance through the CIM_SystemComponent association where the value of the
401 CIM_SystemComponentPartComponent property is the CIM_ComputerSystem instance that represents
402 the service processor, the CIM_RedundancySet instance shall be associated with the
403 CIM_ComputerSystem instance that is the value of the CIM_SystemComponent.GroupComponent
404 property through the CIM_OwningCollectionElement association.

405 7.2.3 Active / Active Redundancy

406 When the CIM_RedundancySet.TypeOfSet property contains a value of 3 (Load Balanced) or 2 (N+1),
407 the CIM_ComputerSystem instances that are associated the CIM_RedundancySet instance shall comply
408 with the following requirements:

- 409 • The CIM_ComputerSystem instances shall not be associated with the CIM_RedundancySet
410 instance through the CIM_IsSpare association.
- 411 • For each instance of CIM_ComputerSystem, the CIM_ComputerSystem.EnabledState property
412 shall not have the value 6 (Enabled but Offline).

413 7.2.4 Active / Standby Redundancy

414 When the CIM_RedundancySet.TypeOfSet property contains a value of 4 (Sparing) or 5 (Limited
415 Sparing), one or more standby service processor s may exist. Each standby service processor shall be
416 associated to the CIM_RedundancySet instance through the CIM_IsSpare association.

417 Each standby service processor shall comply with one of the following requirements:

- 418 • When the CIM_ComputerSystem.EnabledState property has the value 6 (Enabled but Offline),
419 the SpareStatus property of the referencing CIM_IsSpare instance shall have the value 2 (Hot
420 Standby).
- 421 • When the CIM_ComputerSystem.EnabledState property has the value 3 (Disabled), the
422 SpareStatus property of the referencing CIM_IsSpare instance shall have the value 3 (Cold
423 Standby).
- 424 • When the CIM_ComputerSystem.EnabledState property has a value other than 3 (Disabled) or
425 6 (Enabled but Offline), the SpareStatus property of the referencing CIM_IsSpare instance shall
426 have the value 0 (Unknown).

427 **7.3 Managing Service Processor Time (Optional)**

428 A service processor can maintain an internal clock. This internal clock provides the service processor with
429 the current time (for example, to provide time stamps for log entries). Management of the current time of
430 the service processor may be supported. This behavior is optional. When management of the current time
431 of the service processor is supported, the requirements specified in this section shall be met.

432 An instance of CIM_TimeService shall be associated with the Central Instance through the
433 CIM_HostedService association. The instance of CIM_TimeService shall also be associated with the
434 Central Instance through the CIM_ServiceAffectsElement association.

435 **7.4 User Account Management (Optional)**

436 The [Simple Identity Management Profile](#) and the [Role Based Authorization Profile](#) may be implemented to
437 model user access to the service processor. When the [Simple Identity Management Profile](#) is
438 implemented, an instance of CIM_AccountManagementService shall be associated with the Central
439 Instance through the CIM_HostedService association. When the [Role Based Authorization Profile](#) is
440 implemented, an instance of CIM_RoleBasedAuthorizationService shall be associated with the Central
441 Instance through the CIM_HostedService association.

442 **7.5 Boot Control Profile (Optional)**

443 The [Boot Control Profile](#) may be implemented to model the ability of the service processor to manage its
444 own boot configuration or that of the systems it managed. If the [Boot Control Profile](#) is implemented, an
445 instance of CIM_BootService shall be associated with the Central Instance through the
446 CIM_HostedService association.

447 **7.6 CLP Service Profile (Optional)**

448 The [CLP Service Profile](#) may be implemented to model a CLP service hosted on the service processor.
449 When the [CLP Service Profile](#) is implemented, at least one instance of CIM_ProtocolService shall be
450 associated with the Central Instance through an instance of CIM_HostedService.

451 **7.7 DHCP Client Profile (Optional)**

452 The [DHCP Client Profile](#) may be implemented to model the DHCP client of a service processor. When the
453 [DHCP Client Profile](#) is implemented, at least one instance of CIM_DHCPProtocolEndpoint shall be
454 associated with the Central Instance through an instance of CIM_HostedAccessPoint.

455 **7.8 DNS Client Profile (Optional)**

456 The [DNS Client Profile](#) may be implemented to model the DNS client of a service processor. When the
457 [DNS Client Profile](#) is implemented, at least one instance of CIM_DNSProtocolEndpoint shall be
458 associated with the Central Instance through an instance of CIM_HostedAccessPoint.

459 **7.9 Ethernet Port Profile (Optional)**

460 The [Ethernet Port Profile](#) may be implemented to model an Ethernet interface of a service processor.
461 When the [Ethernet Port Profile](#) is implemented, at least one instance of CIM_EthernetPort shall be
462 associated with the Central Instance through an instance of CIM_SystemDevice.

463 **7.10 Software Inventory Profile (Optional)**

464 The [Software Inventory Profile](#) may be implemented to model the software version information of the
465 service processor. When the [Software Inventory Profile](#) is implemented, at least one instance of
466 CIM_SoftwareIdentity shall be associated with the Central Instance of this profile through an instance of
467 CIM_InstalledSoftwareIdentity.

