



1

2

3

4

Document Identifier: DSP1018

Date: 2019-03-15

Version: 1.1.2

5 **Service Processor Profile**

6 **Supersedes: 1.1.1**

7 **Document Class: Normative**

8 **Document Status: Published**

9 **Document Language: en-US**

10

11 Copyright Notice

12 Copyright © 2009, 2011, 2012, 2019 DMTF. All rights 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27 withdrawn or modified after publication, and shall be indemnified and held harmless by any party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>.

33 This document's normative language is English. Translation into other languages is permitted.

34

CONTENTS

36	Foreword	4
37	Introduction.....	7
38	1 Scope	9
39	2 Normative references	9
40	4 Symbols and abbreviated terms.....	11
41	5 Synopsis	11
42	6 Description	12
43	7 Implementation.....	13
44	7.1 Representing a service processor	13
45	7.2 Modeling service processor redundancy (optional)	16
46	7.3 Managing service processor time (optional)	17
47	7.4 User account management (optional).....	17
48	7.5 Boot Control Profile (optional).....	17
49	7.6 CLP Service Profile (optional).....	17
50	7.7 DHCP Client Profile (optional)	17
51	7.8 DNS Client Profile (optional).....	17
52	7.9 Ethernet Port Profile (optional)	17
53	7.10 Software Inventory Profile (optional).....	17
54	7.11 Software Update Profile (optional).....	18
55	7.12 IP Interface Profile (optional)	18
56	7.13 Physical Asset Profile (optional)	18
57	7.14 Record Log Profile (optional)	18
58	7.15 Sensors Profile (optional)	18
59	7.16 Power State Management Profile (optional).....	18
60	7.17 Shared Device Management Profile (optional).....	18
61	7.18 SMASH Collections Profile (optional)	18
62	7.19 SSH Service Profile (optional)	18
63	7.20 Telnet Service Profile (optional).....	19
64	7.21 Text Console Redirection Profile (optional)	19
65	7.22 PCI Device Profile (optional).....	19
66	8 Methods.....	19
67	8.1 Method: CIM_ComputerSystem.RequestStateChange()	19
68	8.2 Method: CIM_RedundancySet.Failover()	20
69	8.3 Method: CIM_TimeService.ManageTime()	21
70	8.4 Profile conventions for operations	22
71	8.5 CIM_ComputerSystem.....	22
72	8.6 CIM_HostedService	23
73	8.7 CIM_IsSpare	23
74	8.8 CIM_ElementCapabilities	23
75	8.9 CIM_EnabledLogicalElementCapabilities.....	24
76	8.10 CIM_MemberOfCollection	24
77	8.11 CIM_RedundancySet.....	24
78	8.12 CIM_TimeService	24
79	8.13 CIM_ServiceAffectsElement	24
80	9 Use cases.....	25
81	9.1 Object diagrams.....	25
82	9.2 Reset a service processor	28
83	9.3 Retrieve the service processor redundancy status.....	29
84	9.4 Determine whether manual failover is supported	29
85	9.5 Force a service processor failover	29
86	9.6 Determine whether the elementname is modifiable	29
87	9.7 Determining whether state management is supported.....	29

88 10 CIM Elements 30
 89 10.1 CIM_ComputerSystem 30
 90 10.2 CIM_ElementCapabilities 31
 91 10.3 CIM_EnabledLogicalElementCapabilities 31
 92 10.4 CIM_HostedService 31
 93 10.5 CIM_IsSpare 32
 94 10.6 CIM_MemberOfCollection 32
 95 10.7 CIM_OwningCollectionElement 32
 96 10.8 CIM_RedundancySet 33
 97 10.9 CIM_RegisteredProfile 33
 98 10.10 CIM_ServiceAffectsElement 33
 99 10.11 CIM_TimeService 34
 100 10.12 CIM_ManagementController 34
 101 ANNEX A (informative) Change log 35
 102

103 **Figures**

104 Figure 1 – Service Processor Profile: Class diagram 13
 105 Figure 2 – Base server 25
 106 Figure 3 – Modular system 26
 107 Figure 4 – Service processors before failover 27
 108 Figure 5 – Service processors after failover 28
 109

110 **Tables**

111 Table 1 – Referenced profiles 12
 112 Table 2 – CIM_ComputerSystem.EnabledState Value description 14
 113 Table 3 – CIM_ComputerSystem.RequestStateChange() method: Return code values 19
 114 Table 4 – CIM_ComputerSystem.RequestStateChange() method: Parameters 20
 115 Table 5 – CIM_RedundancySet.Failover() method: Return code values 21
 116 Table 6 – CIM_RedundancySet.Failover() method: Parameters 21
 117 Table 7 – CIM_TimeService.ManageTime() method: Return code values 21
 118 Table 8 – CIM_TimeService.ManageTime() method: Parameters 21
 119 Table 9 – Operations: CIM_ComputerSystem 22
 120 Table 10 – Operations: CIM_HostedService 23
 121 Table 11 – Operations: CIM_IsSpare 23
 122 Table 12 – Operations: CIM_ElementCapabilities 23
 123 Table 13 – Operations: CIM_MemberOfCollection 24
 124 Table 14 – Operations: CIM_ServiceAffectsElement 24
 125 Table 15 – CIM Elements: Service Processor Profile 30
 126 Table 16 – Class: CIM_ComputerSystem 30
 127 Table 17 – Class: CIM_ElementCapabilities 31
 128 Table 18 – Class: CIM_EnabledLogicalElementCapabilities 31
 129 Table 19 – Class: CIM_HostedService 31
 130 Table 20 – Class: CIM_IsSpare 32
 131 Table 21 – Class: CIM_MemberOfCollection 32
 132 Table 22 – Class: CIM_OwningCollectionElement 32
 133 Table 23 – Class: CIM_RedundancySet 33

