



1

2

3

4

Document Number: DSP1018

Date: 2009-06-22

Version: 1.0.0

5 **Service Processor Profile**

6 **Document Type: Specification**

7 **Document Status: DMTF Standard**

8 **Document Language: E**

9

10 Copyright Notice

11 Copyright © 2006–2007, 2009 Distributed Management Task Force, Inc. (DMTF). All rights reserved.

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

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26 withdrawn or modified after publication, and shall be indemnified and held harmless by any party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

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

CONTENTS

33	Foreword	7
34	Introduction	8
35	1 Scope	9
36	2 Normative References.....	9
37	2.1 Approved References	9
38	2.2 Other References.....	10
39	3 Terms and Definitions	10
40	4 Symbols and Abbreviated Terms	12
41	5 Synopsis	12
42	6 Description	13
43	7 Implementation.....	14
44	7.1 Representing a Service Processor	14
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).....	18
53	7.10 Software Inventory Profile (Optional).....	18
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)	19
61	7.18 SMASH Collections Profile (Optional)	19
62	7.19 SSH Service Profile (Optional)	19
63	7.20 Telnet Service Profile (Optional).....	19
64	7.21 Text Console Redirection Profile (Optional)	19
65	8 Methods.....	19
66	8.1 Method: CIM_ComputerSystem.RequestStateChange()	19
67	8.2 Method: CIM_RedundancySet.Failover()	20
68	8.3 Method: CIM_TimeService.ManageTime()	21
69	8.4 Profile Conventions for Operations.....	22
70	8.5 CIM_ComputerSystem.....	22
71	8.6 CIM_HostedService	23
72	8.7 CIM_IsSpare	23
73	8.8 CIM_ElementCapabilities	24
74	8.9 CIM_EnabledLogicalElementCapabilities.....	24
75	8.10 CIM_MemberOfCollection	24
76	8.11 CIM_RedundancySet.....	24
77	8.12 CIM_TimeService	24
78	8.13 CIM_ServiceAffectsElement	25
79	9 Use Cases.....	25
80	9.1 Object Diagrams	25
81	9.2 Reset a Service Processor	28
82	9.3 Retrieve the Service Processor Redundancy Status.....	29
83	9.4 Determine Whether Manual Failover Is Supported	29
84	9.5 Force a Service Processor Failover.....	29

85	9.6 Determine Whether the ElementName Is Modifiable	29
86	9.7 Determining If State Management Is Supported	29
87	10 CIM Elements	30
88	10.1 CIM_ComputerSystem.....	30
89	10.2 CIM_ElementCapabilities	31
90	10.3 CIM_EnabledLogicalElementCapabilities.....	31
91	10.4 CIM_HostedService	31
92	10.5 CIM_IsSpare	32
93	10.6 CIM_MemberOfCollection	32
94	10.7 CIM_OwningCollectionElement	32
95	10.8 CIM_RedundancySet.....	33
96	10.9 CIM_RegisteredProfile.....	33
97	10.10 CIM_ServiceAffectsElement	33
98	10.11 CIM_TimeService	34
99	ANNEX A (informative) Change Log.....	35
100		

101 Figures

102	Figure 1 – Service Processor Profile: Class Diagram.....	13
103	Figure 2 – Base Server	25
104	Figure 3 – Modular System	26
105	Figure 4 – Service Processors before Failover.....	27
106	Figure 5 – Service Processors after Failover.....	28
107		

108 Tables

109	Table 1 – Referenced Profiles	12
110	Table 2 – CIM_ComputerSystem.EnabledState Value Description.....	14
111	Table 3 – CIM_ComputerSystem.RequestStateChange() Method: Return Code Values	20
112	Table 4 – CIM_ComputerSystem.RequestStateChange() Method: Parameters.....	20
113	Table 5 – CIM_RedundancySet.Failover() Method: Return Code Values.....	21
114	Table 6 – CIM_RedundancySet.Failover() Method: Parameters.....	21
115	Table 7 – CIM_TimeService.ManageTime() Method: Return Code Values	22
116	Table 8 – CIM_TimeService.ManageTime() Method: Parameters	22
117	Table 9 – Operations: CIM_ComputerSystem.....	22
118	Table 10 – Operations: CIM_HostedService	23
119	Table 11 – Operations: CIM_IsSpare	23
120	Table 12 – Operations: CIM_ElementCapabilities	24
121	Table 13 – Operations: CIM_MemberOfCollection.....	24
122	Table 14 – Operations: CIM_ServiceAffectsElement	25
123	Table 15 – CIM Elements: Service Processor Profile.....	30
124	Table 16 – Class: CIM_ComputerSystem.....	30
125	Table 17 – Class: CIM_ElementCapabilities.....	31
126	Table 18 – Class: CIM_EnabledLogicalElementCapabilities.....	31
127	Table 19 – Class: CIM_HostedService	31
128	Table 20 – Class: CIM_IsSpare	32
129	Table 21 – Class: CIM_MemberOfCollection.....	32
130	Table 22 – Class: CIM_OwningCollectionElement	32