468 7.11 Software Update Profile (Optional)

469 The [Software Update Profile](#) may be implemented to model the ability of the service processor to update
470 software installed on one or more components of managed systems, including the service processor
471 itself. When the [Software Update Profile](#) is implemented, an instance of CIM_SoftwareInstallationService
472 shall be associated with the Central Instance through an instance of CIM_HostedService.

473 7.12 IP Interface Profile (Optional)

474 The [IP Interface Profile](#) may be implemented to model the IP interface of a service processor. When the
475 [IP Interface Profile](#) is implemented, at least one instance of CIM_IPProtocolEndpoint shall be associated
476 with the Central Instance through an instance of CIM_HostedAccessPoint.

477 7.13 Physical Asset Profile (Optional)

478 The [Physical Asset Profile](#) may be implemented to model the physical package and physical asset
479 information of a service processor. When the [Physical Asset Profile](#) is implemented, at least one instance
480 of CIM_PhysicalPackage shall be associated with the Central Instance through an instance of
481 CIM_ComputerSystemPackage.

482 7.14 Record Log Profile (Optional)

483 The [Record Log Profile](#) may be implemented to model one or more logs of the service processor. When
484 the [Record Log Profile](#) is implemented, an instance of CIM_RecordLog shall be associated with Central
485 Instance through an instance of CIM_UseOfLog.

486 7.15 Sensors Profile (Optional)

487 The [Sensors Profile](#) may be implemented to model the sensors of the service processor. When the
488 [Sensors Profile](#) is implemented, at least one instance of CIM_Sensor or CIM_NumericSensor shall be
489 associated with the Central Instance through an instance of CIM_SystemDevice.

490 7.16 Power State Management Profile (Optional)

491 The [Power State Management Profile](#) may be implemented to model the ability of the service processor
492 to perform power control operations for the managed system or the service processor itself. When the
493 [Power State Management Profile](#) is implemented, an instance of CIM_PowerManagementService shall
494 be associated with the Central Instance through an instance of CIM_HostedService.

495 7.17 Shared Device Management Profile (Optional)

496 The [Shared Device Management Profile](#) may be implemented to model the ability of the service
497 processor to control shared devices of a modular system. When the [Shared Device Management Profile](#)
498 is implemented, an instance of CIM_SharedDeviceManagementService shall be associated with the
499 Central Instance through an instance of CIM_HostedService.

500 7.18 SMASH Collections Profile (Optional)

501 The [SMASH Collections Profile](#) may be implemented. When the [SMASH Collections Profile](#) is
502 implemented, each instance of CIM_ConcreteCollection that is defined by the [SMASH Collections Profile](#)
503 shall be associated with the Central Instance through an instance of CIM_OwningCollectionElement.

504 7.19 SSH Service Profile (Optional)

505 The [SSH Service Profile](#) may be implemented to model an SSH service hosted on the service processor.
506 When the [SSH Service Profile](#) is implemented, at least one instance of CIM_ProtocolService shall be

507 associated with the Central Instance through an instance of CIM_HostedService where the
508 CIM_ProtocolService.Protocol property has the value 2 (SSH).

509 **7.20 Telnet Service Profile (Optional)**

510 The [Telnet Service Profile](#) may be implemented to model a Telnet service hosted on the service
511 processor. When the [Telnet Service Profile](#) is implemented, at one instance of CIM_ProtocolService shall
512 be associated with the Central Instance through an instance of CIM_HostedService where the
513 CIM_ProtocolService.Protocol property has the value 3 (Telnet).

514 **7.21 Text Console Redirection Profile (Optional)**

515 The [Text Console Redirection Profile](#) may be implemented to model the ability of the service processor to
516 provide text console redirection for managed systems. When the [Text Console Redirection Profile](#) is
517 implemented, at least one instance of CIM_TextRedirectionService shall be associated with the Central
518 Instance through an instance of CIM_HostedService.

519 **7.22 PCI Device Profile (Optional)**

520 The [PCI Device Profile](#) may be implemented to model the ability of the service processor to provide PCI
521 configuration information for managed systems. When the [PCI Device Profile](#) is implemented and the
522 ServiceProcessor is modeled as a PCI device, at least one instance of CIM_ManagementController shall
523 be associated with the Central Instance of this profile through an instance of CIM_SystemDevice and the
524 CIM_ManagementController shall be associated with at least one instance of CIM_PCIDevice through an
525 instance of CIM_ConcreteIdentity.

526 **8 Methods**

527 This section details the requirements for supporting intrinsic operations and extrinsic methods for the CIM
528 elements defined by this profile.

529 **8.1 Method: CIM_ComputerSystem.RequestStateChange()**

530 Invocation of the CIM_ComputerSystem.RequestStateChange() method changes the element's state to
531 the value specified in the RequestedState parameter.

532 Return values for the RequestStateChange() method are specified in Table 3. Parameters for the
533 RequestStateChange() method are specified in Table 4.

534 The RequestStateChange() method shall be implemented and shall not return a value of 1 (Not
535 Supported) when state management of the service processor is supported (see 7.1.2).