134 Table 24 – Class: CIM_RegisteredProfile 33
135 Table 25 – Class: CIM_ServiceAffectsElement 33
136 Table 26 – Class: CIM_TimeService 34
137 Table 27 – Class: CIM_ManagementController 34

138

Foreword

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

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

143 Acknowledgments

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

145 Editor:

- 146 • Jeff Hilland – Hewlett Packard Enterprise
- 147 • Aaron Merkin – IBM

148 Contributors:

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

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.
168

170

Service Processor Profile

171 1 Scope

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

173 2 Normative references

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

178 DMTF DSP0004, *CIM Infrastructure Specification 2.5*,
179 https://www.dmtf.org/sites/default/files/standards/documents/DSP0004_2.5.pdf

180 DMTF DSP0200, *CIM Operations over HTTP 1.2*,
181 <https://www.dmtf.org/sites/default/files/standards/documents/DSP200.html>

182 DMTF DSP1001, *Management Profile Specification Usage Guide 1.0*,
183 https://www.dmtf.org/sites/default/files/standards/documents/DSP1001_1.0.pdf

184 DMTF DSP1004, *Base Server Profile 1.0*,
185 https://www.dmtf.org/sites/default/files/standards/documents/DSP1004_1.0.pdf

186 DMTF DSP1005, *CLP Service Profile 1.0*,
187 https://www.dmtf.org/sites/default/files/standards/documents/DSP1005_1.0.pdf

188 DMTF DSP1006, *SMASH Collections Profile 1.0*,
189 https://www.dmtf.org/sites/default/files/standards/documents/DSP1006_1.0.pdf

190 DMTF DSP1008, *Modular System Profile 1.0*,
191 https://www.dmtf.org/sites/default/files/standards/documents/DSP1008_1.0.pdf

192 DMTF DSP1009, *Sensors Profile 1.0*,
193 https://www.dmtf.org/sites/default/files/standards/documents/DSP1009_1.0.pdf

194 DMTF DSP1010, *Record Log Profile 1.0*,
195 https://www.dmtf.org/sites/default/files/standards/documents/DSP1010_1.0.pdf

196 DMTF DSP1011, *Physical Asset Profile 1.0*,
197 https://www.dmtf.org/sites/default/files/standards/documents/DSP1011_1.0.pdf

198 DMTF DSP1012, *Boot Control Profile 1.0*,
199 https://www.dmtf.org/sites/default/files/standards/documents/DSP1012_1.0.pdf

200 DMTF DSP1014, *Ethernet Port Profile 1.0*,
201 https://www.dmtf.org/sites/default/files/standards/documents/DSP1014_1.0.pdf

202 DMTF DSP1016, *Telnet Service Profile 1.0*,
203 http://www.dmtf.org/sites/default/files/standards/documents/DSP1016_1.0.pdf

204 DMTF DSP1017, *SSH Service Profile 1.0*,
205 http://www.dmtf.org/sites/default/files/standards/documents/DSP1017_1.0.pdf

- 206 DMTF DSP1021, *Shared Device Management Profile 1.0*,
207 http://www.dmtf.org/sites/default/files/standards/documents/DSP1021_1.0.pdf
- 208 DMTF DSP1023, *Software Inventory Profile 1.0*,
209 http://www.dmtf.org/sites/default/files/standards/documents/DSP1023_1.0.pdf
- 210 DMTF DSP1024, *Text Console Redirection Profile 1.0*,
211 http://www.dmtf.org/sites/default/files/standards/documents/DSP1024_1.0.pdf
- 212 DMTF DSP1025, *Software Update Profile 1.0*,
213 http://www.dmtf.org/sites/default/files/standards/documents/DSP1025_1.0.pdf
- 214 DMTF DSP1027, *Power State Management Profile 1.0*,
215 http://www.dmtf.org/sites/default/files/standards/documents/DSP1027_1.0.pdf
- 216 DMTF DSP1033, *Profile Registration Profile 1.0*,
217 http://www.dmtf.org/sites/default/files/standards/documents/DSP1033_1.0.pdf
- 218 DMTF DSP1034, *Simple Identity Management Profile 1.0*,
219 http://www.dmtf.org/sites/default/files/standards/documents/DSP1034_1.0.pdf
- 220 DMTF DSP1036, *IP Interface Profile 1.0*,
221 http://www.dmtf.org/sites/default/files/standards/documents/DSP1036_1.0.pdf
- 222 DMTF DSP1037, *DHCP Client Profile 1.0*,
223 http://www.dmtf.org/sites/default/files/standards/documents/DSP1037_1.0.pdf
- 224 DMTF DSP1038, *DNS Client Profile 1.0*,
225 http://www.dmtf.org/sites/default/files/standards/documents/DSP1038_1.0.pdf
- 226 DMTF DSP1039, *Role Based Authorization Profile 1.0*,
227 http://www.dmtf.org/sites/default/files/standards/documents/DSP1039_1.0.pdf
- 228 DMTF DSP1075, *PCI Device Profile 1.0*,
229 http://www.dmtf.org/sites/default/files/standards/documents/DSP1075_1.0.pdf
- 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 In this document, some terms have a specific meaning beyond the normal English meaning. Those terms
234 are defined in this clause.

235 The terms "shall" ("required"), "shall not", "should" ("recommended"), "should not" ("not recommended"),
236 "may", "need not" ("not required"), "can" and "cannot" in this document are to be interpreted as described
237 in [ISO/IEC Directives, Part 2](#), Clause 7. The terms in parentheses are alternatives for the preceding term,
238 for use in exceptional cases when the preceding term cannot be used for linguistic reasons. Note that
239 [ISO/IEC Directives, Part 2](#), Clause 7 specifies additional alternatives. Occurrences of such additional
240 alternatives shall be interpreted in their normal English meaning.

241 The terms "clause", "subclause", "paragraph", and "annex" in this document are to be interpreted as
242 described in [ISO/IEC Directives, Part 2](#), Clause 6.