131 Table 23 – Class: CIM_RedundancySet..... 33
132 Table 24 – Class: CIM_RegisteredProfile..... 33
133 Table 25 – Class: CIM_ServiceAffectsElement 33
134 Table 26 – Class: CIM_TimeService 34
135

137

Foreword

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

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

142 **Acknowledgments**

143 The authors wish to acknowledge the following people.

144 **Editor:**

- 145 • Aaron Merkin – IBM

146 **Contributors:**

- 147 • Jon Hass – Dell
- 148 • Khachatur Papanyan – Dell
- 149 • Enoch Suen – Dell
- 150 • Jeff Hilland – HP
- 151 • Christina Shaw – HP
- 152 • Aaron Merkin – IBM
- 153 • Perry Vincent – Intel
- 154 • John Leung – Intel

155

156

Introduction

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

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

163

Service Processor Profile

164 1 Scope

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

166 2 Normative References

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

170 2.1 Approved References

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

173 DMTF DSP0200, *CIM Operations over HTTP 1.2*,
174 http://www.dmtf.org/standards/published_documents/DSP0200_1.2.pdf

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

177 DMTF DSP1004, *Base Server Profile 1.0*,
178 http://www.dmtf.org/standards/published_documents/DSP1004_1.0.pdf

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

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

183 DMTF DSP1008, *Modular System Profile 1.0*,
184 http://www.dmtf.org/standards/published_documents/DSP1008_1.0.pdf

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

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

189 DMTF DSP1011, *Physical Asset Profile 1.0*,
190 http://www.dmtf.org/standards/published_documents/DSP1011_1.0.pdf

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

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

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

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

199 DMTF DSP1021, *Shared Device Management Profile 1.0*,
200 http://www.dmtf.org/standards/published_documents/DSP1021_1.0.pdf

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

203 DMTF DSP1024, *Text Console Redirection Profile 1.0*,
204 http://www.dmtf.org/standards/published_documents/DSP1024_1.0.pdf

205 DMTF DSP1025, *Software Update Profile 1.0*,
206 http://www.dmtf.org/standards/published_documents/DSP1025_1.0.pdf

207 DMTF DSP1027, *Power State Management Profile 1.0*,
208 http://www.dmtf.org/standards/published_documents/DSP1027_1.0.pdf

209 DMTF DSP1033, *Profile Registration Profile 1.0*,
210 http://www.dmtf.org/standards/published_documents/DSP1033_1.0.pdf

211 DMTF DSP1034, *Simple Identity Management Profile 1.0*,
212 http://www.dmtf.org/standards/published_documents/DSP1034_1.0.pdf

213 DMTF DSP1036, *IP Interface Profile 1.0*,
214 http://www.dmtf.org/standards/published_documents/DSP1036_1.0.pdf

215 DMTF DSP1037, *DHCP Client Profile 1.0*,
216 http://www.dmtf.org/standards/published_documents/DSP1037_1.0.pdf

217 DMTF DSP1038, *DNS Client Profile 1.0*,
218 http://www.dmtf.org/standards/published_documents/DSP1038_1.0.pdf

219 DMTF DSP1039, *Role Based Authorization Profile 1.0*,
220 http://www.dmtf.org/standards/published_documents/DSP1039_1.0.pdf

221 **2.2 Other References**

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

224 **3 Terms and Definitions**

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

227 **3.1**

228 **can**

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

230 **3.2**

231 **cannot**

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

233 **3.3**

234 **conditional**

235 indicates requirements to be followed strictly to conform to the document when the specified conditions
236 are met

