



1

2

3

4

Document Number: DSP1040

Date: 2009-06-19

Version: 1.0.0

5 **Platform Watchdog Profile**

6 **Document Type: Specification**

7 **Document Status: DMTF Standard**

8 **Document Language: E**

9 Copyright Notice

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

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

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25 withdrawn or modified after publication, and shall be indemnified and held harmless by any party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

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

CONTENTS

32	1	Scope	7
33	2	Normative References.....	7
34	2.1	Approved References	7
35	2.2	Other References.....	7
36	3	Terms and Definitions	7
37	4	Symbols and Abbreviated Terms	9
38	5	Synopsis.....	9
39	6	Description (Informative)	9
40	7	Implementation.....	12
41	7.1	Representing a Watchdog	12
42	7.2	Representing the Watchdog Device (Optional)	18
43	7.3	Representing the Monitored Entity (Optional)	18
44	7.4	Representing the Entity on Which Action on Expiration Is Taken (Optional)	18
45	7.5	State Management of a Watchdog (Optional)	18
46	8	Methods.....	20
47	8.1	CIM_PlatformWatchdogService.RequestStateChange().....	20
48	8.2	Profile Conventions for Operations.....	21
49	8.3	CIM_DeviceServiceImplementation Operations.....	21
50	8.4	CIM_ElementCapabilities Operations.....	21
51	8.5	CIM_HostedService Operations	22
52	8.6	CIM_ServiceAffectsElement Operations	22
53	8.7	CIM_ServiceAvailableToElement Operations	22
54	8.8	CIM_LogicalDevice Operations	23
55	8.9	CIM_PlatformWatchdogService Operations.....	23
56	8.10	CIM_PlatformWatchdogServiceCapabilities Operations	23
57	9	Use Cases (Informative).....	24
58	9.1	Advertising the Profile Conformance	24
59	9.2	Object Diagram for a Monolithic Server.....	24
60	9.3	Object Diagram for a Monolithic Server with a Service Processor.....	25
61	9.4	Object Diagram for a Monolithic System with a Watchdog Device	26
62	9.5	Object Diagram for OS Reset When a Watchdog Timer Expires.....	26
63	9.6	Object Diagram for System with Multiple Watchdogs.....	27
64	9.7	Representing the Watchdog States	28
65	9.8	Finding the Watchdogs Hosted on a Computer System.....	29
66	9.9	Finding the Watchdogs Monitoring an Entity Type	29
67	9.10	Finding the Watchdogs Monitoring an Entity Instance	29
68	9.11	Determining Whether a Watchdog Supports State Management	29
69	9.12	Activating a Watchdog	30
70	9.13	Obtaining Information Regarding the Last Watchdog Expiration	30
71	9.14	Determining Whether CIM_PlatformWatchdogService.ElementName Can Be Modified.....	30
72	10	CIM Elements.....	30
73	10.1	CIM_RegisteredProfile.....	31
74	10.2	CIM_DeviceServiceImplementation.....	31
75	10.3	CIM_ElementCapabilities	32
76	10.4	CIM_HostedService	32
77	10.5	CIM_ServiceAffectsElement, Relating CIM_PlatformWatchdogService to CIM_ComputerSystem.....	32
78	10.6	CIM_ServiceAffectsElement, Relating CIM_PlatformWatchdogService to a Concrete Subclass of CIM_LogicalElement.....	33
79	10.7	CIM_ServiceAvailableToElement	33
80			
81			

82	10.8 CIM_LogicalDevice	33
83	10.9 CIM_PlatformWatchdogService.....	34
84	10.10 CIM_PlatformWatchdogServiceCapabilities	34

85
86

Figures

87	Figure 1 – Platform Watchdog Profile: Class Diagram	11
88	Figure 2 – CIM_PlatformWatchdogService State Chart	12
89	Figure 3 – Registered Profile	24
90	Figure 4 – Object Diagram: Monolithic Server	25
91	Figure 5 – Object Diagram: Monolithic Server with a Service Processor	25
92	Figure 6 – Object Diagram: Reset System When a Watchdog Expires	26
93	Figure 7 – Object Diagram: Reset OS When a Watchdog Timer Expires	26
94	Figure 8 – Object Diagram: System with Multiple Watchdogs.....	27
95	Figure 9 – CIM_PlatformWatchdogService Property Values per Watchdog State	28

96

Tables

98	Table 1 – Referenced Profiles	9
99	Table 2 – CIM_PlatformWatchdogService.RequestStateChange() Method: Return Code Values	20
100	Table 3 – CIM_PlatformWatchdogService.RequestStateChange() Method: Parameters	20
101	Table 4 – Operations: CIM_DeviceServiceImplementation	21
102	Table 5 – Operations: CIM_ElementCapabilities	22
103	Table 6 – Operations: CIM_HostedService	22
104	Table 7 – Operations: CIM_ServiceAffectsElement	22
105	Table 8 – Operations: CIM_ServiceAvailableToElement.....	22
106	Table 9 – Operations: CIM_LogicalDevice	23
107	Table 10 – Operations: CIM_PlatformWatchdogService	23
108	Table 11 – Operations: CIM_PlatformWatchdogServiceCapabilities	24
109	Table 12 – CIM Elements: Platform Watchdog Profile	31
110	Table 13 – Class: CIM_RegisteredProfile	31
111	Table 14 – Class: CIM_DeviceServiceImplementation.....	31
112	Table 15 – Class: CIM_ElementCapabilities.....	32
113	Table 16 – Class: CIM_HostedService	32
114	Table 17 – Class: CIM_ServiceAffectsElement Referencing CIM_ComputerSystem	32
115	Table 18 – Class: CIM_ServiceAffectsElement Referencing CIM_LogicalElement	33
116	Table 19 – Class: CIM_ServiceAvailableToElement	33
117	Table 20 – Class: CIM_LogicalDevice	33
118	Table 21 – Class: CIM_PlatformWatchdogService.....	34
119	Table 22 – Class: CIM_PlatformWatchdogServiceCapabilities	34

120

121

Foreword

122 The *Platform Watchdog Profile* (DSP1040) was prepared by the Server Management Working Group and
123 the Physical Platform Profiles Working Group of the DMTF.

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

126 Acknowledgments

127 The authors wish to acknowledge the following people.

128 Editor:

- 129 • John Leung – Intel

130 Contributors:

- 131 • Aaron Merkin – IBM
- 132 • Jon Hass – Dell
- 133 • Khachatur Papanyan – Dell
- 134 • Jeff Hilland – HP
- 135 • Christina Shaw – HP
- 136 • Joel Clark – Intel

137

Introduction

138 The information in this specification and referenced specifications should be sufficient for a provider or
139 consumer of this data to identify unambiguously the classes, properties, methods, and values that shall
140 be instantiated and manipulated using the DMTF CIM core and common model definitions.

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

143

Platform Watchdog Profile

144 1 Scope

145 The *Platform Watchdog Profile* extends the management capabilities of referencing profiles by providing
146 the capability to manage watchdog timers provided by the system.

147 2 Normative References

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

151 2.1 Approved References

152 DMTF DSP0004, *CIM Infrastructure Specification 2.3*,
153 http://www.dmtf.org/standards/published_documents/DSP0004_2.3.pdf

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

156 DMTF DSP1001, *Management Profile Specification Usage Guide 1.0*,
157 http://www.dmtf.org/standards/published_documents/DSP1001.pdf

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

160 2.2 Other References

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

163 IETF RFC5234, *Augmented BNF for Syntax Specifications: ABNF*, January 2008,
164 <http://www.ietf.org/rfc/rfc5234.txt?number=5234>

165 3 Terms and Definitions

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

168 3.1

169 **can**

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

171 3.2

172 **cannot**

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

174 3.3

175 **conditional**

176 indicates requirements to be followed strictly to conform to the document when the specified conditions
177 are met

- 178 **3.4**
179 **mandatory**
180 indicates requirements to be followed strictly to conform to the document and from which no deviation is
181 permitted
- 182 **3.5**
183 **may**
184 indicates a course of action permissible within the limits of the document
- 185 **3.6**
186 **need not**
187 indicates a course of action permissible within the limits of the document
- 188 **3.7**
189 **optional**
190 indicates a course of action permissible within the limits of the document
- 191 **3.8**
192 **referencing profile**
193 indicates a profile that owns the definition of this class and can include a reference to this profile in its
194 "Referenced Profiles" table
- 195 **3.9**
196 **shall**
197 indicates requirements to be followed strictly to conform to the document and from which no deviation is
198 permitted
- 199 **3.10**
200 **shall not**
201 indicates requirements to be followed strictly to conform to the document and from which no deviation is
202 permitted
- 203 **3.11**
204 **should**
205 indicates that among several possibilities, one is recommended as particularly suitable, without
206 mentioning or excluding others, or that a certain course of action is preferred but not necessarily required
- 207 **3.12**
208 **should not**
209 indicates that a certain possibility or course of action is deprecated but not prohibited
- 210 **3.13**
211 **unspecified**
212 indicates that this profile does not define any constraints for the referenced CIM element or operation
- 213 **3.14**
214 **Watchdog**
215 **watchdog timer**
216 a timer mechanism used to monitor the health of a software or hardware entity

217 **4 Symbols and Abbreviated Terms**

218 **4.1**
 219 **BIOS**
 220 basic input output system

221 **4.2**
 222 **OS**
 223 operating system

224 **4.3**
 225 **UTC**
 226 Coordinated Universal Time

227 **5 Synopsis**

228 **Profile Name:** Platform Watchdog
 229 **Version:** 1.0.0
 230 **Organization:** DMTF
 231 **CIM Schema Version:** 2.22
 232 **Central Class:** CIM_PlatformWatchdogService
 233 **Scoping Class:** CIM_ComputerSystem

234 The *Platform Watchdog Profile* is a component profile that extends the management capability of the
 235 referencing profiles by adding the capability to describe Watchdog information.