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

246 The terms defined in [DSP0004](#), [DSP0223](#), and [DSP1001](#) apply to this document. The following additional
247 terms are used in this document.

- 248 **3.1**
249 **conditional**
250 indicates requirements to be followed strictly to conform to the document when the specified conditions
251 are met
- 252 **3.2**
253 **mandatory**
254 indicates requirements to be followed strictly to conform to the document and from which no deviation is
255 permitted
- 256 **3.3**
257 **optional**
258 indicates a course of action permissible within the limits of the document
- 259 **3.4**
260 **referencing profile**
261 indicates a profile that owns the definition of this class and can include a reference to this profile in its
262 "Referenced Profiles" table
- 263 **3.5**
264 **unspecified**
265 indicates that this profile does not define any constraints for the referenced CIM element or operation
- 266 **3.6**
267 **service processor**
268 a specialized device dedicated to management
- 269 **3.7**
270 **standby service processor**
271 an instance of CIM_ComputerSystem that represents a standby service processor of a redundancy set

272 **4 Symbols and abbreviated terms**

273 None.

274 **5 Synopsis**

275 **Profile Name:** Service Processor

276 **Version:** 1.1.1

277 **Organization:** DMTF

278 **CIM Schema Version:** 2.20

279 **Central Class:** CIM_ComputerSystem

280 **Scoping Class:** CIM_ComputerSystem

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

282

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.

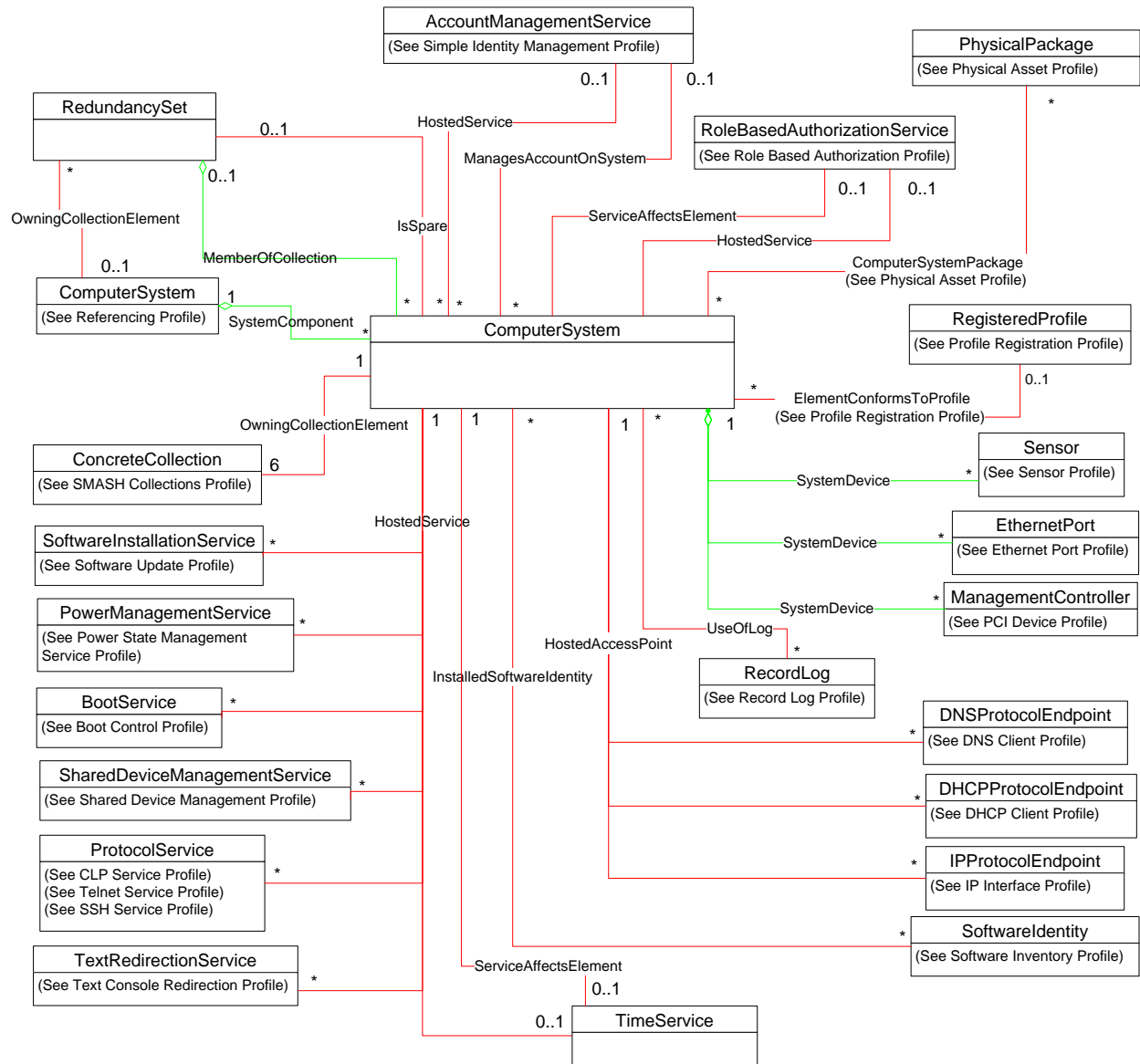
283 6 Description

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

288 Some examples of the service processors are:

- 289 • management processor (MP)
- 290 • service processor (SP)
- 291 • baseboard management controller (BMC)
- 292 • chassis manager

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



295

296

Figure 1 – Service Processor Profile: Class diagram

297 7 Implementation

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

301 7.1 Representing a service processor

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

303 7.1.1 CIM_ComputerSystem.EnabledState

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

310 **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).

311 7.1.2 Service processor state management is supported — conditional

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

314 7.1.2.1 CIM_EnabledLogicalElementCapabilities

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

318 7.1.2.1.1 CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported

319 The RequestedStatesSupported property may contain zero or more of the following values: 2 (Enabled),
 320 3 (Disabled), 6 (Offline), or 11 (Reset).