- 237 **3.4**
238 **mandatory**
239 indicates requirements to be followed strictly to conform to the document and from which no deviation is
240 permitted
- 241 **3.5**
242 **may**
243 indicates a course of action permissible within the limits of the document
- 244 **3.6**
245 **need not**
246 indicates a course of action permissible within the limits of the document
- 247 **3.7**
248 **optional**
249 indicates a course of action permissible within the limits of the document
- 250 **3.8**
251 **referencing profile**
252 indicates a profile that owns the definition of this class and can include a reference to this profile in its
253 "Referenced Profiles" table
- 254 **3.9**
255 **shall**
256 indicates requirements to be followed strictly to conform to the document and from which no deviation is
257 permitted
- 258 **3.10**
259 **shall not**
260 indicates requirements to be followed strictly to conform to the document and from which no deviation is
261 permitted
- 262 **3.11**
263 **should**
264 indicates that among several possibilities, one is recommended as particularly suitable, without
265 mentioning or excluding others, or that a certain course of action is preferred but not necessarily required
- 266 **3.12**
267 **should not**
268 indicates that a certain possibility or course of action is deprecated but not prohibited
- 269 **3.13**
270 **unspecified**
271 indicates that this profile does not define any constraints for the referenced CIM element or operation
- 272 **3.14**
273 **service processor**
274 a specialized device dedicated to management
- 275 **3.15**
276 **standby service processor**
277 an instance of CIM_ComputerSystem that represents a standby service processor of a redundancy set

278 **4 Symbols and Abbreviated Terms**

279 None.

280 **5 Synopsis**

281 **Profile Name:** Service Processor

282 **Version:** 1.0.0

283 **Organization:** DMTF

284 **CIM Schema Version:** 2.20

285 **Central Class:** CIM_ComputerSystem

286 **Scoping Class:** CIM_ComputerSystem

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

288

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.

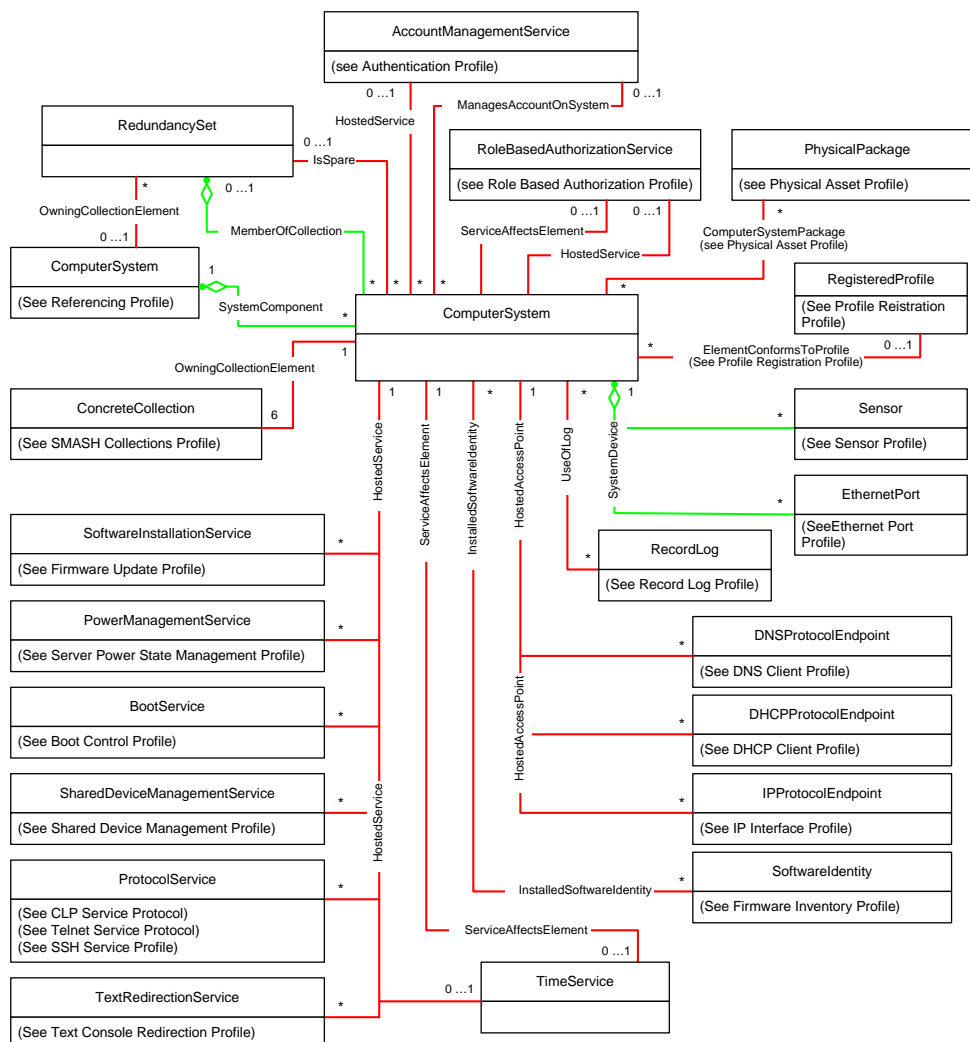
289 **6 Description**

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

294 Some examples of the service processors are:

- 295 • management processor (MP)
- 296 • service processor (SP)
- 297 • baseboard management controller (BMC)
- 298 • chassis manager

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



301

302

Figure 1 – Service Processor Profile: Class Diagram

303 7 Implementation

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

307 7.1 Representing a Service Processor

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

309 7.1.1 CIM_ComputerSystem.EnabledState

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

316 **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).