536 When the RequestedState parameter has a value of 6 (Offline) and the CIM_ComputerSystem instance is
537 not a standby service processor, the RequestStateChange() method shall return a value of 2 (Error
538 Occurred).

539 Invoking the RequestStateChange() method multiple times could result in earlier requests being
540 overwritten or lost.

541 No standard messages are defined for this method.

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

Value	Description
0	Request was successfully executed.
1	Method is not supported in the implementation.
2	Error occurred.
4096	Job started.

543 **Table 4 – CIM_ComputerSystem.RequestStateChange() Method: Parameters**

Qualifiers	Name	Type	Description/Values
IN, REQ	RequestedState	uint16	2 (Enabled) 3 (Disabled), see 8.1.1 6 (Offline), see 8.1.1 11 (Reset)
OUT	Job	CIM_ConcreteJob REF	Returned if job started
IN, REQ	TimeoutPeriod	Datetime	Client specified maximum amount of time the transition to a new state is supposed to take: 0 or NULL – No time requirements <interval> – Maximum time allowed

544 **8.1.1 RequestStateChange() for the Standby Service Processor**

545 After the successful execution of the RequestStateChange() method on the standby service processor
546 with the RequestedState parameter set to 6 (Offline), the SpareStatus property of the referenced
547 CIM_IsSpare association shall have a value of 2 (Hot Standby).

548 After the successful execution of the RequestStateChange() method on the standby service processor
549 with the RequestedState parameter set to 3 (Disabled), the SpareStatus property of the referenced
550 CIM_IsSpare association shall have value of 3 (Cold Standby).

551 **8.2 Method: CIM_RedundancySet.Failover()**

552 The CIM_RedundancySet.Failover() method forces a failover from one member of a
553 CIM_RedundancySet collection to another. After the successful execution of the method, the service
554 processor that is represented by the CIM_ComputerSystem instance referenced by the FailoverFrom
555 parameter becomes inactive. The service processor that is represented by CIM_ComputerSystem
556 instance referenced by the FailoverTo parameter takes over as the active service processor.

557 The Failover() method may be supported if the FailoverSupported property of at least one instance of
558 CIM_IsSpare that references the CIM_RedundancySet instance has a value of 3 (Manual) or 4 (Both
559 Manual and Automatic).

560 The Failover() method shall not be supported if the FailoverSupported property of every instance of
561 CIM_IsSpare that references the CIM_RedundancySet instance has a value of 2 (Automatic).

562 The execution of the Failover() method shall return a value of 2 (Error Occurred) under the following
563 circumstances:

- 564 • The CIM_ComputerSystem instance that is referenced by the FailoverTo parameter is not a
565 standby service processor.

- 566 • The CIM_ComputerSystem instance that is referenced by the FailoverFrom parameter is not
567 associated with the CIM_RedundancySet instance only through the CIM_MemberOfCollection
568 association.

569 After the successful execution of the Failover() method, the following events occur:

- 570 • The CIM_ComputerSystem that is referenced by the FailoverTo parameter shall take over as the
571 active service processor.
- 572 • The CIM_ComputerSystem instance that is referenced by the FailoverTo parameter shall be
573 associated with the CIM_RedundancySet instance only through the CIM_MemberOfCollection
574 association.
- 575 • The CIM_ComputerSystem instance that is referenced by the FailoverFrom parameter shall
576 become a standby service processor. This instance will conform to the requirements for a
577 standby service processor specified in 7.2.4.
- 578 • When management of the service processor state is supported, the CIM_ComputerSystem
579 instance that is referenced by the FailoverFrom parameter shall not have an EnabledState
580 property value of 2 (Enabled) but may have a value of 6 (Enabled but Offline).

581 Return code values for the CIM_RedundancySet.Failover() method are specified in Table 5. Parameters
582 for the CIM_RedundancySet.Failover() method are specified in Table 6. No standard messages are
583 defined for this method.

584 **Table 5 – CIM_RedundancySet.Failover() Method: Return Code Values**

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

585 **Table 6 – CIM_RedundancySet.Failover() Method: Parameters**

Qualifiers	Name	Type	Description/Values
IN, REQ	FailoverFrom	CIM_ManagedElement REF	The redundant element that will become inactive
IN, REQ	FailoverTo	CIM_ManagedElement REF	The redundant element that will become active and take over the inactivated element

586 **8.3 Method: CIM_TimeService.ManageTime()**

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

592 Detailed requirements of the CIM_TimeService() method are specified in Table 7 and Table 8. No
593 standard messages are defined for this method.

594

Table 7 – CIM_TimeService.ManageTime() Method: Return Code Values

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

595

Table 8 – CIM_TimeService.ManageTime() Method: Parameters

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

596 **8.4 Profile Conventions for Operations**

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

599 The default list of operations is as follows:

- 600 • GetInstance
- 601 • Associators
- 602 • AssociatorNames
- 603 • References
- 604 • ReferenceNames
- 605 • EnumerateInstances
- 606 • EnumerateInstanceNames

607 **8.5 CIM_ComputerSystem**

608 Table 9 lists implementation requirements for operations. If implemented, these operations shall be
 609 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 9, all operations in
 610 the default list in 8.4 shall be implemented as defined in [DSP0200](#).