321 7.1.2.2 CIM_ComputerSystem.RequestedState

322 When the CIM_ComputerSystem.RequestStateChange() method is successfully invoked, the value of the
 323 RequestedState property shall be the value of the RequestedState parameter. If the method is not
 324 successfully invoked, the value of the RequestedState property is indeterminate.

325 The CIM_ComputerSystem.RequestedState property shall have one of the values specified in the
 326 CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported property or a value of 5 (No
 327 Change).

328 7.1.2.3 CIM_ComputerSystem.EnabledState

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

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

333 7.1.3 Service processor state management is not supported

334 This clause describes the CIM elements and behaviors that shall be implemented when management of
 335 the service processor state is not supported.

336 7.1.3.1 CIM_EnabledLogicalElementCapabilities

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

340 7.1.3.1.1 CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported

341 The CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported property shall not contain any
342 values.

343 7.1.3.2 CIM_ComputerSystem.RequestedState

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

345 7.1.4 Modifying ElementName is supported — conditional

346 The CIM_ComputerSystem.ElementName property may support being modified by the ModifyInstance
347 operation. See 8.5.1. This behavior is conditional. This clause describes the CIM elements and behavior
348 requirements when an implementation supports client modification of the
349 CIM_ComputerSystem.ElementName property.

350 7.1.4.1 CIM_EnabledLogicalElementCapabilities

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

353 7.1.4.1.1 CIM_EnabledLogicalElementCapabilities.ElementNameEditSupported

354 The ElementNameEditSupported property shall have a value of TRUE.

355 7.1.4.1.2 CIM_EnabledLogicalElement.MaxElementNameLen

356 The MaxElementNameLen property shall be implemented.

357 7.1.5 Modifying ElementName is not supported

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

360 7.1.5.1 CIM_EnabledLogicalElementCapabilities

361 An instance of CIM_EnabledLogicalElementCapabilities may be associated with the
362 CIM_ComputerSystem instance through an instance of CIM_ElementCapabilities.

363 7.1.5.1.1 CIM_EnabledLogicalElementCapabilities.ElementNameEditSupported

364 The ElementNameEditSupported shall have a value of FALSE.

365 7.1.5.1.2 CIM_EnabledLogicalElement.MaxElementNameLen

366 The MaxElementNameLen property may be implemented. The MaxElementNameLen property is
367 irrelevant in this context.

368 7.1.6 Representing the physical packaging (optional)

369 Support for representing the physical packaging of the service processor is optional. The physical
370 packaging may be modeled using one or more instances of CIM_PhysicalElement in accordance with the
371 [DSP1011](#).

372 7.2 Modeling service processor redundancy (optional)

373 Modeling of service processor redundancy is optional. When service processor redundancy is supported,
374 the requirements in this clause apply.

375 At least one instance of CIM_RedundancySet shall exist.

376 7.2.1 Relationship between redundancy set and redundant service processors

377 Each CIM_ComputerSystem instance that represents a service processor participating in the redundancy
378 shall be associated with the CIM_RedundancySet instance through the CIM_MemberOfCollection
379 association. Each instance of CIM_ComputerSystem that is associated with the CIM_RedundancySet
380 instance through the CIM_MemberOfCollection association shall be associated with the same instance of
381 CIM_ComputerSystem through the CIM_SystemComponent association where the value of the
382 CIM_SystemComponent.PartComponent property is the instance of CIM_ComputerSystem that is
383 associated with the CIM_RedundancySet.

384 7.2.2 Relationship between redundancy set and containing system

385 When the CIM_ComputerSystem instance that represents a service processor is associated with another
386 CIM_ComputerSystem instance through the CIM_SystemComponent association where the value of the
387 CIM_SystemComponentPartComponent property is the CIM_ComputerSystem instance that represents
388 the service processor, the CIM_RedundancySet instance shall be associated with the
389 CIM_ComputerSystem instance that is the value of the CIM_SystemComponent.GroupComponent
390 property through the CIM_OwningCollectionElement association.

391 7.2.3 Active/active redundancy

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

- 395 • The CIM_ComputerSystem instances shall not be associated with the CIM_RedundancySet
396 instance through the CIM_IsSpare association.
- 397 • For each instance of CIM_ComputerSystem, the CIM_ComputerSystem.EnabledState property
398 shall not have the value 6 (Enabled but Offline).

399 7.2.4 Active/standby redundancy

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

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

- 404 • When the CIM_ComputerSystem.EnabledState property has the value 6 (Enabled but Offline),
405 the SpareStatus property of the referencing CIM_IsSpare instance shall have the value 2 (Hot
406 Standby).
- 407 • When the CIM_ComputerSystem.EnabledState property has the value 3 (Disabled), the
408 SpareStatus property of the referencing CIM_IsSpare instance shall have the value 3 (Cold
409 Standby).
- 410 • When the CIM_ComputerSystem.EnabledState property has a value other than 3 (Disabled) or
411 6 (Enabled but Offline), the SpareStatus property of the referencing CIM_IsSpare instance shall
412 have the value 0 (Unknown).

413 **7.3 Managing service processor time (optional)**

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

418 An instance of CIM_TimeService shall be associated with the Central Instance through the
419 CIM_HostedService association. The instance of CIM_TimeService shall also be associated with the
420 Central Instance through the CIM_ServiceAffectsElement association.

421 **7.4 User account management (optional)**

422 [DSP1034](#) and the [DSP1039](#) may be implemented to model user access to the service processor. When
423 [DSP1034](#) is implemented, an instance of CIM_AccountManagementService shall be associated with the
424 Central Instance through the CIM_HostedService association. When [DSP1039](#) is implemented, an
425 instance of CIM_RoleBasedAuthorizationService shall be associated with the Central Instance through
426 the CIM_HostedService association.