317 7.1.2 Service Processor State Management Is Supported — Conditional

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

320 7.1.2.1 CIM_EnabledLogicalElementCapabilities

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

324 7.1.2.1.1 CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported

325 The RequestedStatesSupported property may contain zero or more of the following values: 2 (Enabled),
 326 3 (Disabled), 6 (Offline), or 11 (Reset).

327 7.1.2.2 CIM_ComputerSystem.RequestedState

328 When the CIM_ComputerSystem.RequestStateChange() method is successfully invoked, the value of the
 329 RequestedState property shall be the value of the RequestedState parameter. If the method is not
 330 successfully invoked, the value of the RequestedState property is indeterminate.

331 The CIM_ComputerSystem.RequestedState property shall have one of the values specified in the
 332 CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported property or a value of 5 (No
 333 Change).

334 7.1.2.3 **CIM_ComputerSystem.EnabledState**

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

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

339 7.1.3 **Service Processor State Management Is Not Supported**

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

342 7.1.3.1 **CIM_EnabledLogicalElementCapabilities**

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

346 7.1.3.1.1 **CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported**

347 The CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported property shall not contain any
348 values.

349 7.1.3.2 **CIM_ComputerSystem.RequestedState**

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

351 7.1.4 **Modifying ElementName Is Supported — Conditional**

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

356 7.1.4.1 **CIM_EnabledLogicalElementCapabilities**

357 An instance of CIM_EnabledLogicalElementCapabilities shall be associated with the
358 CIM_ComputerSystem instance through an instance of CIM_ElementCapabilities.

359 7.1.4.1.1 **CIM_EnabledLogicalElementCapabilities.ElementNameEditSupported**

360 The ElementNameEditSupported property shall have a value of TRUE.

361 7.1.4.1.2 **CIM_EnabledLogicalElement.MaxElementNameLen**

362 The MaxElementNameLen property shall be implemented.

363 7.1.5 **Modifying ElementName Is Not Supported**

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

366 7.1.5.1 **CIM_EnabledLogicalElementCapabilities**

367 An instance of CIM_EnabledLogicalElementCapabilities may be associated with the
368 CIM_ComputerSystem instance through an instance of CIM_ElementCapabilities.

369 7.1.5.1.1 CIM_EnabledLogicalElementCapabilities.ElementNameEditSupported

370 The ElementNameEditSupported shall have a value of FALSE.

371 7.1.5.1.2 CIM_EnabledLogicalElement.MaxElementNameLen

372 The MaxElementNameLen property may be implemented. The MaxElementNameLen property is
373 irrelevant in this context.

374 7.1.6 Representing the Physical Packaging (Optional)

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

378 7.2 Modeling Service Processor Redundancy (Optional)

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

381 At least one instance of CIM_RedundancySet shall exist.

382 7.2.1 Relationship between Redundancy Set and Redundant Service Processors

383 Each CIM_ComputerSystem instance that represents a service processor participating in the redundancy
384 shall be associated with the CIM_RedundancySet instance through the CIM_MemberOfCollection
385 association. Each instance of CIM_ComputerSystem that is associated with the CIM_RedundancySet
386 instance through the CIM_MemberOfCollection association shall be associated with the same instance of
387 CIM_ComputerSystem through the CIM_SystemComponent association where the value of the
388 CIM_SystemComponent.PartComponent property is the instance of CIM_ComputerSystem that is
389 associated with the CIM_RedundancySet.

390 7.2.2 Relationship between Redundancy Set and Containing System

391 When the CIM_ComputerSystem instance that represents a service processor is associated with another
392 CIM_ComputerSystem instance through the CIM_SystemComponent association where the value of the
393 CIM_SystemComponentPartComponent property is the CIM_ComputerSystem instance that represents
394 the service processor, the CIM_RedundancySet instance shall be associated with the
395 CIM_ComputerSystem instance that is the value of the CIM_SystemComponent.GroupComponent
396 property through the CIM_OwningCollectionElement association.