236 Table 1 identifies the profile on which this profile has a dependency.

237 CIM_PlatformWatchdogService shall be the Central Class of the *Platform Watchdog Profile*. The
 238 instances of CIM_PlatformWatchdogService shall be the Central Instances of this profile.

239 CIM_ComputerSystem shall be the Scoping Class of this profile. The instance of CIM_ComputerSystem
 240 with which the Central Instance is associated through an instance of CIM_HostedService shall be the
 241 Scoping Instance of this profile.

242 **Table 1 – Referenced Profiles**

Profile Name	Organization	Version	Relationship	Behavior
Profile Registration	DMTF	1.0	Mandatory	

243 **6 Description (Informative)**

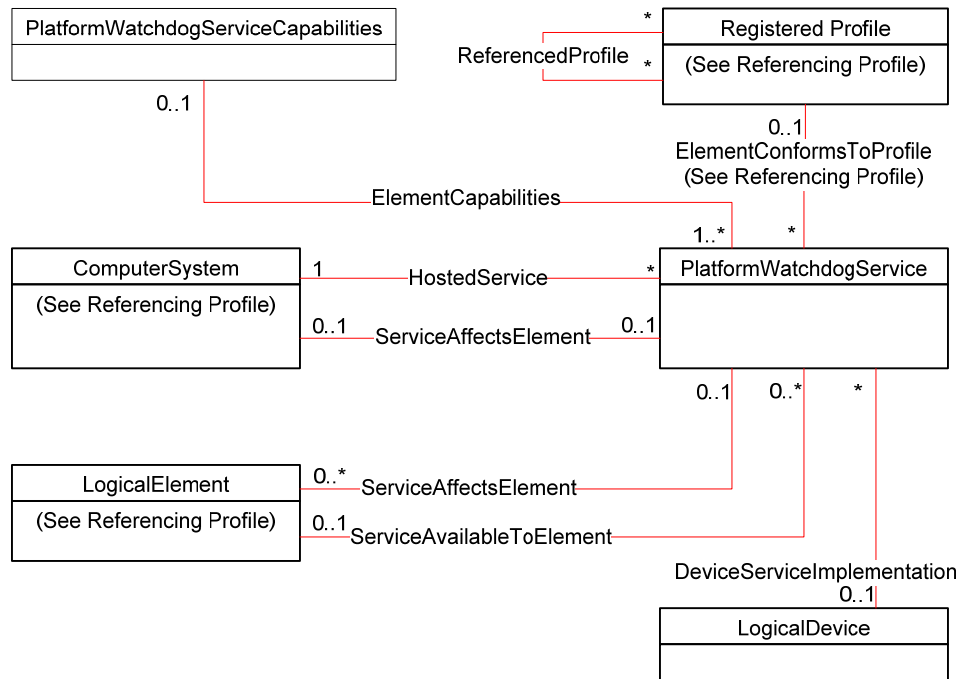
244 The *Platform Watchdog Profile* describes the elements needed to provide the capability to manage a
 245 Watchdog. A Watchdog is a timer mechanism used to monitor the health of a software or hardware entity.

246 A watchdog timer behaves as follows:

- 247 1) The timeout value is set to an initial value.
- 248 2) The timer is enabled, which causes it to begin a monotonic countdown from the timeout value to
 249 zero.

- 250 The countdown is performed with a specified timer resolution. When the timer reaches zero, the
251 Watchdog expires.
- 252 Timeout values are represented in the data model using the interval format of the datetime type. The
253 datetime format for intervals is ddddddhhmmss.mmmmm:000, where the meaning of each field is as
254 follows:
- 255 • dddddd is the number of days.
 - 256 • hh is the remaining number of hours.
 - 257 • mm is the remaining number of minutes.
 - 258 • ss is the remaining number of seconds.
 - 259 • mmmmm is the remaining number of microseconds.
 - 260 • A colon (:) indicates that the value is an interval.
 - 261 • 000 (the UTC offset field) is always zero for interval properties.
- 262 Fields that are not significant are replaced with asterisk (*) characters. Non-significant fields are those that
263 are beyond the resolution of the data source.
- 264 A timeout value might start, for example, with a value of 0000000000001.000***:000, which represents a
265 timeout of 1 second expressed with a 1 millisecond precision. After some time counting down, the
266 remaining timeout value might be 0000000000000.125***.000, which indicates that 125 milliseconds
267 remain before the timer expires.
- 268 The watchdog timer is continuously prevented from expiring if the monitored entity is operational. This is
269 accomplished by either stopping the timer within a specified timeout interval or periodically resetting the
270 value of the timeout interval.
- 271 The expiration of the watchdog timer can cause a specific action to be performed. The action can be
272 performed on the monitored entity or on another element on the platform.
- 273 One example is a Watchdog that monitors the operating system and resets the computer system if the
274 watchdog timer expires. Another example is a Watchdog that monitors an application and generates a
275 non-maskable interrupt if the watchdog timer expires.
- 276 A system can have zero or more Watchdogs.

277 Figure 1 presents the class diagram for the *Platform Watchdog Profile*. For simplicity, the prefix *CIM_* has
 278 been removed from the names of the classes.



279

280

Figure 1 – Platform Watchdog Profile: Class Diagram

281 A computer system may host one or more Watchdogs. Each instance of *CIM_PlatformWatchdogService*
 282 that represents a Watchdog is associated with the *CIM_ComputerSystem* instance through the
 283 *CIM_HostedService* association.

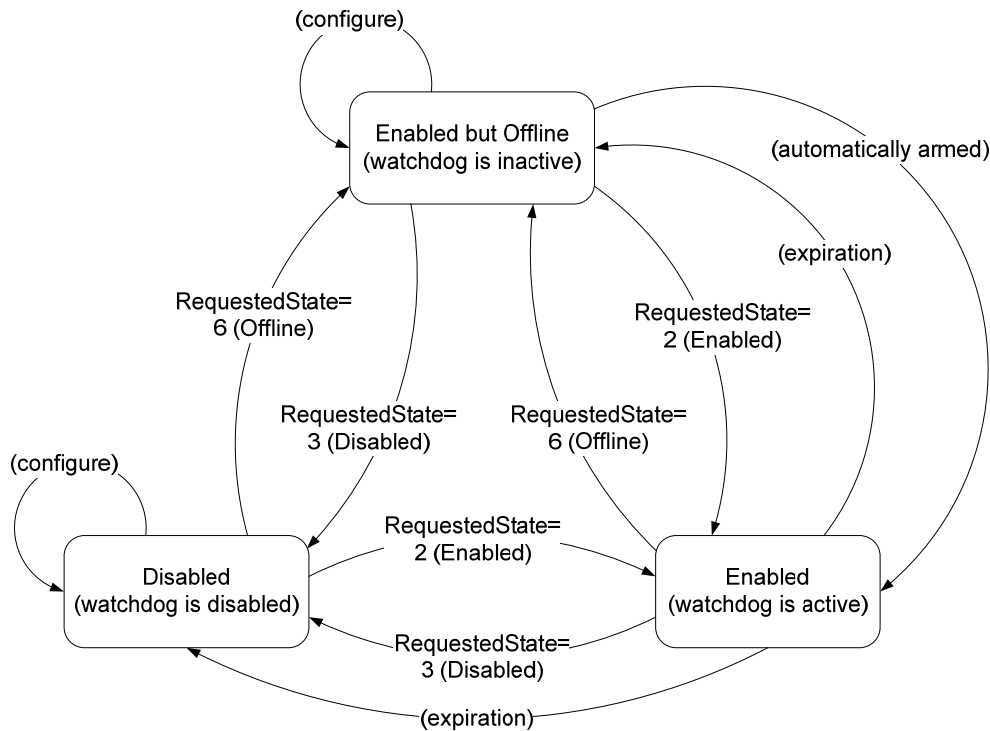
284 The concrete instance of *CIM_LogicalElement* that represents the monitored entity is associated with the
 285 *CIM_PlatformWatchdogService* instance through the *CIM_ServiceAvailableToElement* association.

286 The concrete instance of *CIM_LogicalElement* that represents the element upon which action is
 287 performed on expiration of the watchdog timer is associated with the *CIM_PlatformWatchdogService*
 288 instance through the *CIM_ServiceAffectsElement* association.

289 The Watchdog may be implemented as a device. The concrete instance of *CIM_LogicalDevice* that
 290 represents the device that implements the Watchdog is associated with the
 291 *CIM_PlatformWatchdogService* instance through the *CIM_DeviceServiceImplementation* association.