427 **7.5 Boot Control Profile (optional)**

428 [DSP1012](#) may be implemented to model the ability of the service processor to manage its own boot
429 configuration or that of the systems it managed. If [DSP1012](#) is implemented, an instance of
430 CIM_BootService shall be associated with the Central Instance through the CIM_HostedService
431 association.

432 **7.6 CLP Service Profile (optional)**

433 [DSP1005](#) may be implemented to model a CLP service hosted on the service processor. When [DSP1005](#)
434 is implemented, at least one instance of CIM_ProtocolService shall be associated with the Central
435 Instance through an instance of CIM_HostedService.

436 **7.7 DHCP Client Profile (optional)**

437 [DSP1037](#) may be implemented to model the DHCP client of a service processor. When [DSP1037](#) is
438 implemented, at least one instance of CIM_DHCPProtocolEndpoint shall be associated with the Central
439 Instance through an instance of CIM_HostedAccessPoint.

440 **7.8 DNS Client Profile (optional)**

441 [DSP1038](#) may be implemented to model the DNS client of a service processor. When [DSP1038](#) is
442 implemented, at least one instance of CIM_DNSProtocolEndpoint shall be associated with the Central
443 Instance through an instance of CIM_HostedAccessPoint.

444 **7.9 Ethernet Port Profile (optional)**

445 [DSP1014](#) may be implemented to model an Ethernet interface of a service processor. When [DSP1014](#) is
446 implemented, at least one instance of CIM_EthernetPort shall be associated with the Central Instance
447 through an instance of CIM_SystemDevice.

448 **7.10 Software Inventory Profile (optional)**

449 [DSP1023](#) may be implemented to model the software version information of the service processor. When
450 [DSP1023](#) is implemented, at least one instance of CIM_SoftwareIdentity shall be associated with the
451 Central Instance of this profile through an instance of CIM_InstalledSoftwareIdentity.

452 7.11 Software Update Profile (optional)

453 [DSP1025](#) may be implemented to model the ability of the service processor to update software installed
454 on one or more components of managed systems, including the service processor itself. When [DSP1025](#)
455 is implemented, an instance of CIM_SoftwareInstallationService shall be associated with the Central
456 Instance through an instance of CIM_HostedService.

457 7.12 IP Interface Profile (optional)

458 [DSP1036](#) may be implemented to model the IP interface of a service processor. When [DSP1036](#) is
459 implemented, at least one instance of CIM_IPProtocolEndpoint shall be associated with the Central
460 Instance through an instance of CIM_HostedAccessPoint.

461 7.13 Physical Asset Profile (optional)

462 [DSP1011](#) may be implemented to model the physical package and physical asset information of a service
463 processor. When [DSP1011](#) is implemented, at least one instance of CIM_PhysicalPackage shall be
464 associated with the Central Instance through an instance of CIM_ComputerSystemPackage.

465 7.14 Record Log Profile (optional)

466 [DSP1010](#) may be implemented to model one or more logs of the service processor. When [DSP1010](#) is
467 implemented, an instance of CIM_RecordLog shall be associated with Central Instance through an
468 instance of CIM_UseOfLog.

469 7.15 Sensors Profile (optional)

470 [DSP1009](#) may be implemented to model the sensors of the service processor. When [DSP1009](#) is
471 implemented, at least one instance of CIM_Sensor or CIM_NumericSensor shall be associated with the
472 Central Instance through an instance of CIM_SystemDevice.

473 7.16 Power State Management Profile (optional)

474 [DSP1027](#) may be implemented to model the ability of the service processor to perform power control
475 operations for the managed system or the service processor itself. When [DSP1027](#) is implemented, an
476 instance of CIM_PowerManagementService shall be associated with the Central Instance through an
477 instance of CIM_HostedService.

478 7.17 Shared Device Management Profile (optional)

479 [DSP1021](#) may be implemented to model the ability of the service processor to control shared devices of a
480 modular system. When [DSP1021](#) is implemented, an instance of CIM_SharedDeviceManagementService
481 shall be associated with the Central Instance through an instance of CIM_HostedService.

482 7.18 SMASH Collections Profile (optional)

483 [DSP1006](#) may be implemented. When [DSP1006](#) is implemented, each instance of
484 CIM_ConcreteCollection that is defined by [DSP1006](#) shall be associated with the Central Instance
485 through an instance of CIM_OwningCollectionElement.

486 7.19 SSH Service Profile (optional)

487 [DSP1017](#) may be implemented to model an SSH service hosted on the service processor. When
488 [DSP1017](#) is implemented, at least one instance of CIM_ProtocolService shall be associated with the
489 Central Instance through an instance of CIM_HostedService where the CIM_ProtocolService.Protocol
490 property has the value 2 (SSH).

491 **7.20 Telnet Service Profile (optional)**

492 [DSP1016](#) may be implemented to model a Telnet service hosted on the service processor. When
 493 [DSP1016](#) is implemented, at one instance of CIM_ProtocolService shall be associated with the Central
 494 Instance through an instance of CIM_HostedService where the CIM_ProtocolService.Protocol property
 495 has the value 3 (Telnet).

496 **7.21 Text Console Redirection Profile (optional)**

497 [DSP1024](#) may be implemented to model the ability of the service processor to provide text console
 498 redirection for managed systems. When [DSP1024](#) is implemented, at least one instance of
 499 CIM_TextRedirectionService shall be associated with the Central Instance through an instance of
 500 CIM_HostedService.

501 **7.22 PCI Device Profile (optional)**

502 [DSP1075](#) may be implemented to model the ability of the service processor to provide PCI configuration
 503 information for managed systems. When [DSP1075](#) is implemented and the ServiceProcessor is modeled
 504 as a PCI device, at least one instance of CIM_ManagementController shall be associated with the Central
 505 Instance of this profile through an instance of CIM_SystemDevice and the CIM_ManagementController
 506 shall be associated with at least one instance of CIM_PCIDevice through an instance of
 507 CIM_ConcretelDentity.