397 7.2.3 Active / Active Redundancy

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

- 401 • The CIM_ComputerSystem instances shall not be associated with the CIM_RedundancySet
402 instance through the CIM_IsSpare association.
- 403 • For each instance of CIM_ComputerSystem, the CIM_ComputerSystem.EnabledState property
404 shall not have the value 6 (Enabled but Offline).

405 7.2.4 Active / Standby Redundancy

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

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

- 410 • When the CIM_ComputerSystem.EnabledState property has the value 6 (Enabled but Offline),
411 the SpareStatus property of the referencing CIM_IsSpare instance shall have the value 2 (Hot
412 Standby).
- 413 • When the CIM_ComputerSystem.EnabledState property has the value 3 (Disabled), the
414 SpareStatus property of the referencing CIM_IsSpare instance shall have the value 3 (Cold
415 Standby).
- 416 • When the CIM_ComputerSystem.EnabledState property has a value other than 3 (Disabled) or
417 6 (Enabled but Offline), the SpareStatus property of the referencing CIM_IsSpare instance shall
418 have the value 0 (Unknown).

419 **7.3 Managing Service Processor Time (Optional)**

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

424 An instance of CIM_TimeService shall be associated with the Central Instance through the
425 CIM_HostedService association. The instance of CIM_TimeService shall also be associated with the
426 Central Instance through the CIM_ServiceAffectsElement association.

427 **7.4 User Account Management (Optional)**

428 The [Simple Identity Management Profile](#) and the [Role Based Authorization Profile](#) may be implemented to
429 model user access to the service processor. When the [Simple Identity Management Profile](#) is
430 implemented, an instance of CIM_AccountManagementService shall be associated with the Central
431 Instance through the CIM_HostedService association. When the [Role Based Authorization Profile](#) is
432 implemented, an instance of CIM_RoleBasedAuthorizationService shall be associated with the Central
433 Instance through the CIM_HostedService association.

434 **7.5 Boot Control Profile (Optional)**

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

439 **7.6 CLP Service Profile (Optional)**

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

443 **7.7 DHCP Client Profile (Optional)**

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

447 **7.8 DNS Client Profile (Optional)**

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

451 **7.9 Ethernet Port Profile (Optional)**

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

455 **7.10 Software Inventory Profile (Optional)**

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

460 **7.11 Software Update Profile (Optional)**

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

465 **7.12 IP Interface Profile (Optional)**

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

469 **7.13 Physical Asset Profile (Optional)**

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

474 **7.14 Record Log Profile (Optional)**

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

478 **7.15 Sensors Profile (Optional)**

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

482 **7.16 Power State Management Profile (Optional)**

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

487 7.17 Shared Device Management Profile (Optional)

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

492 7.18 SMASH Collections Profile (Optional)

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

496 7.19 SSH Service Profile (Optional)

497 The [SSH Service Profile](#) may be implemented to model an SSH service hosted on the service processor.
498 When the [SSH Service Profile](#) is implemented, at least one instance of CIM_ProtocolService shall be
499 associated with the Central Instance through an instance of CIM_HostedService where the
500 CIM_ProtocolService.Protocol property has the value 2 (SSH).

501 7.20 Telnet Service Profile (Optional)

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

506 7.21 Text Console Redirection Profile (Optional)

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

511 8 Methods

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

514 8.1 Method: CIM_ComputerSystem.RequestStateChange()

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

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

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

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

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

526 No standard messages are defined for this method.

527 **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.

528 **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

529 **8.1.1 RequestStateChange() for the Standby Service Processor**

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

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

536 **8.2 Method: CIM_RedundancySet.Failover()**

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

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

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

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

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

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

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

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

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

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.

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

8.3 Method: CIM_TimeService.ManageTime()

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

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

579 **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.

580 **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

581 **8.4 Profile Conventions for Operations**

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

584 The default list of operations is as follows:

- 585 • GetInstance
- 586 • Associators
- 587 • AssociatorNames
- 588 • References
- 589 • ReferenceNames
- 590 • EnumerateInstances
- 591 • EnumerateInstanceNames

592 **8.5 CIM_ComputerSystem**

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

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

597 **Table 9 – Operations: CIM_ComputerSystem**

Operation	Requirement	Messages
ModifyInstance	Optional. See 8.5.1.	None

598 **8.5.1 CIM_ComputerSystem — ModifyInstance**

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

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

605 **8.5.1.1 CIM_ComputerSystem.ElementName**

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

611 When the ElementNameEditSupported property of the CIM_EnabledLogicalElementCapabilities instance
 612 has a value of FALSE, the implementation shall not allow the ModifyInstance operation to change the
 613 value of the ElementName property of the CIM_ComputerSystem instance.