292 Figure 2 shows the state chart for the *CIM_PlatformWatchdogService* instance. Pre-configured
 293 Watchdogs are initially inactive, which means they have been configured and are ready to be enabled. An
 294 inactive Watchdog can automatically arm (enable) itself when the system is powered on. When a
 295 watchdog timer is counting down, the Watchdog is active. When a watchdog timer expires, the Watchdog
 296 becomes either inactive or disabled. A disabled Watchdog will not automatically arm itself; it requires an
 297 explicit state change request. The *CIM_PlatformWatchdogService.TimerExpired* property (a Boolean
 298 property) is used to distinguish between an “Enabled but Offline” Watchdog that has never been active
 299 and one whose timer has expired.

Enabled, but Offline
- Pre-Configure Watchdogs should be found in this state



300

301

Figure 2 – CIM_PlatformWatchdogService State Chart

302 7 Implementation

303 This section describes the classes and class properties required by the *Platform Watchdog Profile*.
 304 Section 8 describes the class methods required by the profile.

305 7.1 Representing a Watchdog

306 An instance of CIM_PlatformWatchdogService shall be used to represent a Watchdog.

307 7.1.1 CIM_PlatformWatchdogServiceCapabilities

308 Exactly one instance of CIM_PlatformWatchdogServiceCapabilities may be associated with each
 309 instance of CIM_PlatformWatchdogService through an instance of the CIM_ElementCapabilities
 310 association. The CIM_ElementCapabilities association's ManagedElement property shall reference the
 311 instance of CIM_PlatformWatchdogService, and its Capabilities property shall reference the instance of
 312 CIM_PlatformWatchdogServiceCapabilities.

313 7.1.2 Relationship to the Hosting Computer System

314 The instance of CIM_PlatformWatchdogService shall be associated with the Scoping Instance through
315 the CIM_HostedService association.

316 The CIM_HostedService association shall associate the instance of CIM_PlatformWatchdogService with
317 the instance of CIM_ComputerSystem. The CIM_HostedService association's Antecedent property shall
318 reference the CIM_ComputerSystem instance, and its Dependent property shall reference the
319 CIM_PlatformWatchdogService instance.

320 7.1.3 States of a Watchdog

321 A Watchdog can be active, inactive, or disabled. When the Watchdog is inactive or disabled, the timer
322 mechanism is not counting down. When the Watchdog is active, the timer mechanism is counting down.

- 323 • The Watchdog shall be in an active state when its CIM_PlatformWatchdogService.EnabledState
324 property has a value of 2 (Enabled).
- 325 • The Watchdog shall be in a disabled state when its
326 CIM_PlatformWatchdogService.EnabledState property has a value of 3 (Disabled).
- 327 • The Watchdog shall be in an inactive state when its
328 CIM_PlatformWatchdogService.EnabledState property has a value of 6 (Enabled but Offline).

329 7.1.4 Timeout Interval

330 The value of the CIM_PlatformWatchdogService.TimeoutInterval property shall represent the initial value
331 of the watchdog timer. The TimeoutInterval property shall use the interval notation for the datetime.

332 7.1.5 Timer Expired

333 The CIM_PlatformWatchdogService.TimerExpired property shall a value of FALSE when the watchdog
334 did not expire the last time it was active, or if this information is unknown. The
335 CIM_PlatformWatchdogService.TimerExpired property shall have the value of TRUE when the watchdog
336 expired the last time it was active.

337 7.1.6 Timer Resolution (Optional)

338 Support for the CIM_PlatformWatchdogService.TimerResolution property is optional. This subclause
339 describes the CIM elements and behaviors that allow the client to determine whether the TimerResolution
340 property is supported.

341 7.1.6.1 TimerResolution Is Supported — Conditional

342 This subclause describes the CIM elements and behaviors that shall be implemented when the
343 TimerResolution property is supported.

344 7.1.6.1.1 CIM_PlatformWatchdogServiceCapabilities

345 When the TimerResolution property is supported, exactly one instance of
346 CIM_PlatformWatchdogServiceCapabilities shall be associated with the instance of
347 CIM_PlatformWatchdogService through an instance of CIM_ElementCapabilities.

348 7.1.6.1.1.1 CIM_PlatformWatchdogServiceCapabilities.WatchdogTimerDataSupported

349 The WatchdogTimerDataSupported array property shall contain a value of 3 (Timer Resolution).

350 **7.1.6.1.2 CIM_PlatformWatchdogService.TimerResolution**

351 The TimerResolution property shall use the interval notation of the datetime type.

352 When the Watchdog is in an inactive or disabled state, and CIM_PlatformWatchdogService.TimerExpired
353 has the value TRUE, the TimerResolution property shall represent the resolution of the watchdog timer
354 over the timeout interval. Otherwise, the value of the TimerResolution property may be indeterminate.

355 The value of the TimerResolution property shall be interpreted to mean that the watchdog timer expired
356 between TimeoutInterval-TimerResolution and TimeoutInterval+TimerResolution.

357 **7.1.6.2 TimerResolution Is Not Supported**

358 This subclause describes the CIM elements and behaviors that shall be implemented when the
359 TimerResolution property is not supported.

360 **7.1.6.2.1 CIM_PlatformWatchdogServiceCapabilities**

361 When the TimerResolution property is not supported, an instance of
362 CIM_PlatformWatchdogServiceCapabilities may be associated with the instance of
363 CIM_PlatformWatchdogService through an instance of CIM_ElementCapabilities.

364 **7.1.6.2.1.1 CIM_PlatformWatchdogServiceCapabilities.WatchdogTimerDataSupported**

365 The WatchdogTimerDataSupported array property shall not contain a value of 3 (Timer Resolution).

366 **7.1.6.2.2 CIM_PlatformWatchdogService.TimerResolution**

367 The value of the CIM_PlatformWatchdogService.TimerResolution property shall be irrelevant in this
368 context.

369 **7.1.7 Type of Monitored Entity**

370 The value of the CIM_PlatformWatchdogService.MonitoredEntityType property shall represent the type of
371 entity being monitored.

372 When an instance of a concrete subclass of CIM_LogicalElement that represents the entity being
373 monitored exists, its relationship to the Watchdog may be modeled as described in 7.3.

374 **7.1.8 Current Timer Value (Optional)**

375 Support for the CIM_PlatformWatchdogService.CurrentTimerValue property is optional. This subclause
376 describes the CIM elements and behaviors that allow the client to determine whether the
377 CurrentTimerValue property is supported.

378 **7.1.8.1 CurrentTimerValue Is Supported — Conditional**

379 This subclause describes the CIM elements and behaviors that shall be implemented when the
380 CurrentTimerValue property is supported.

381 **7.1.8.1.1 CIM_PlatformWatchdogServiceCapabilities**

382 When the CurrentTimerValue property is supported, exactly one instance of
383 CIM_PlatformWatchdogServiceCapabilities shall be associated with the instance of
384 CIM_PlatformWatchdogService through an instance of CIM_ElementCapabilities.

385 7.1.8.1.1.1 CIM_PlatformWatchdogServiceCapabilities.WatchdogTimerDataSupported

386 The WatchdogTimerDataSupported array property shall contain a value of 2 (Current Value).

387 7.1.8.1.2 CIM_PlatformWatchdogService.CurrentTimerValue

388 The CurrentTimerValue property shall use the interval notation of the datetime type.

389 When the Watchdog is in an active state, the CurrentTimerValue property shall have the current value of
390 the watchdog timer. Otherwise, the value of the CurrentTimerValue property may be indeterminate.

391 When the Watchdog is activated, the value of the CurrentTimerValue property shall initially match the
392 value of the TimeoutInterval property.

393 When the watchdog timer expires, the value of the CurrentTimerValue property shall have a value of 0
394 (00000000000000.000000:000).

395 7.1.8.2 CurrentTimerValue Is Not Supported

396 This subclause describes the CIM elements and behaviors that shall be implemented when the
397 CurrentTimerValue property is not supported.

398 7.1.8.2.1 CIM_PlatformWatchdogServiceCapabilities

399 When the CurrentTimerValue property is not supported, an instance of
400 CIM_PlatformWatchdogServiceCapabilities may be associated with the instance of
401 CIM_PlatformWatchdogService through an instance of CIM_ElementCapabilities.

402 7.1.8.2.1.1 CIM_PlatformWatchdogServiceCapabilities.WatchdogTimerDataSupported

403 The WatchdogTimerDataSupported array property shall not contain a value of 2 (Current Value).

404 7.1.8.2.2 CIM_PlatformWatchdogService.CurrentTimerValue

405 The value of the CIM_PlatformWatchdogService.CurrentTimerValue property shall be irrelevant in this
406 context.

407 7.1.9 Time of Last Expiration (Optional)

408 Support for the CIM_PlatformWatchdogService.TimeOfLastExpiration property is optional. This subclause
409 describes the CIM elements and behaviors that allow the client to determine whether the
410 TimeOfLastExpiration property is supported.

411 7.1.9.1 TimeOfLastExpiration Is Supported — Conditional

412 This subclause describes the CIM elements and behaviors that shall be implemented when the
413 TimeOfLastExpiration property is supported.