508 **8 Methods**

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

511 **8.1 Method: CIM_ComputerSystem.RequestStateChange()**

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

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

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

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

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

523 No standard messages are defined for this method.

524 **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.

525

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

526 8.1.1 RequestStateChange() for the standby service processor

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

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

533 8.2 Method: CIM_RedundancySet.Failover()

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

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

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

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

- 546 • The CIM_ComputerSystem instance that is referenced by the FailoverTo parameter is not a
547 standby service processor.
- 548 • The CIM_ComputerSystem instance that is referenced by the FailoverFrom parameter is not
549 associated with the CIM_RedundancySet instance only through the CIM_MemberOfCollection
550 association.

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

- 552 • The CIM_ComputerSystem that is referenced by the FailoverTo parameter shall take over as the
553 active service processor.

- 554 • The CIM_ComputerSystem instance that is referenced by the FailoverTo parameter shall be
555 associated with the CIM_RedundancySet instance only through the CIM_MemberOfCollection
556 association.
- 557 • The CIM_ComputerSystem instance that is referenced by the FailoverFrom parameter shall
558 become a standby service processor. This instance will conform to the requirements for a
559 standby service processor specified in 7.2.4.
- 560 • When management of the service processor state is supported, the CIM_ComputerSystem
561 instance that is referenced by the FailoverFrom parameter shall not have an EnabledState
562 property value of 2 (Enabled) but may have a value of 6 (Enabled but Offline).

563 Return code values for the CIM_RedundancySet.Failover() method are specified in Table 5. Parameters
564 for the CIM_RedundancySet.Failover() method are specified in Table 6. No standard messages are
565 defined for this method.

566 **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.

567 **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

568 **8.3 Method: CIM_TimeService.ManageTime()**

569 The CIM_TimeService.ManageTime() method is used to query or modify the service processor time.
570 When the GetRequest parameter has a value of TRUE, the TimeData parameter shall be ignored. If the
571 GetRequest parameter is not specified, the method shall return a value of 2 (Error Occurred). When the
572 ManagedElement parameter is not a reference to the Central Instance, the method shall return a value of
573 2 (Error Occurred).

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

576 **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.

577 **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

Qualifiers	Name	Type	Description/Values
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_ManagedElement	Reference to the Central Instance

578 8.4 Profile conventions for operations

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

581 The default list of operations is as follows:

- 582 • GetInstance
- 583 • Associators
- 584 • AssociatorNames
- 585 • References
- 586 • ReferenceNames
- 587 • EnumerateInstances
- 588 • EnumerateInstanceNames

589 8.5 CIM_ComputerSystem

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

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

594 **Table 9 – Operations: CIM_ComputerSystem**

Operation	Requirement	Messages
ModifyInstance	Optional. See 8.5.1.	None

595 8.5.1 CIM_ComputerSystem — ModifyInstance

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

598 The ModifyInstance operation shall be supported and the CIM_ComputerSystem.ElementName property
599 shall be modifiable when the ElementNameEditSupported property of the
600 CIM_EnabledLogicalElementCapabilities instance that is associated with the CIM_ComputerSystem
601 instance has a value of TRUE. See 8.5.1.1.

602 8.5.1.1 CIM_ComputerSystem.ElementName

603 When the ElementNameEditSupported property of the CIM_EnabledLogicalElementCapabilities instance
604 that is associated with the CIM_ComputerSystem instance has a value of TRUE, the implementation shall
605 allow the ModifyInstance operation to change the value of the ElementName property of the
606 CIM_ComputerSystem instance. The ModifyInstance operation shall enforce the length restriction
607 specified in the MaxElementNameLen property of the CIM_EnabledLogicalElementCapabilities instance.

608 When the ElementNameEditSupported property of the CIM_EnabledLogicalElementCapabilities instance
 609 has a value of FALSE, the implementation shall not allow the ModifyInstance operation to change the
 610 value of the ElementName property of the CIM_ComputerSystem instance.