614 **8.6 CIM_HostedService**

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

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

619 **Table 10 – Operations: CIM_HostedService**

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

620 **8.7 CIM_IsSpare**

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

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

625 **Table 11 – Operations: CIM_IsSpare**

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

626 8.8 CIM_ElementCapabilities

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

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

631 **Table 12 – Operations: CIM_ElementCapabilities**

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

632 8.9 CIM_EnabledLogicalElementCapabilities

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

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

635 8.10 CIM_MemberOfCollection

636 Table 13 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 13, 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 13 – Operations: CIM_MemberOfCollection**

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

641 8.11 CIM_RedundancySet

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 8.12 CIM_TimeService

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

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

647 **8.13 CIM_ServiceAffectsElement**

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

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

652 **Table 14 – Operations: CIM_ServiceAffectsElement**

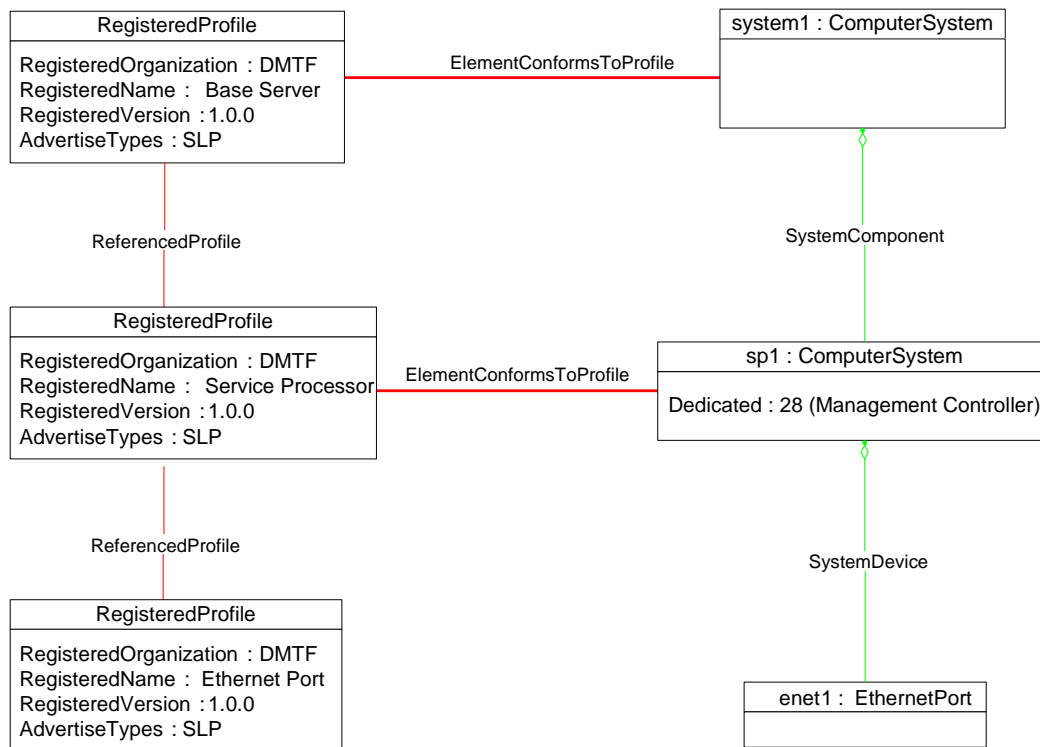
Operation	Requirement	Messages
Associators	Unspecified	None
AssociatorNames	Unspecified	None
References	Unspecified	None
ReferenceNames	Unspecified	None