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

612 **Table 9 – Operations: CIM_ComputerSystem**

Operation	Requirement	Messages
ModifyInstance	Optional. See 8.5.1.	None

613 **8.5.1 CIM_ComputerSystem — ModifyInstance**

614 This section details the requirements for the ModifyInstance operation applied to an instance of
 615 CIM_ComputerSystem. The ModifyInstance operation may be supported.

616 The ModifyInstance operation shall be supported and the CIM_ComputerSystem.ElementName property
 617 shall be modifiable when the ElementNameEditSupported property of the
 618 CIM_EnabledLogicalElementCapabilities instance that is associated with the CIM_ComputerSystem
 619 instance has a value of TRUE. See 8.5.1.1.

620 **8.5.1.1 CIM_ComputerSystem.ElementName**

621 When the ElementNameEditSupported property of the CIM_EnabledLogicalElementCapabilities instance
 622 that is associated with the CIM_ComputerSystem instance has a value of TRUE, the implementation shall
 623 allow the ModifyInstance operation to change the value of the ElementName property of the
 624 CIM_ComputerSystem instance. The ModifyInstance operation shall enforce the length restriction
 625 specified in the MaxElementNameLen property of the CIM_EnabledLogicalElementCapabilities instance.

626 When the ElementNameEditSupported property of the CIM_EnabledLogicalElementCapabilities instance
 627 has a value of FALSE, the implementation shall not allow the ModifyInstance operation to change the
 628 value of the ElementName property of the CIM_ComputerSystem instance.

629 **8.6 CIM_HostedService**

630 Table 10 lists implementation requirements for operations. If implemented, these operations shall be
 631 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 10, all operations
 632 in the default list in 8.4 shall be implemented as defined in [DSP0200](#).

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

634 **Table 10 – Operations: CIM_HostedService**

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

635 **8.7 CIM_IsSpare**

636 Table 11 lists implementation requirements for operations. If implemented, these operations shall be
 637 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 11, all operations
 638 in the default list in 8.4 shall be implemented as defined in [DSP0200](#).

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

640 **Table 11 – Operations: CIM_IsSpare**

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

641 8.8 CIM_ElementCapabilities

642 Table 12 lists implementation requirements for operations. If implemented, these operations shall be
 643 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 12, all operations
 644 in the default list in 8.4 shall be implemented as defined in [DSP0200](#).

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

646 **Table 12 – Operations: CIM_ElementCapabilities**

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

647 8.9 CIM_EnabledLogicalElementCapabilities

648 All operations in the default list in 8.4 shall be implemented as defined in [DSP0200](#).

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

650 8.10 CIM_MemberOfCollection

651 Table 13 lists implementation requirements for operations. If implemented, these operations shall be
 652 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 13, all operations
 653 in the default list in 8.4 shall be implemented as defined in [DSP0200](#).

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

655 **Table 13 – Operations: CIM_MemberOfCollection**

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

656 8.11 CIM_RedundancySet

657 All operations in the default list in 8.4 shall be implemented as defined in [DSP0200](#).

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

659 8.12 CIM_TimeService

660 All operations in the default list in 8.4 shall be implemented as defined in [DSP0200](#).

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

662 **8.13 CIM_ServiceAffectsElement**

663 Table 14 lists implementation requirements for operations. If implemented, these operations shall be
 664 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 14, all operations
 665 in the default list in 8.4 shall be implemented as defined in [DSP0200](#).

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

667 **Table 14 – Operations: CIM_ServiceAffectsElement**

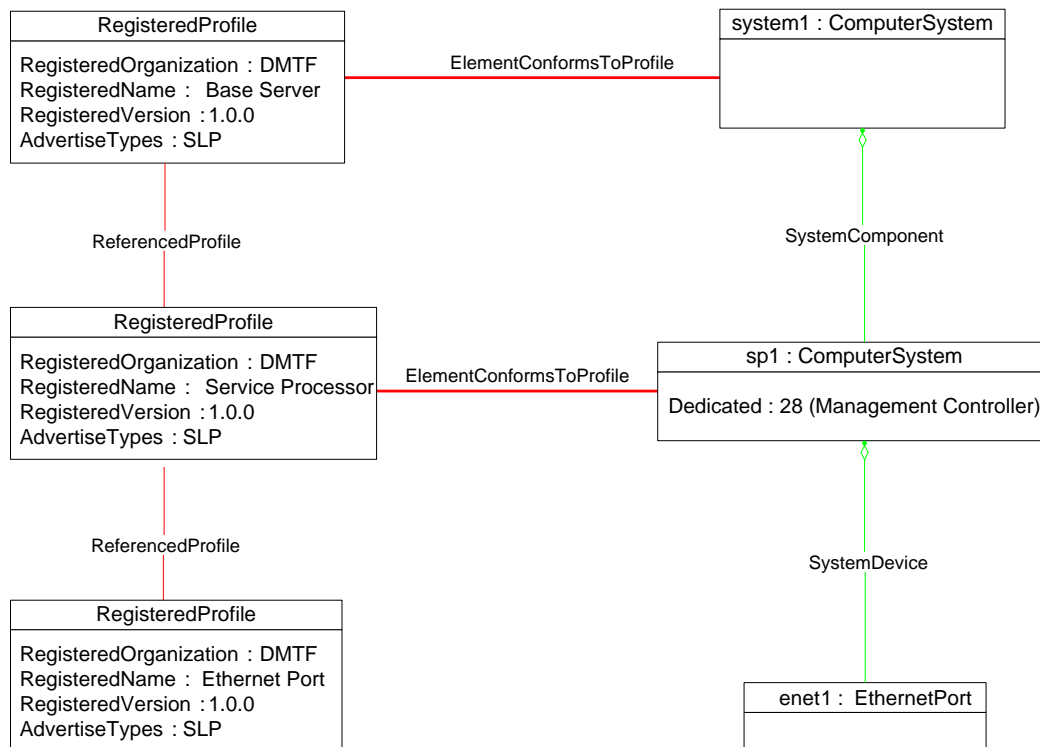
Operation	Requirement	Messages
Associators	Unspecified	None
AssociatorNames	Unspecified	None
References	Unspecified	None
ReferenceNames	Unspecified	None