611 **8.6 CIM_HostedService**

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

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

616 **Table 10 – Operations: CIM_HostedService**

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

617 **8.7 CIM_IsSpare**

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

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

622 **Table 11 – Operations: CIM_IsSpare**

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

623 **8.8 CIM_ElementCapabilities**

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

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

628 **Table 12 – Operations: CIM_ElementCapabilities**

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

629 8.9 CIM_EnabledLogicalElementCapabilities

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

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

632 8.10 CIM_MemberOfCollection

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

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

637 **Table 13 – Operations: CIM_MemberOfCollection**

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

638 8.11 CIM_RedundancySet

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

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

641 8.12 CIM_TimeService

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

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

644 8.13 CIM_ServiceAffectsElement

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

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

649 **Table 14 – Operations: CIM_ServiceAffectsElement**

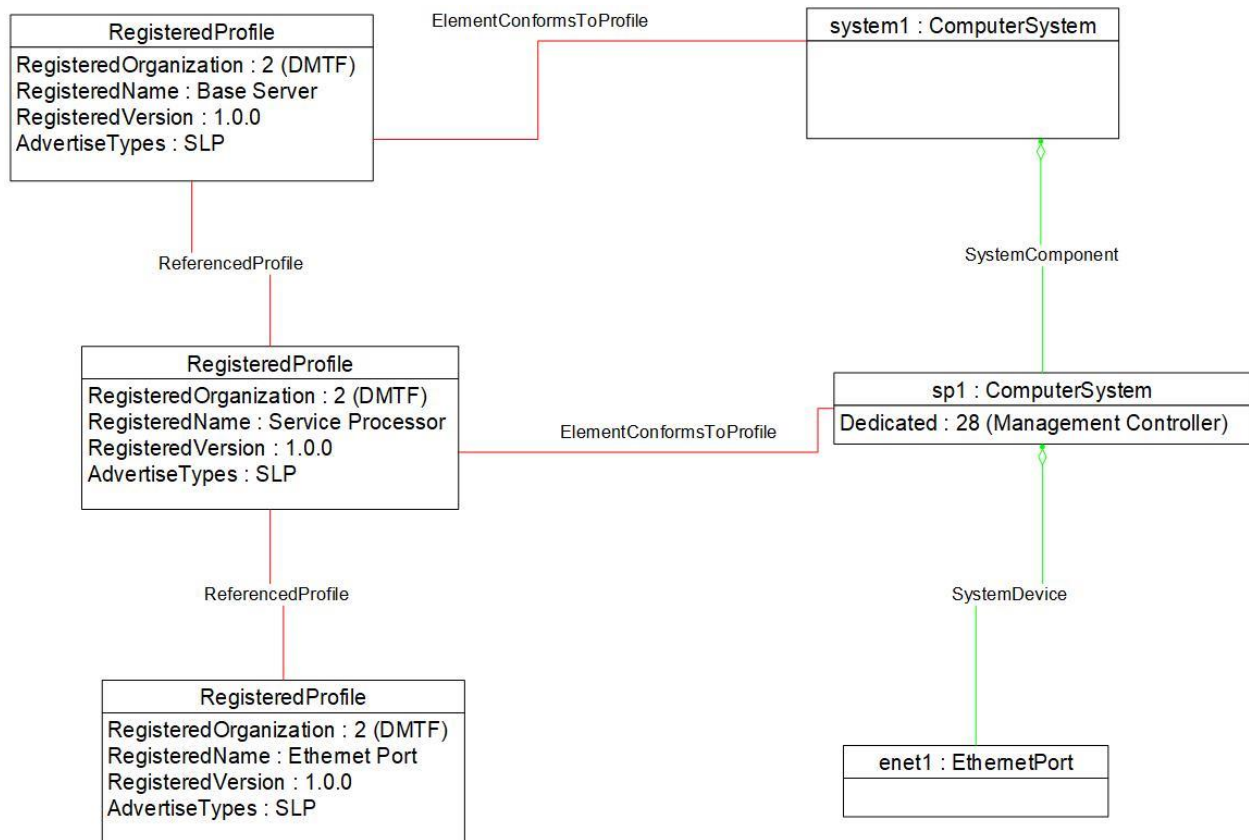
Operation	Requirement	Messages
Associators	Unspecified	None
AssociatorNames	Unspecified	None
References	Unspecified	None
ReferenceNames	Unspecified	None

650 **9 Use cases**

651 This clause contains object diagrams and use cases for the *Service Processor Profile*.

652 **9.1 Object diagrams**

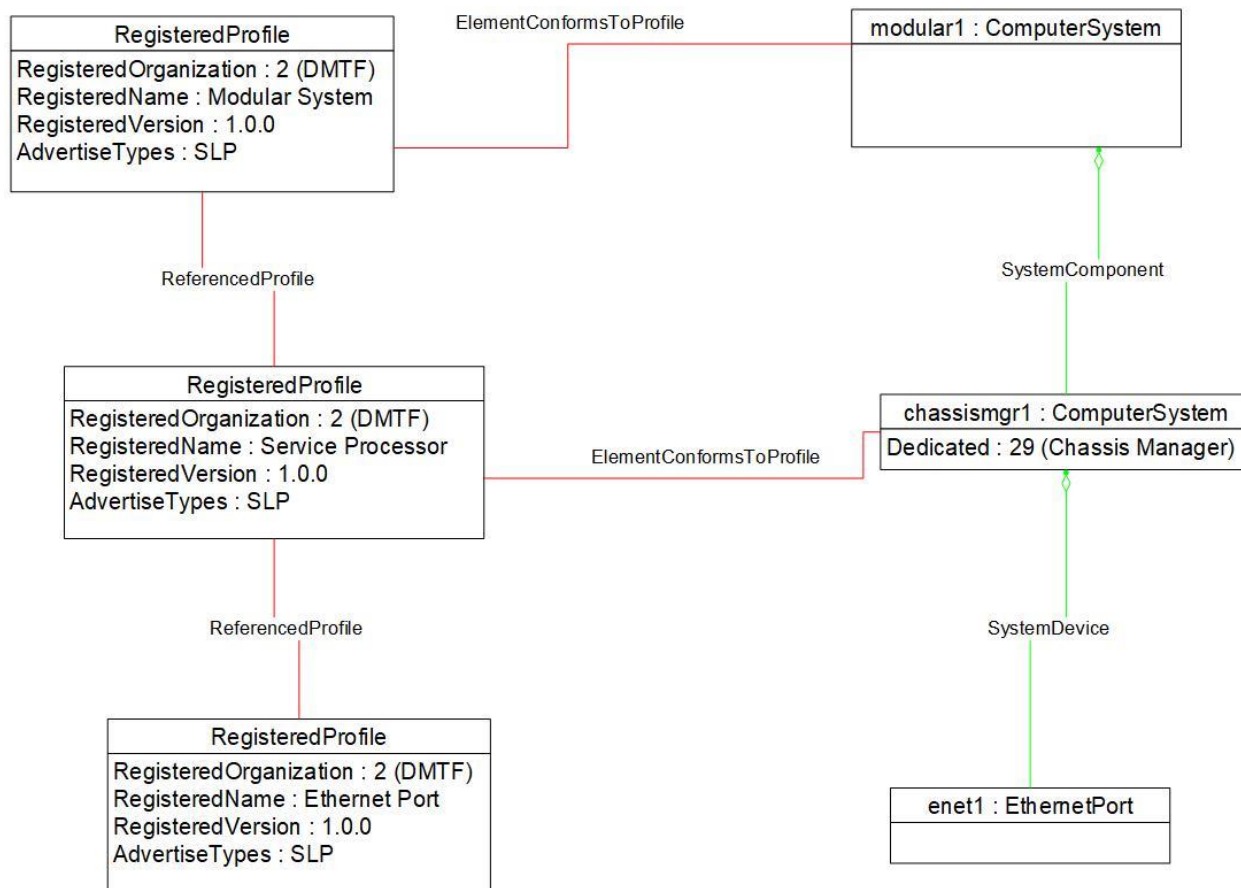
653 Figure 2 depicts an implementation of a service processor dedicated to a single computer system. Notice
 654 that the dedicated property of sp1 is 29 (Management Controller) and the managed computer system,
 655 system1 implements [DSP1004](#). Figure 3 depicts an implementation of a Modular System with a chassis
 656 manager. Notice that the dedicated property of chassismgr1 is 29 (Chassis Manager) and that the
 657 manage system implements [DSP1008](#).