653 **9 Use Cases**

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

655 **9.1 Object Diagrams**

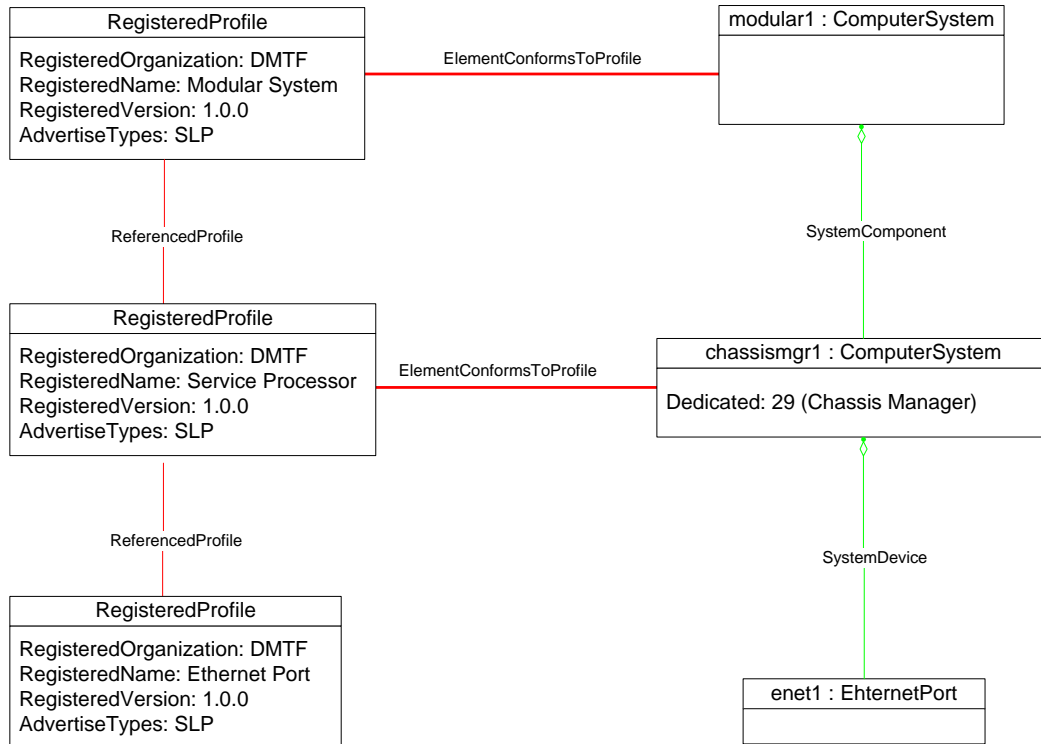
656 Figure 2 depicts an implementation of a service processor dedicated to a single computer system. Notice
 657 that the dedicated property of sp1 is 29 (Management Controller) and the managed computer system,
 658 system1 implements the [Base Server Profile](#). Figure 3 depicts an implementation of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