668 **9 Use Cases**

669 This section contains object diagrams and use cases for the *Service Processor Profile*.

670 **9.1 Object Diagrams**

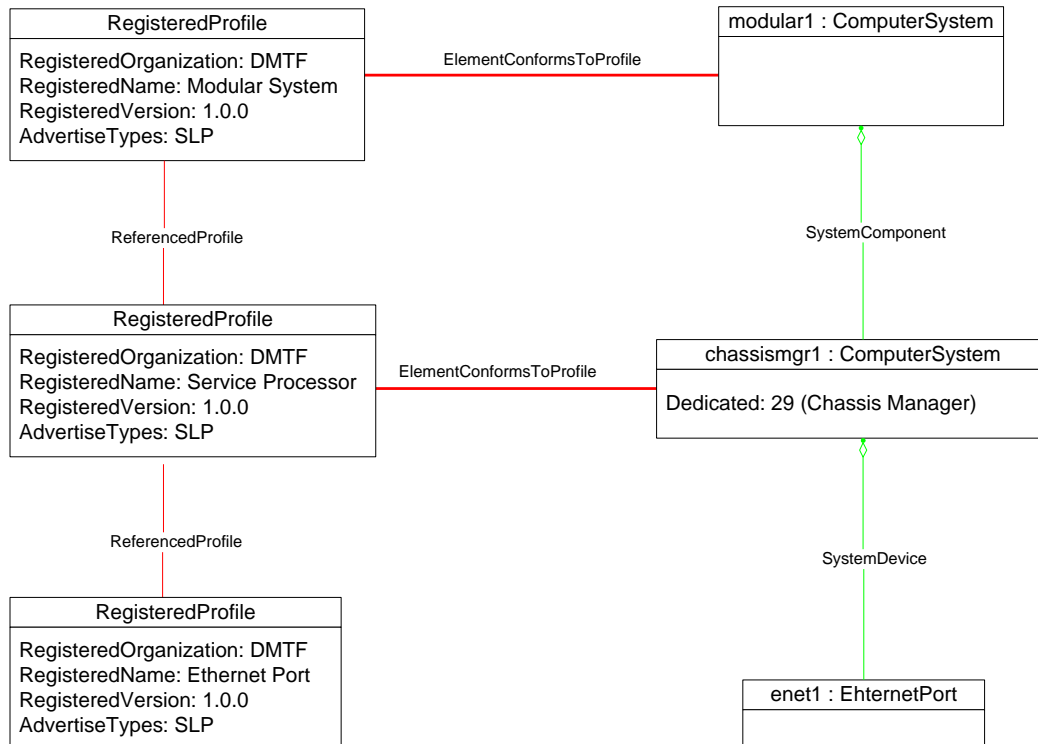
671 Figure 2 depicts an implementation of a service processor dedicated to a single computer system. Notice
 672 that the dedicated property of sp1 is 29 (Management Controller) and the managed computer system,
 673 system1 implements the [Base Server Profile](#). Figure 3 depicts an implementation of a Modular System
 674 with a chassis manager. Notice that the dedicated property of chassismgr1 is 29 (Chassis Manager) and
 675 that the manage system implements the [Modular System Profile](#).