658

659

Figure 2 – Base server



660

661

Figure 3 – Modular system

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

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

691 **9.3 Retrieve the service processor redundancy status**

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

- 693 1) Find the instance of CIM_RedundancySet that is associated with the instance of
694 CIM_ComputerSystem through an instance of CIM_MemberOfCollection.
- 695 2) Retrieve the value of the CIM_RedundancySet.RedundancyStatus property.

696 **9.4 Determine whether manual failover is supported**

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

- 698 1) Starting with an instance of CIM_ComputerSystem, find an instance of CIM_RedundancySet
699 that is associated with the CIM_ComputerSystem instance through the
700 CIM_MemberOfCollection association.
- 701 2) Find all instances of CIM_IsSpare that reference the CIM_RedundancySet instance. Query the
702 FailoverSupported property of each instance. If the FailoverSupported property of any instance
703 has the value of 3 (Manual) or 4 (Both Manual and Automatic), manual failover is supported.

704 **9.5 Force a service processor failover**

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

- 706 1) Starting with the CIM_ComputerSystem instance to failover from, find the instance of
707 CIM_RedundancySet that is associated with the CIM_ComputerSystem instance through the
708 CIM_MemberOfCollection association.
- 709 2) Find an instance of CIM_ComputerSystem associated with the CIM_RedundancySet instance
710 through the CIM_IsSpare association where the CIM_IsSpare.FailoverSupported property has
711 the value of 3 (Manual) or 4 (Both Manual and Automatic). This instance will be the service
712 processor to failover to.
- 713 3) Invoke the CIM_RedundancySet.Failover() method, specifying the CIM_ComputerSystem
714 instance from step 1) as the value for the FailoverFrom parameter and the
715 CIM_ComputerSystem instance from step 2) as the value for the FailoverTo parameter.

716 **9.6 Determine whether the ElementName is modifiable**

717 A client can determine whether it can modify the CIM_ComputerSystem.ElementName property as
718 follows:

- 719 1) Find the CIM_EnabledLogicalElementCapabilities instance that is associated with the
720 CIM_ComputerSystem instance.
- 721 2) Query the value of the ElementNameEditSupported property of the
722 CIM_EnabledLogicalElementCapabilities instance. If the value is TRUE, the client can modify
723 the CIM_ComputerSystem.ElementName property.

724 **9.7 Determining whether state management is supported**

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

- 727 1) Find the CIM_EnabledLogicalElementCapabilities instance that is associated with the
728 CIM_ComputerSystem instance.
- 729 2) Query the value of the RequestedStatesSupported property of the
730 CIM_EnabledLogicalElementCapabilities instance. If at least one value is specified, state
731 management is supported.

732 **10 CIM Elements**

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

736 **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		

737 **10.1 CIM_ComputerSystem**

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

740 **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.

741 **10.2 CIM_ElementCapabilities**

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

744 **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

745 **10.3 CIM_EnabledLogicalElementCapabilities**

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

748 **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.

749 **10.4 CIM_HostedService**

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

752 **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

753 **10.5 CIM_IsSpare**

754 CIM_IsSpare associates an instance of CIM_ComputerSystem with the CIM_RedundancySet for which
 755 the CIM_ComputerSystem instance represents a spare service processor. Table 20 contains the
 756 requirements for properties of the instance.

757 **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.

758 **10.6 CIM_MemberOfCollection**

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

762 **Table 21 – Class: CIM_MemberOfCollection**

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

763 **10.7 CIM_OwningCollectionElement**

764 CIM_OwningCollectionElement associates the CIM_RedundancySet instance with the
 765 CIM_ComputerSystem instance of which the CIM_RedundancySet instance is a member. The instance of
 766 CIM_OwningCollectionElement is conditional on having instantiation of the CIM_RedundancySet class.
 767 Table 22 contains the requirements for properties of the instance.

768 **Table 22 – Class: CIM_OwningCollectionElement**

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

769 **10.8 CIM_RedundancySet**

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

772 **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.

773 **10.9 CIM_RegisteredProfile**

774 CIM_RegisteredProfile identifies the *Service Processor Profile* in order for a client to determine whether
 775 an instance of CIM_ComputerSystem is conformant with this profile. The CIM_RegisteredProfile class is
 776 defined by [DSP1033](#). With the exception of the mandatory values specified for the properties in Table 24,
 777 the behavior of the CIM_RegisteredProfile instance is in accordance with [DSP1033](#).

778 **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 "1.1.2".
RegisteredOrganization	Mandatory	This property shall have a value of 2 (DMTF).

779 **10.10 CIM_ServiceAffectsElement**

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

782 **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)

783 **10.11 CIM_TimeService**

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

786 **Table 26 – Class: CIM_TimeService**

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

787 **10.12 CIM_ManagementController**

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

791 **Table 27 – Class: CIM_ManagementController**

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

792

793

794

795
796
797
798

**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	2012-09-23	Removed HostedAccessPoint for RemoteServiceAccessPoint in Fig 4 and Fig 5 Renamed one of enet2 as enet1 in Fig 4
1.1.2	2019-03-15	This errata addresses these issues: <ul style="list-style-type: none"> • Updated RegisteredVersion to reflect errata version number in clause 10.9 • Added and Updated RegisteredOrganization description to reflect correct value of 2 for DMTF in clause 10.9 and figure 2 in 9.1

799