661

662

Figure 2 – Base Server



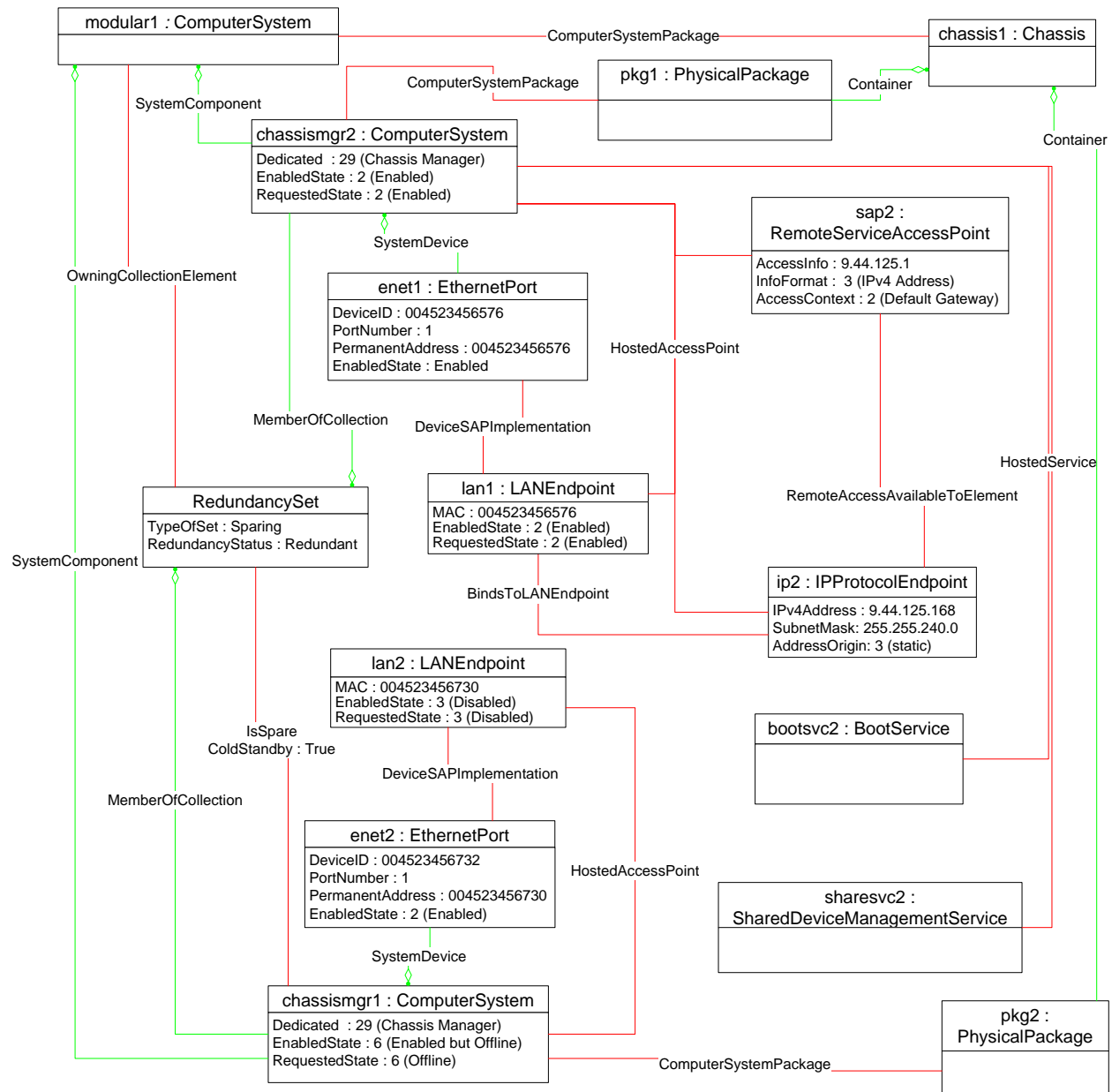
663

664

Figure 3 – Modular System

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

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



684

685

Figure 5 – Service Processors after Failover

686 **9.2 Reset a Service Processor**

687 A client can reset the service processor as follows:

- 688 1) For the given instance of CIM_ComputerSystem, find the associated instance of
 689 CIM_EnabledLogicalElementCapabilities.
- 690 2) If the CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported property is a non-
 691 empty array that contains the value 11 (Reset), execute the RequestStateChange()
 692 with the value of the RequestedState parameter set to 11 (Reset).

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

694 **9.3 Retrieve the Service Processor Redundancy Status**

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

- 696 1) Find the instance of CIM_RedundancySet that is associated with the instance of
697 CIM_ComputerSystem through an instance of CIM_MemberOfCollection.
- 698 2) Retrieve the value of the CIM_RedundancySet.RedundancyStatus property.

699 **9.4 Determine Whether Manual Failover Is Supported**

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

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

707 **9.5 Force a Service Processor Failover**

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

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

719 **9.6 Determine Whether the ElementName Is Modifiable**

720 A client can determine whether it can modify the CIM_ComputerSystem.ElementName property as
721 follows:

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

727 **9.7 Determining If State Management Is Supported**

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

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

735 **10 CIM Elements**

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

739 **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.
Indications		
None defined in this profile		

740 **10.1 CIM_ComputerSystem**

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

743 **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.

744 **10.2 CIM_ElementCapabilities**

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

747 **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

748 **10.3 CIM_EnabledLogicalElementCapabilities**

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

751 **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.

752 **10.4 CIM_HostedService**

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

755 **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

756 **10.5 CIM_IsSpare**

757 CIM_IsSpare associates an instance of CIM_ComputerSystem with the CIM_RedundancySet for which
 758 the CIM_ComputerSystem instance represents a spare service processor. Table 20 contains the
 759 requirements for properties of the instance.

760 **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.

761 **10.6 CIM_MemberOfCollection**

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

765 **Table 21 – Class: CIM_MemberOfCollection**

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

766 **10.7 CIM_OwningCollectionElement**

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

771 **Table 22 – Class: CIM_OwningCollectionElement**

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

772 **10.8 CIM_RedundancySet**

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

775 **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.

776 **10.9 CIM_RegisteredProfile**

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

782 **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 "1.0.0".

783 **10.10 CIM_ServiceAffectsElement**

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

786 **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)

787 **10.11 CIM_TimeService**

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

790

Table 26 – Class: CIM_TimeService

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

791

792
793
794
795

ANNEX A (informative)

Change Log

Version	Date	Description
1.0.0b	08-30-2006	Added Class Diagram
1.0.0e	03-06-2007	Changed name to Service Processor, edited ComputerSystem.Dedicated to reflect single or modular system management; updated usecases to include an instance diagram of both.
1.0.0	06-22-2009	DMTF Standard Release

796