414 7.1.9.1.1 CIM_PlatformWatchdogServiceCapabilities

415 When the TimeOfLastExpiration property is supported, exactly one instance of
416 CIM_PlatformWatchdogServiceCapabilities shall be associated with the instance of
417 CIM_PlatformWatchdogService through an instance of CIM_ElementCapabilities.

418 7.1.9.1.1.1 CIM_PlatformWatchdogServiceCapabilities.LastExpirationDataSupported

419 The LastExpirationDataSupported array property shall contain a value of 2 (Time).

420 7.1.9.1.2 CIM_PlatformWatchdogService.TimeOfLastExpiration

421 The TimeOfLastExpiration property shall use the interval notation of the datetime type.

422 When the Watchdog has expired, the TimeOfLastExpiration property shall have the value of the time
423 when the watchdog timer last expired. Otherwise, the value of the TimeOfLastExpiration property may be
424 indeterminate.

425 7.1.9.2 TimeOfLastExpiration Is Not Supported

426 This subclause describes the CIM elements and behaviors that shall be implemented when the
427 TimeOfLastExpiration property is not supported.

428 7.1.9.2.1 CIM_PlatformWatchdogServiceCapabilities

429 When the TimeOfLastExpiration property is not supported, an instance of
430 CIM_PlatformWatchdogServiceCapabilities may be associated with the instance of
431 CIM_PlatformWatchdogService through an instance of CIM_ElementCapabilities.

432 7.1.9.2.1.1 CIM_PlatformWatchdogServiceCapabilities.LastExpirationDataSupported

433 The LastExpirationDataSupported array property shall not contain a value of 2 (Time).

434 7.1.9.2.2 CIM_PlatformWatchdogService.TimeOfLastExpiration

435 The value of the CIM_PlatformWatchdogService.TimeOfLastExpiration property shall be irrelevant in this
436 context.

437 7.1.10 Action on Expiration of the Watchdog Timer (Optional)

438 The implementation may support performing an action when the watchdog timer expires.

439 This subclause describes the CIM elements and behavior required to determine whether an
440 implementation supports performing an action upon expiration.

441 7.1.10.1 Action on Expiration Is Supported — Conditional

442 This subclause describes the CIM elements and behavior requirements when an implementation supports
443 performing an action upon expiration of the watchdog timer.

444 7.1.10.1.1 CIM_PlatformWatchdogServiceCapabilities

445 When the CIM_PlatformWatchdogService.ActionOnExpiration property is supported, exactly one instance
446 of CIM_PlatformWatchdogServiceCapabilities shall be associated with the instance of
447 CIM_PlatformWatchdogService through an instance of CIM_ElementCapabilities.

448 7.1.10.1.1.1 CIM_PlatformWatchdogServiceCapabilities.ActionOnExpirationSupported

449 The ActionOnExpirationSupported array property shall contain one or more values specifying the actions
450 that are supported when the watchdog timer expires.

451 7.1.10.1.2 CIM_PlatformWatchdogService.ActionOnExpiration

452 The ActionOnExpiration property shall specify the action that is performed when the watchdog timer
453 expires. The value shall be one or more of the values contained in the
454 CIM_PlatformWatchdogServiceCapabilities.ActionOnExpirationSupported property.

455 **7.1.10.2 ActionOnExpiration Is Not Supported**

456 This subclause describes the CIM elements and behaviors that shall be implemented when the
457 ActionOnExpiration property is not supported.

458 **7.1.10.2.1 CIM_PlatformWatchdogServiceCapabilities**

459 When the ActionOnExpiration property is not supported, an instance of
460 CIM_PlatformWatchdogServiceCapabilities may be associated with the instance of
461 CIM_PlatformWatchdogService through an instance of CIM_ElementCapabilities.

462 **7.1.10.2.1.1 CIM_PlatformWatchdogServiceCapabilities.ActionOnExpirationSupported**

463 The ActionOnExpirationSupported array property shall not contain any values.

464 **7.1.10.2.2 CIM_PlatformWatchdogService.ActionOnExpiration**

465 The value of the CIM_PlatformWatchdogService.ActionOnExpiration property shall be irrelevant in this
466 context.

467 **7.1.11 CIM_PlatformWatchdogService.ElementName**

468 The ElementName property shall be formatted as a free-form string of variable length (pattern ".**").

469 The ElementName property may support being modified by the ModifyInstance operation (see 8.9.1.1).
470 This behavior is conditional. This subclause describes the CIM elements and behavior required to
471 determine whether an implementation supports client modification of the ElementName property.

472 **7.1.11.1 Modifying ElementName Is Supported — Conditional**

473 This subclause describes the CIM elements and behavior requirements when an implementation supports
474 client modification of the CIM_PlatformWatchdogService.ElementName property.

475 **7.1.11.1.1 CIM_PlatformWatchdogServiceCapabilities**

476 When client modification of the CIM_PlatformWatchdogService.ElementName property is supported,
477 exactly one instance of CIM_PlatformWatchdogServiceCapabilities shall be associated with the instance
478 of CIM_PlatformWatchdogService through an instance of CIM_ElementCapabilities.

479 **7.1.11.1.1.1 CIM_PlatformWatchdogServiceCapabilities.ElementNameSupported**

480 The CIM_PlatformWatchdogServiceCapabilities.ElementNameSupported property shall have a value of
481 TRUE.

482 **7.1.11.1.1.2 CIM_PlatformWatchdogServiceCapabilities.MaxElementNameLen**

483 The CIM_PlatformWatchdogServiceCapabilities.MaxElementNameLen property shall be implemented.

484 **7.1.11.2 Modifying ElementName Is Not Supported**

485 This subclause describes the CIM elements and behaviors that shall be implemented when the
486 CIM_PlatformWatchdogService.ElementName property does not support being modified by the
487 ModifyInstance operation.

488 **7.1.11.2.1 CIM_PlatformWatchdogServiceCapabilities**

489 When client modification of the CIM_PlatformWatchdogService.ElementName property is not supported,
490 an instance of CIM_PlatformWatchdogServiceCapabilities may be associated with the instance of
491 CIM_PlatformWatchdogService through an instance of CIM_ElementCapabilities.

492 7.1.11.2.1.1 CIM_PlatformWatchdogServiceCapabilities.ElementNameSupported

493 The CIM_PlatformWatchdogServiceCapabilities.ElementNameSupported property shall have a value of
494 FALSE.

495 7.1.11.2.1.2 CIM_PlatformWatchdogServiceCapabilities.MaxElementNameLen

496 The CIM_PlatformWatchdogServiceCapabilities.MaxElementNameLen property is irrelevant in this
497 context.

498 7.2 Representing the Watchdog Device (Optional)

499 The instance of a concrete subclass of CIM_LogicalDevice may be used to represent the device that
500 implements a Watchdog.

501 The instance of a concrete subclass of CIM_LogicalDevice shall be associated with the instance of
502 CIM_PlatformWatchdogService through the CIM_DeviceServiceImplementation association.

503 The CIM_DeviceServiceImplementation association's Antecedent property shall reference the instance of
504 a concrete subclass of CIM_LogicalDevice, and its Dependent property shall reference the
505 CIM_PlatformWatchdogService instance.

506 7.3 Representing the Monitored Entity (Optional)

507 An instance of a concrete subclass of CIM_LogicalElement may be used to represent the entity that is
508 monitored by the Watchdog.

509 When an instance of a concrete subclass of CIM_LogicalElement exists that represents the monitored
510 entity, an instance of CIM_ServiceAvailableToElement shall be used between the instance of
511 CIM_PlatformWatchdogService and the instance of a concrete subclass of CIM_LogicalElement.

512 The CIM_ServiceAvailableToElement association's UserOfService property shall reference the instance
513 of a concrete subclass of CIM_LogicalElement, and its ServiceProvided property shall reference the
514 instance of CIM_PlatformWatchdogService.

515 7.4 Representing the Entity on Which Action on Expiration Is Taken (Optional)

516 An instance of a concrete subclass of CIM_LogicalElement may be used to represent the entity on which
517 action on expiration is taken by the CIM_PlatformWatchdogService instance.

518 When an instance of a subclass of CIM_LogicalElement exists that represents the monitored entity on
519 which action is taken, an instance of CIM_ServiceAffectsElement shall be used between the instance of
520 CIM_PlatformWatchdogService and the instance of a concrete subclass of CIM_LogicalElement.

521 The CIM_ServiceAffectsElement association's AffectedElement property shall reference the instance of a
522 concrete subclass of CIM_LogicalElement, and its AffectingElement property shall reference the instance
523 of CIM_PlatformWatchdogService.

524 7.5 State Management of a Watchdog (Optional)

525 State management of the Watchdog is optional behavior. This clause describes the CIM elements and
526 behaviors that allow the client to determine whether state management of the Watchdog is supported.

527 7.5.1 Watchdog State Management Is Supported — Conditional

528 This subclause describes the CIM elements and behaviors that shall be implemented when state
529 management of the Watchdog is supported.

530 7.5.1.1 CIM_PlatformWatchdogServiceCapabilities

531 When state management of the Watchdog is supported, exactly one instance of
532 CIM_PlatformWatchdogServiceCapabilities shall be associated with the instance of
533 CIM_PlatformWatchdogService through an instance of CIM_ElementCapabilities.