676

677

Figure 2 – Base Server



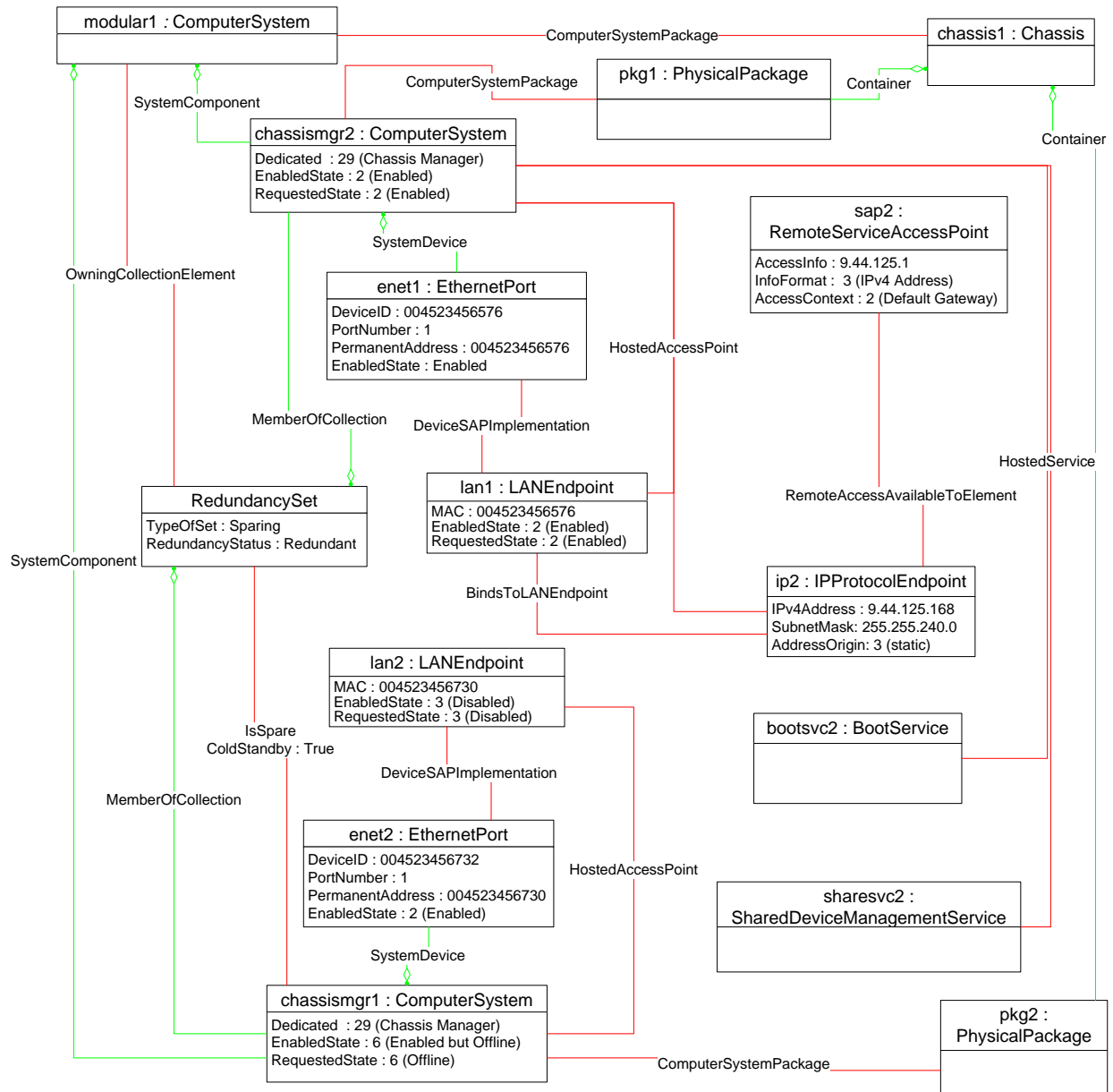
678

679

Figure 3 – Modular System

680 Figure 4 is an object diagram showing redundant service processors installed in a modular system.
 681 chassismgr1 is the active service processor. chassismgr2 is the backup service processor. This is
 682 indicated by the values of the EnabledState and RequestedState properties of the two instances and by
 683 the CIM_IsSpare association between the CIM_RedundancySet instance and chassismgr2.

684 In the illustrated system, a single configuration exists for the service processors. All functionality, including
 685 management interfaces, is hosted on and accessed at the active service processor. This is indicated by
 686 the active IP interface (ip1) bound to the Ethernet interface (enet2) of chassismgr1 and by the services
 687 (bootsvc1 and sharesvc1) associated through CIM_HostedService with chassismgr1.



699

700

Figure 5 – Service Processors after Failover

701 **9.2 Reset a Service Processor**

702 A client can reset the service processor as follows:

- 703 1) For the given instance of CIM_ComputerSystem, find the associated instance of
 704 CIM_EnabledLogicalElementCapabilities.
- 705 2) If the CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported property is a non-
 706 empty array that contains the value 11 (Reset), execute the RequestStateChange()
 707 with the value of the RequestedState parameter set to 11 (Reset).

708 The service processor represented by this instance will be disabled and then enabled.

709 9.3 Retrieve the Service Processor Redundancy Status

710 A client can determine the redundancy status for a given instance of CIM_ComputerSystem as follows:

- 711 1) Find the instance of CIM_RedundancySet that is associated with the instance of
712 CIM_ComputerSystem through an instance of CIM_MemberOfCollection.
- 713 2) Retrieve the value of the CIM_RedundancySet.RedundancyStatus property.

714 9.4 Determine Whether Manual Failover Is Supported

715 A client can determine whether a manual failover of the service processor is supported as follows:

- 716 1) Starting with an instance of CIM_ComputerSystem, find an instance of CIM_RedundancySet
717 that is associated with the CIM_ComputerSystem instance through the
718 CIM_MemberOfCollection association.
- 719 2) Find all instances of CIM_IsSpare that reference the CIM_RedundancySet instance. Query the
720 FailoverSupported property of each instance. If the FailoverSupported property of any instance
721 has the value of 3 (Manual) or 4 (Both Manual and Automatic), manual failover is supported.