534 7.5.1.2 CIM_PlatformWatchdogServiceCapabilities.RequestedStatesSupported

535 The CIM_PlatformWatchdogServiceCapabilities.RequestedStatesSupported property shall contain the
536 following values: 2 (Enabled), 3 (Disabled), and 6 (Offline).

537 7.5.1.3 CIM_PlatformWatchdogService.RequestedState

538 When the CIM_PlatformWatchdogService.RequestStateChange() method is successfully invoked, the
539 value of the RequestedState property shall be the value of the RequestedState parameter. If the method
540 is not successfully invoked, the value of the RequestedState property is indeterminate.

541 The CIM_PlatformWatchdogService.RequestedState property shall have one of the values specified in
542 the CIM_PlatformWatchdogServiceCapabilities.RequestedStatesSupported property or a value of 5 (No
543 Change).

544 7.5.1.4 CIM_PlatformWatchdogService.EnabledState

545 When the RequestedState parameter has a value of 2 (Enabled), 3 (Disabled), or 6 (Offline) and the
546 CIM_PlatformWatchdogService.RequestStateChange() method completes successfully, the value of the
547 EnabledState property shall equal the value of the CIM_PlatformWatchdogService.RequestedState
548 property.

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

550 7.5.2 Watchdog State Management Is Not Supported

551 This subclause describes the CIM elements and behaviors that shall be implemented when state
552 management of the Watchdog is not supported.

553 7.5.2.1.1 CIM_PlatformWatchdogServiceCapabilities

554 When state management of the Watchdog is not supported, an instance of
555 CIM_PlatformWatchdogServiceCapabilities may be associated with the instance of
556 CIM_PlatformWatchdogService through an instance of CIM_ElementCapabilities.

557 7.5.2.2 CIM_PlatformWatchdogServiceCapabilities.RequestedStatesSupported

558 The CIM_PlatformWatchdogServiceCapabilities.RequestedStatesSupported property shall not contain
559 any values.

560 7.5.2.3 CIM_PlatformWatchdogService.RequestedState

561 The RequestedState property shall have a value of 12 (Not Applicable).

562 7.5.2.4 CIM_PlatformWatchdogService.EnabledState

563 The EnabledState property shall have one of the following values: 2 (Enabled), 3 (Disabled), 5 (Not
564 Applicable), or 6 (Enabled but Offline). The value 5 (Not Applicable) may be set when non-CIM
565 instrumentation has manipulated the instance of CIM_PlatformWatchdogService.

566 8 Methods

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

569 8.1 CIM_PlatformWatchdogService.RequestStateChange()

570 Invocation of the RequestStateChange() method changes the element's state to the value specified in the
571 RequestedState parameter. A value of 2 (Enabled) shall correspond to a request to place the platform
572 watchdog service in an enabled state. A value of 3 (Disabled) shall correspond to a request to place the
573 platform watchdog service in a disabled state and clear the platform watchdog service configuration. A
574 value of 6 (Offline) shall correspond to a request to place the platform watchdog service into an "Enabled
575 but Offline" state.

576 When the RequestedState parameter has the value 2 (Enabled), the method may return the value 2 if the
577 platform watchdog service is not properly configured.

578 The method shall be considered successful when, upon completion of the method, the resultant state is
579 equal to the requested state. An actual change in state does not need to occur for the method to be
580 considered successful.

581 Return values for RequestStateChange() shall be as specified in Table 2 where the method-execution
582 behavior matches the return-code description. RequestStateChange() method's parameters are specified
583 in Table 3.

584 No standard messages are defined for this method.

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

587 **Table 2 – CIM_PlatformWatchdogService.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

588 **Table 3 – CIM_PlatformWatchdogService.RequestStateChange() Method: Parameters**

Qualifiers	Name	Type	Description/Values
IN	RequestedState	uint16	Valid state values: 2 (Enabled) 3 (Disabled) 6 (Offline)
OUT	Job	CIM_ConcreteJob REF	Returned if job started
IN	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

589 **8.1.1 General Requirements**

590 If the RequestedState parameter is NULL, the CIM_PlatformWatchdogService.RequestStateChange()
591 method shall return a value of 2 (Unknown or Unspecified Error).

592 The CIM_PlatformWatchdogService.RequestStateChange() method shall return a value of 2 (Unknown
593 or Unspecified Error) if the RequestedState parameter specifies a value that is not listed in the
594 CIM_PlatformWatchdogServiceCapabilities.RequestedStatesSupported property of the associated
595 instance of CIM_PlatformWatchdogServiceCapabilities.

596 The CIM_PlatformWatchdogService.RequestStateChange() method shall return a value of 0 (Completed
597 with No Error) if the state change completed synchronous with the method invocation. The
598 CIM_PlatformWatchdogService.RequestStateChange() method may return a value of 0 (Completed with
599 No Error) if the state change was initiated synchronously with the method invocation and the state
600 transition has not completed.

601 **8.1.2 Conditional Requirement**

602 If the behavior specified in 7.5.1 is implemented, the
603 CIM_PlatformWatchdogService.RequestStateChange() method shall be implemented and shall not
604 return a value of 1 (Not Supported).

605 **8.2 Profile Conventions for Operations**

606 Support for operations for each profile class (including associations) is specified in the following
607 subclauses. Each subclause includes a table listing all the operations supported by this profile. Compliant
608 implementations of this profile shall support all these operations.

609 **8.3 CIM_DeviceServiceImplementation Operations**

610 Compliant implementations of this profile shall support the operations listed in Table 4 for the
611 CIM_DeviceServiceImplementation class. Each operation shall be supported as defined in [DSP0200](#).

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

613 **Table 4 – Operations: CIM_DeviceServiceImplementation**

Operation	Requirement	Messages
GetInstance	Mandatory	None
EnumerateInstances	Mandatory	None
EnumerateInstanceNames	Mandatory	None

614 **8.4 CIM_ElementCapabilities Operations**

615 Compliant implementations of this profile shall support the operations listed in Table 5 for the
616 CIM_ElementCapabilities class. Each operation shall be supported as defined in [DSP0200](#).

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

618

Table 5 – Operations: CIM_ElementCapabilities

Operation	Requirement	Messages
GetInstance	Mandatory	None
EnumerateInstances	Mandatory	None
EnumerateInstanceNames	Mandatory	None

619 8.5 CIM_HostedService Operations

620 Compliant implementations of this profile shall support the operations listed in Table 6 for the
621 CIM_HostedService class. Each operation shall be supported as defined in [DSP0200](#).

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

623

Table 6 – Operations: CIM_HostedService

Operation	Requirement	Messages
GetInstance	Mandatory	None
EnumerateInstances	Mandatory	None
EnumerateInstanceNames	Mandatory	None

624 8.6 CIM_ServiceAffectsElement Operations

625 Compliant implementations of this profile shall support the operations listed in Table 7 for the
626 CIM_ServiceAffectsElement class. Each operation shall be supported as defined in [DSP0200](#).

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

628

Table 7 – Operations: CIM_ServiceAffectsElement

Operation	Requirement	Messages
GetInstance	Mandatory	None
EnumerateInstances	Mandatory	None
EnumerateInstanceNames	Mandatory	None

629 8.7 CIM_ServiceAvailableToElement Operations

630 Compliant implementations of this profile shall support the operations listed in Table 8 for the
631 CIM_ServiceAvailableToElement class. Each operation shall be supported as defined in [DSP0200](#).

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

633

Table 8 – Operations: CIM_ServiceAvailableToElement

Operation	Requirement	Messages
GetInstance	Mandatory	None
EnumerateInstances	Mandatory	None
EnumerateInstanceNames	Mandatory	None

634 **8.8 CIM_LogicalDevice Operations**

635 Compliant implementations of this profile shall support the operations listed in Table 9 for
 636 CIM_LogicalDevice. Each operation shall be supported as defined in [DSP0200](#).

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

638 **Table 9 – Operations: CIM_LogicalDevice**

Operation	Requirement	Messages
GetInstance	Mandatory	None
EnumerateInstances	Mandatory	None
EnumerateInstanceNames	Mandatory	None

639 **8.9 CIM_PlatformWatchdogService Operations**

640 Compliant implementations of this profile shall support the operations listed in Table 10 for
 641 CIM_PlatformWatchdogService. Each operation shall be supported as defined in [DSP0200](#).

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

643 **Table 10 – Operations: CIM_PlatformWatchdogService**

Operation	Requirement	Messages
GetInstance	Mandatory	None
ModifyInstance	Optional	See 8.9.1.
EnumerateInstances	Mandatory	None
EnumerateInstanceNames	Mandatory	None

644 **8.9.1 CIM_PlatformWatchdogService — ModifyInstance Operation**

645 This subclause details the specific requirements for the ModifyInstance operation applied to an instance
 646 of CIM_PlatformWatchdogService.