722 9.5 Force a Service Processor Failover

723 A client can force a failover of the service processor as follows:

- 724 1) Starting with the CIM_ComputerSystem instance to failover from, find the instance of
725 CIM_RedundancySet that is associated with the CIM_ComputerSystem instance through the
726 CIM_MemberOfCollection association.
- 727 2) Find an instance of CIM_ComputerSystem associated with the CIM_RedundancySet instance
728 through the CIM_IsSpare association where the CIM_IsSpare.FailoverSupported property has
729 the value of 3 (Manual) or 4 (Both Manual and Automatic). This instance will be the service
730 processor to failover to.
- 731 3) Invoke the CIM_RedundancySet.Failover() method, specifying the CIM_ComputerSystem
732 instance from step 1) as the value for the FailoverFrom parameter and the
733 CIM_ComputerSystem instance from step 2) as the value for the FailoverTo parameter.

734 9.6 Determine Whether the ElementName Is Modifiable

735 A client can determine whether it can modify the CIM_ComputerSystem.ElementName property as
736 follows:

- 737 1) Find the CIM_EnabledLogicalElementCapabilities instance that is associated with the
738 CIM_ComputerSystem instance.
- 739 2) Query the value of the ElementNameEditSupported property of the
740 CIM_EnabledLogicalElementCapabilities instance. If the value is TRUE, the client can modify
741 the CIM_ComputerSystem.ElementName property.

742 9.7 Determining If State Management Is Supported

743 For a given instance of CIM_ComputerSystem, a client can determine whether state management of the
744 service processor is supported as follows:

- 745 1) Find the CIM_EnabledLogicalElementCapabilities instance that is associated with the
746 CIM_ComputerSystem instance.
- 747 2) Query the value of the RequestedStatesSupported property of the
748 CIM_EnabledLogicalElementCapabilities instance. If at least one value is specified, state
749 management is supported.

750 **10 CIM Elements**

751 Table 15 shows the instances of CIM Elements for this profile. Instances of the CIM Elements shall be
 752 implemented as described in Table 15. Sections 7 (“Implementation”) and 8 (“Methods”) may impose
 753 additional requirements on these elements.

754 **Table 15 – CIM Elements: Service Processor Profile**

Element Name	Requirement	Description
Classes		
CIM_ComputerSystem	Mandatory	See 10.1.
CIM_ElementCapabilities	Conditional	See 10.2.
CIM_EnabledLogicalElementCapabilities	Optional	See 10.3.
CIM_HostedService	Conditional	See 10.4.
CIM_IsSpare	Optional	See 10.5.
CIM_MemberOfCollection	Conditional	See 10.6.
CIM_OwningCollectionElement	Conditional	See 10.7.
CIM_RedundancySet	Optional	See 10.8.
CIM_RegisteredProfile	Mandatory	See 10.9.
CIM_ServiceAffectsElement	Optional	See 10.10.
CIM_TimeService	Optional	See 10.11.
CIM_ManagementController	Optional	See 10.12.
Indications		
None defined in this profile		

755 **10.1 CIM_ComputerSystem**

756 An instance of CIM_ComputerSystem represents each service processor installed in the enclosure.
 757 Table 16 contains the requirements for properties of the instance.

758 **Table 16 – Class: CIM_ComputerSystem**

Elements	Requirement	Notes
Dedicated	Mandatory	Matches 28 (Management Controller) when the service processor is dedicated to a single base system or 29 (Chassis Manager) when the service processor is dedicated to a Modular System.
Name	Mandatory	None
CreationClassName	Mandatory	None
OtherIdentifyingInfo	Optional	This property should be implemented.
IdentifyingDescriptions	Optional	This property should be implemented.
EnabledState	Mandatory	See 7.1.1.
RequestedState	Mandatory	See 7.1.2.2 and 7.1.3.2.
OperationalStatus	Mandatory	None
HealthState	Mandatory	None
ElementName	Mandatory	See 7.1.4 and 7.1.5.
RequestStateChange()	Conditional	See 7.1.2 and 8.1.

Elements	Requirement	Notes

759 10.2 CIM_ElementCapabilities

760 CIM_ElementCapabilities associates an instance of CIM_EnabledLogicalElementCapabilities with an
761 instance of CIM_ComputerSystem. Table 17 contains the requirements for properties of the instance.

762 **Table 17 – Class: CIM_ElementCapabilities**

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

763 10.3 CIM_EnabledLogicalElementCapabilities

764 CIM_EnabledLogicalElementCapabilities indicates support for managing the state of the service
765 processor. Table 18 contains the requirements for properties of the instance.

766 **Table 18 – Class: CIM_EnabledLogicalElementCapabilities**

Elements	Requirement	Notes
InstanceID	Mandatory	None
RequestedStatesSupported	Mandatory	See 7.1.2.1.1 and 7.1.3.1.1.
ElementNameEditSupported	Mandatory	See 7.1.4.1.1 and 7.1.5.1.1.
MaxElementNameLen	Conditional	See 7.1.4.1.2 and 7.1.5.1.2.

767 10.4 CIM_HostedService

768 CIM_HostedService relates the CIM_TimeService instance to its scoping CIM_ComputerSystem
769 instance. Table 19 contains the requirements for properties of the instance.

770 **Table 19 – Class: CIM_HostedService**

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

771 **10.5 CIM_IsSpare**

772 CIM_IsSpare associates an instance of CIM_ComputerSystem with the CIM_RedundancySet for which
 773 the CIM_ComputerSystem instance represents a spare service processor. Table 20 contains the
 774 requirements for properties of the instance.

775 **Table 20 – Class: CIM_IsSpare**

Elements	Requirement	Description
Antecedent	Mandatory	Reference to the CIM_RedundancySet instance of which the current CIM_ComputerSystem instance is a member and where the CIM_ComputerSystem instance is a spare
Dependent	Mandatory	Reference to the current CIM_ComputerSystem instance
SpareStatus	Optional	See 7.2.4.

776 **10.6 CIM_MemberOfCollection**

777 CIM_MemberOfCollection associates an instance of CIM_ComputerSystem that represents a service
 778 processor with the CIM_RedundancySet of which the CIM_ComputerSystem is a member. Table 21
 779 contains the requirements for properties of the instance.

780 **Table 21 – Class: CIM_MemberOfCollection**

Elements	Requirement	Description
Collection	Mandatory	See 7.2.1. Cardinality 0..1
Member	Mandatory	See 7.2.1. Cardinality *

781 **10.7 CIM_OwningCollectionElement**

782 CIM_OwningCollectionElement associates the CIM_RedundancySet instance with the
 783 CIM_ComputerSystem instance of which the CIM_RedundancySet instance is a member. The instance of
 784 CIM_OwningCollectionElement is conditional on having instantiation of the CIM_RedundancySet class.
 785 Table 22 contains the requirements for properties of the instance.

786 **Table 22 – Class: CIM_OwningCollectionElement**

Elements	Requirement	Notes
OwningElement	Mandatory	See 7.2.2. Cardinality 0..1
OwnedElement	Mandatory	See 7.2.2. Cardinality *

787 **10.8 CIM_RedundancySet**

788 CIM_RedundancySet represents a collection of CIM_ComputerSystem instances that operate as
789 redundant service processors. Table 23 contains the requirements for properties of the instance.

790 **Table 23 – Class: CIM_RedundancySet**

Elements	Requirement	Notes
InstanceID	Mandatory	None
RedundancyStatus	Mandatory	None
TypeOfSet	Mandatory	See 7.2.
MinNumberNeeded	Mandatory	This property shall match 0 when the minimum number of service processors needed for the redundancy is unknown.
ElementName	Mandatory	This property shall be formatted as a free-form string of variable length (pattern ".*").
Failover()	Optional	See 8.2.

791 **10.9 CIM_RegisteredProfile**

792 CIM_RegisteredProfile identifies the *Service Processor Profile* in order for a client to determine whether
793 an instance of CIM_ComputerSystem is conformant with this profile. The CIM_RegisteredProfile class is
794 defined by the *Profile Registration Profile*. With the exception of the mandatory values specified for the
795 properties in Table 24, the behavior of the CIM_RegisteredProfile instance is in accordance with the
796 *Profile Registration Profile*.

797 **Table 24 – Class: CIM_RegisteredProfile**

Elements	Requirement	Notes
RegisteredName	Mandatory	This property shall have a value of "Service Processor".
RegisteredVersion	Mandatory	This property shall have a value of "1.1.1".

798 **10.10 CIM_ServiceAffectsElement**

799 CIM_ServiceAffectsElement associates the CIM_TimeService instance with the Central Instance.
800 Table 25 contains the requirements for properties of the instance.

801 **Table 25 – Class: CIM_ServiceAffectsElement**

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

802 **10.11 CIM_TimeService**

803 CIM_TimeService manages the current time on the service processor. Table 26 contains the
 804 requirements for properties of the instance.

805 **Table 26 – Class: CIM_TimeService**

Elements	Requirement	Notes
SystemCreationClassName	Mandatory	Key
SystemName	Mandatory	Key
CreationClassName	Mandatory	Key
Name	Mandatory	Key
ElementName	Mandatory	Pattern (".*")

806 **10.12 CIM_ManagementController**

807 CIM_ManagementController is a model construct used by an implementation to support linking the service
 808 processor to other constructions for managing the settings of the service processor, such as PCI or
 809 register information. Table 27 contains the requirements for properties of the instance.

810 **Table 27 – Class: CIM_ManagementController**

Elements	Requirement	Notes
SystemCreationClassName	Mandatory	Key
SystemName	Mandatory	Key
CreationClassName	Mandatory	Key
DeviceID	Mandatory	Key

811
 812
 813

814
815
816
817

ANNEX A
(informative)
Change Log

Version	Date	Description
1.0.0	2009-06-22	
1.1.0	2011-06-30	Added support for the PCI Profile
1.1.1	2013-03-28	Removed HostedAccessPoint for RemoteServiceAccessPoint in Fig 4 and Fig 5 Renamed one of enet2 as enet1 in Fig 4

818