647 **8.9.1.1 CIM_PlatformWatchdogService.ElementName Property**

648 When the CIM_PlatformWatchdogServiceCapabilities.ElementNameEditSupported property has a value
 649 of TRUE, the implementation shall allow the ModifyInstance operation to change the value of the
 650 ElementName property of the CIM_PlatformWatchdogService instance. The ModifyInstance operation
 651 shall enforce the length restriction specified in the MaxElementNameLen property of the
 652 CIM_PlatformWatchdogServiceCapabilities instance.

653 When the CIM_PlatformWatchdogServiceCapabilities.ElementNameEditSupported property has a value
 654 of FALSE, the implementation shall not allow the ModifyInstance operation to change the value of the
 655 ElementName property of the CIM_PlatformWatchdogService instance.

656 **8.10 CIM_PlatformWatchdogServiceCapabilities Operations**

657 Compliant implementations of this profile shall support the operations listed in Table 11 for the
 658 CIM_PlatformWatchdogServiceCapabilities class. Each operation shall be supported as defined in
 659 [DSP0200](#).

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

661

Table 11 – Operations: CIM_PlatformWatchdogServiceCapabilities

Operation	Requirement	Messages
GetInstance	Mandatory	None
EnumerateInstances	Mandatory	None
EnumerateInstanceNames	Mandatory	None

662 9 Use Cases (Informative)

663 This clause contains object diagrams and use cases specific to the *Platform Watchdog Profile*. The use
664 cases are informative and are not intended to define the requirements for conformance.

665 9.1 Advertising the Profile Conformance

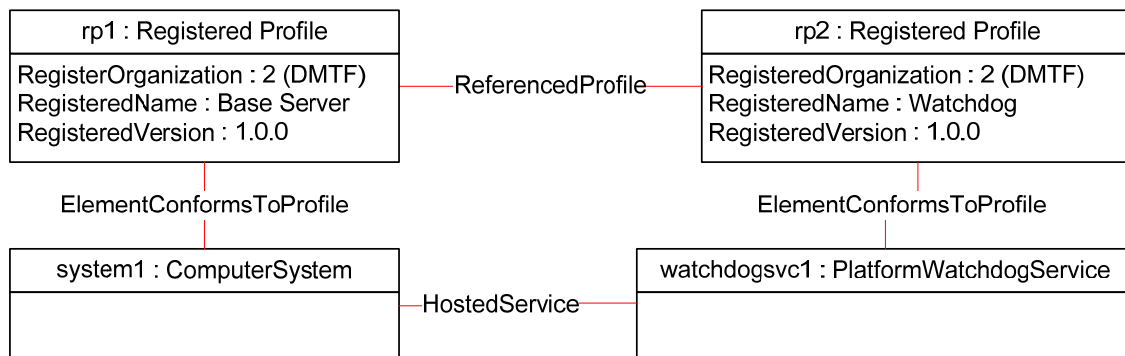
666 The object diagram in Figure 3 shows how instances of CIM_RegisteredProfile are used to identify the
667 version of the *Platform Watchdog Profile* with which an instance of CIM_PlatformWatchdogService and its
668 associated instances are conformant.

669 An instance of CIM_RegisteredProfile exists for each profile that is instrumented in the system. One
670 instance of CIM_RegisteredProfile identifies the DMTF *Base Server Profile*, version 1.0.0. The other
671 instance identifies the DMTF *Platform Watchdog Profile*, version 1.0.0. The Central Instance is the
672 CIM_PlatformWatchdogService instance. The Scoping Instance is the CIM_ComputerSystem instance.

673 This instance of CIM_ComputerSystem is conformant with the DMTF *Base Server Profile*, version 1.0.0,
674 as indicated by the CIM_ElementConformsToProfile association to the CIM_RegisteredProfile instance,
675 rp1.

676 This instance of CIM_PlatformWatchdogService is conformant with the DMTF *Platform Watchdog Profile*,
677 version 1.0.0, as indicated by the CIM_ElementConformsToProfile association to the
678 CIM_RegisteredProfile instance, rp2.

679

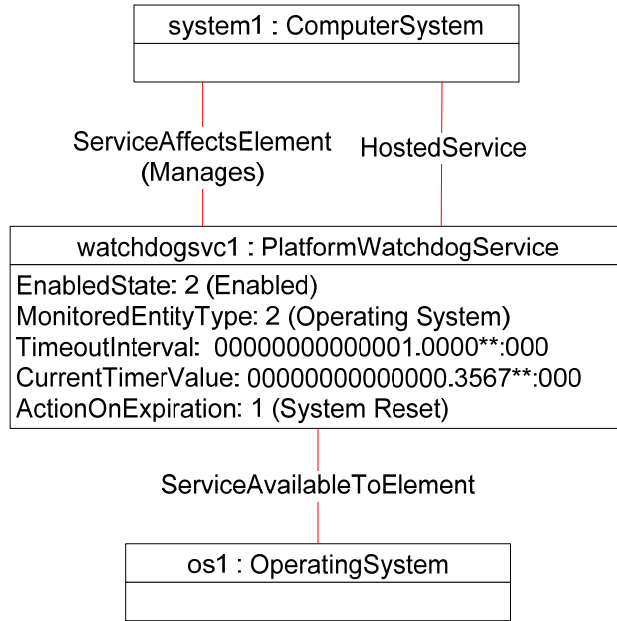


680

Figure 3 – Registered Profile

681 9.2 Object Diagram for a Monolithic Server

682 Figure 4 shows the object diagram for a monolithic server (system1) hosting a Watchdog (watchdogsvc1)
683 that is configured to monitor the operating system (os1). If the watchdog timer expires, the Watchdog will
684 reset system1.



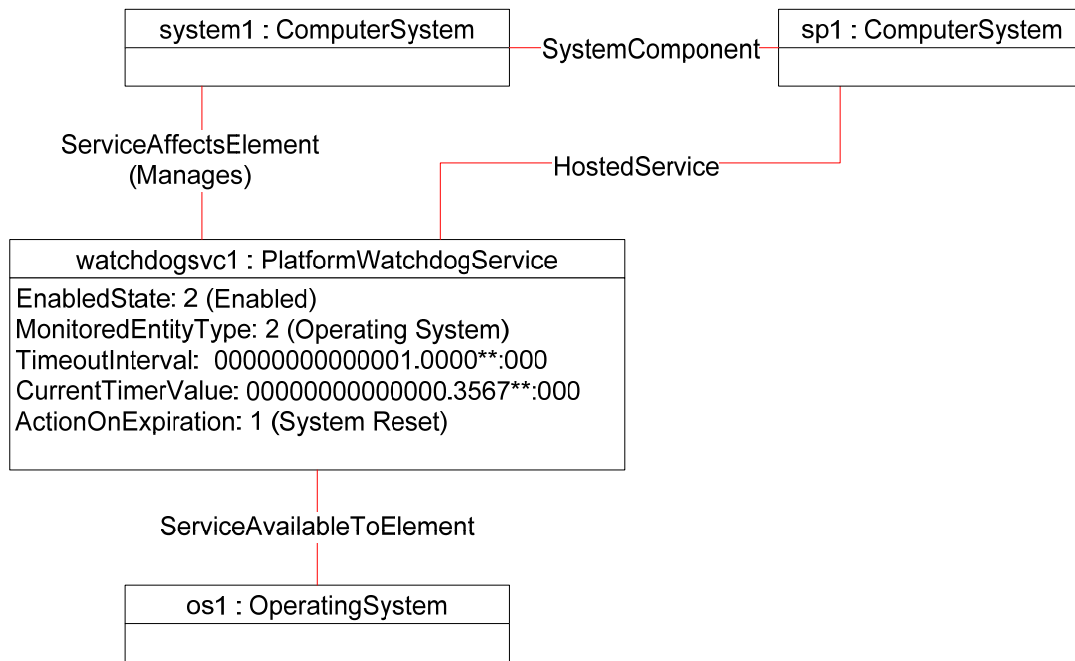
685

686

Figure 4 – Object Diagram: Monolithic Server

687 **9.3 Object Diagram for a Monolithic Server with a Service Processor**

688 Figure 5 shows the object diagram for a monolithic server (system1) with a service processor (sp1). The
689 service processor is hosting a Watchdog (watchdogsvc1) that is configured to monitor the operating
690 system (os1). If the watchdog timer expires, the Watchdog will reset system1.



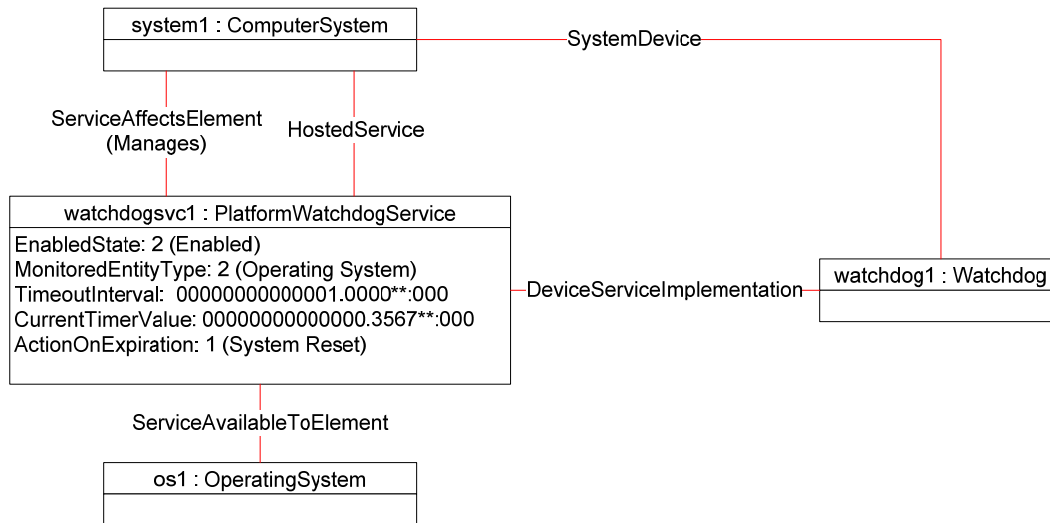
691

692

Figure 5 – Object Diagram: Monolithic Server with a Service Processor

693 **9.4 Object Diagram for a Monolithic System with a Watchdog Device**

694 Figure 6 shows the object diagram for a monolithic server (system1) hosting a Watchdog (watchdogsvc1)
 695 that is implemented as a device (watchdog1) and configured to monitor the operating system (os1). If the
 696 watchdog timer expires, the Watchdog will reset system1.

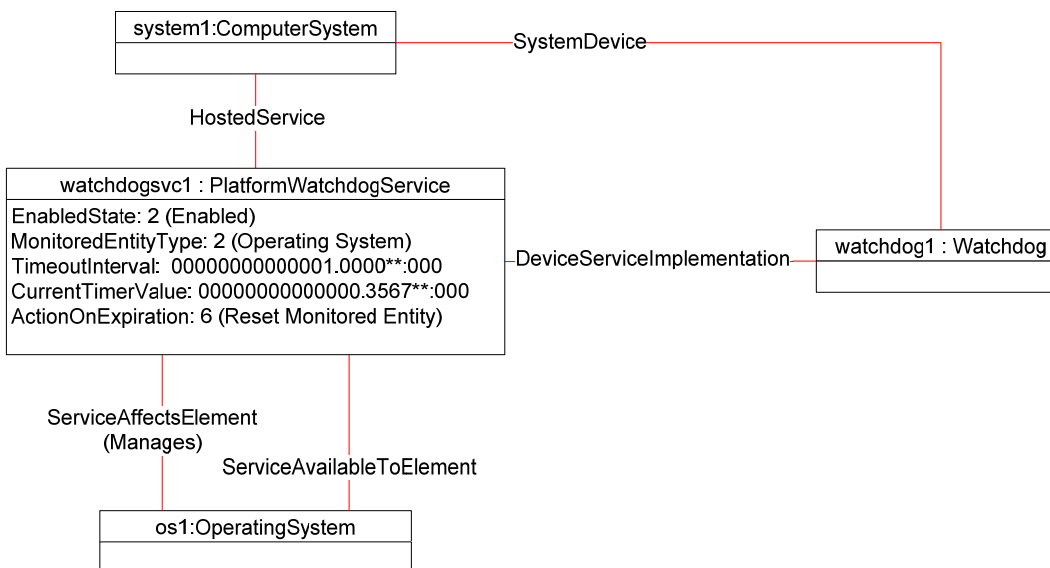


697

698 **Figure 6 – Object Diagram: Reset System When a Watchdog Expires**

699 **9.5 Object Diagram for OS Reset When a Watchdog Timer Expires**

700 Figure 7 shows the object diagram for a monolithic server (system1) hosting a Watchdog (watchdogsvc1)
 701 that is implemented as a device (watchdog1) and configured to monitor the operating system (os1). If the
 702 watchdog timer expires, the Watchdog will reset os1.



703

704 **Figure 7 – Object Diagram: Reset OS When a Watchdog Timer Expires**

705 **9.6 Object Diagram for System with Multiple Watchdogs**

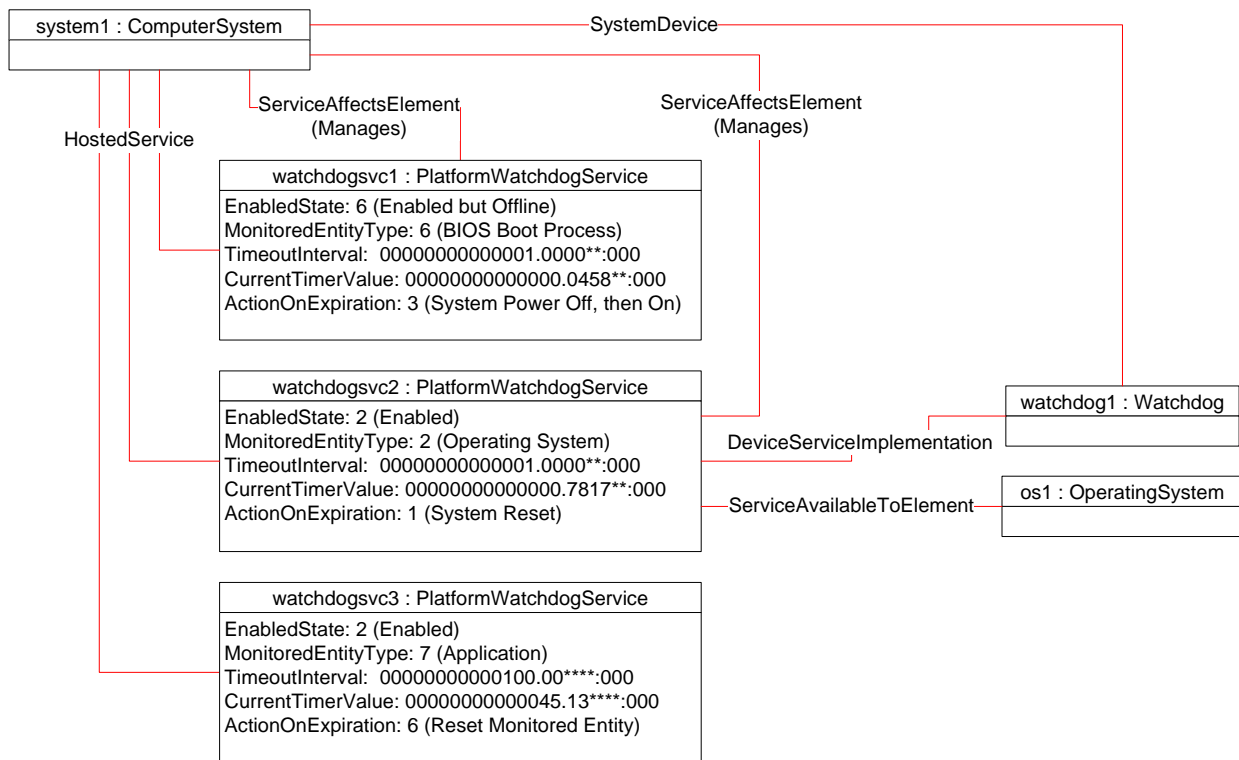
706 Figure 8 shows the object diagram for a monolithic server (system1) hosting three Watchdogs
 707 (watchdogsvc1, watchdogsvc2, and watchdogsvc3).

708 watchdogsvc1 monitors the BIOS boot process and will power cycle the system if its watchdog timer
 709 expires. watchdogsvc2 monitors the operating system and will reset the system if its watchdog timer
 710 expires. watchdogsvc3 monitors an application process and will reset (restart) the application if its
 711 watchdog timer expires.

712 No instances of a concrete subclass of CIM_LogicalElement represent the application or the BIOS boot
 713 process.

714 The states of the Watchdogs are for a system in its lifecycle after the OS has booted and an application
 715 has been started. The Watchdog represented by watchdogsvc1 is inactive because the phase when the
 716 BIOS boots has completed. The CurrentTimerValue property contains the value when the Watchdog was
 717 deactivated. Being inactive, as opposed to disabled, this Watchdog could be automatically rearmed when
 718 the system is powered on again.

719 The Watchdogs represented by watchdogsvc2 and watchdogsvc3 are both active, which means that the
 720 OS and an application are being monitored simultaneously. The TimeoutInterval for the OS is 1 second,
 721 and the TimeoutInterval for the application is 100 seconds.



722

723 **Figure 8 – Object Diagram: System with Multiple Watchdogs**

724 9.7 Representing the Watchdog States

725 Figure 9 shows how the Watchdog states are represented by the CIM_PlatformWatchdogService
 726 properties. The discussion starts at the top left corner of the figure. To illustrate the behavior of the
 727 properties, the use case assumes that timer resolution, time of last expiration, and current timer value are
 728 supported.

729 The CIM_PlatformWatchdogService instance at the upper left represents a configured Watchdog. Its
 730 EnabledState property shall have the value 6 (Enabled but Offline). The MonitoredEntityType,
 731 TimeoutInterval, and ActionOnExpiration properties have been set so the Watchdog can be successfully
 732 activated. The activation can occur through an invocation of the RequestStateChange() method or
 733 automatically during power-on.

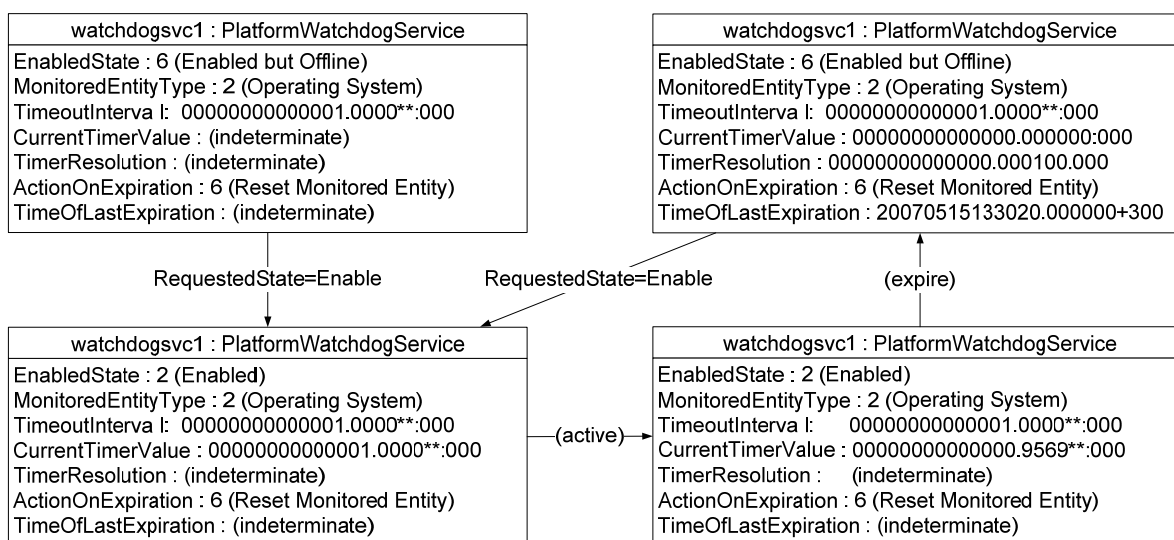
734 The CIM_PlatformWatchdogService instance at the lower left represents a Watchdog immediately after
 735 activation. Its EnabledState property shall have the value 2 (Enabled). The CurrentTimerValue property
 736 value matches the value of the TimeoutInterval property.

737 The CIM_PlatformWatchdogService instance at the lower right represents a Watchdog some time after it
 738 has been activated. The CurrentTimerValue property shall have a value between the TimeoutInterval
 739 value and the 0 (0000000000000.000000:000).

740 The CIM_PlatformWatchdogService instance at the upper right represents a Watchdog after expiration.
 741 Its EnabledState property shall have the value 6 (Enabled but Offline), reflecting the expiration of the
 742 Watchdog. The CurrentTimerValue property shall have a value of 0. The TimerResolution property shall
 743 have a meaningful value. The TimeOfLastExpiration property shall have a value that represents the time
 744 the watchdog timer expired.

745 The CIM_PlatformWatchdogService instance at the upper right could be replaced with an instance whose
 746 EnabledState property has a value of 3 (Disabled), which also reflects the expiration of the Watchdog.
 747 The value of other properties would be the same.

748 From either the Enabled but Offline or Disabled state, the Watchdog is ready to be activated again. Note
 749 that activating the Watchdog will make the values of the TimeOfLastExpiration and TimerResolution
 750 properties indeterminate.



751

752

Figure 9 – CIM_PlatformWatchdogService Property Values per Watchdog State

753 **9.8 Finding the Watchdogs Hosted on a Computer System**

754 A client can determine the Watchdog on a computer system of interest as follows:

- 755 1) Start at the instance of CIM_ComputerSystem that represents the computer system of interest.
- 756 2) Enumerate the instances of the CIM_PlatformWatchdogService that are associated with the
757 CIM_ComputerSystem instance through an instance of the CIM_HostedService association.

758 Each instance of CIM_PlatformWatchdogService thus found is a Watchdog hosted on the
759 computer system of interest.

760 **9.9 Finding the Watchdogs Monitoring an Entity Type**

761 A client can determine the Watchdogs that monitor an entity type of interest as follows:

- 762 1) Enumerate instances of CIM_PlatformWatchdogService.
- 763 2) For each instance of CIM_PlatformWatchdogService, inspect the MonitoredEntityType property
764 for the type of the monitored entity of interest.

765 Each instance of CIM_PlatformWatchdogService thus found is a Watchdog that monitors the
766 entity type of interest.

767 **9.10 Finding the Watchdogs Monitoring an Entity Instance**

768 A client can determine the Watchdogs that are monitoring an entity of interest when an instance of a
769 concrete subclass of CIM_LogicalElement that represents the monitored entity of interest exists, as
770 follows:

- 771 1) Start at the instance of a concrete subclass of CIM_LogicalElement of interest.
- 772 2) Enumerate the instances of the CIM_PlatformWatchdogService that are associated with the
773 instance of a concrete subclass of CIM_LogicalElement through an instance of the
774 CIM_ServiceAvailableToElement association.

775 Each instance of CIM_PlatformWatchdogService thus found is a Watchdog for the instance of a
776 concrete subclass of CIM_LogicalElement of interest.

777 **9.11 Determining Whether a Watchdog Supports State Management**

778 A client can determine whether a Watchdog supports state management as follows:

- 779 1) Start at the instance of CIM_PlatformWatchdogService that represents the Watchdog of
780 interest.
- 781 2) Get the associated CIM_PlatformWatchdogServiceCapabilities instance by traversing the
782 CIM_ElementCapabilities association.

783 If no CIM_PlatformWatchdogServiceCapabilities instance is returned, state management is not
784 supported.

- 785 3) Query the value of the RequestedStatesSupported property array.

786 If the RequestedStatesSupported property array contains no values, the
787 CIM_PlatformWatchdogService does not support state management.

788 9.12 Activating a Watchdog

789 A client can activate a Watchdog of interest as follows:

- 790 1) Start at the instance of the CIM_PlatformWatchdogService that represents the Watchdog of
791 interest.
- 792 2) Query the value of the CIM_PlatformWatchdogService.EnabledState property.
- 793 3) If the value of the CIM_PlatformWatchdogService.EnabledState property is 3 (Disabled) or 6
794 (Enabled but Offline), invoke the RequestStateChange() method with the RequestedState
795 parameter set to 2 (Enabled).
- 796 4) Verify that the CIM_PlatformWatchdogService.EnabledState property has the value of 2
797 (Enabled).

798 The Watchdog should now be active.

799 9.13 Obtaining Information Regarding the Last Watchdog Expiration

800 A client can obtain the information about the Last Watchdog Expiration as follows:

- 801 1) Start at the instance of CIM_PlatformWatchdogService of interest by using the use case in 9.8
802 or 9.9.
- 803 2) Get the associated CIM_PlatformWatchdogServiceCapabilities instance by traversing the
804 CIM_ElementCapabilities association.

805 If no CIM_PlatformWatchdogServiceCapabilities instance is returned, TimeOfLastExpiration is
806 not supported.
- 807 3) Query the value of the TimeOfLastExpirationSupported property.

808 If the value is TRUE, the CIM_PlatformWatchdogService.TimeOfLastExpiration property
809 contains valid information regarding the last expiration.

810 9.14 Determining Whether CIM_PlatformWatchdogService.ElementName Can Be 811 Modified

812 A client can determine whether the ElementName can be modified as follows:

- 813 1) Start at the instance of CIM_PlatformWatchdogService.
- 814 2) Get the associated CIM_PlatformWatchdogServiceCapabilities instance by traversing the
815 CIM_ElementCapabilities association.

816 If no CIM_PlatformWatchdogServiceCapabilities instance is returned, client modification of
817 ElementName is not supported.
- 818 3) Query the value of the ElementNameEditSupported property of the instance.

819 If the value is TRUE, the CIM_PlatformWatchdogService.ElementName property can be
820 modified by a client.

821 10 CIM Elements

822 This clause lists the required properties and methods for each class required for this profile. Clauses 7
823 (“Implementation”) and 8 (“Methods”) may impose additional requirements on these elements.

824 Table 12 lists the CIM elements that are required for this profile. The subsequent subclauses describe
825 CIM elements for which additional normative statements can be made.

826

Table 12 – CIM Elements: Platform Watchdog Profile

Element Name	Requirement	Description
CIM_RegisteredProfile	Mandatory	See 10.1.
CIM_DeviceServiceImplementation	Conditional	See 10.2.
CIM_ElementCapabilities	Conditional	See 10.3.
CIM_HostedService	Mandatory	See 10.4.
CIM_ServiceAffectsElement	Optional	Referencing CIM_ComputerSystem. See 10.5.
CIM_ServiceAffectsElement	Optional	Referencing a concrete subclass of CIM_LogicalElement. See 10.6.
CIM_ServiceAvailableToElement	Optional	See 10.7.
CIM_LogicalDevice	Optional	See 10.8.
CIM_PlatformWatchdogService	Mandatory	See 10.9.
CIM_PlatformWatchdogServiceCapabilities	Optional	See 10.10.

827 **10.1 CIM_RegisteredProfile**

828 CIM_RegisteredProfile identifies the *Platform Watchdog Profile* in order for a client to determine whether
 829 an instance of CIM_ComputerSystem is conformant with this profile. The CIM_RegisteredProfile class is
 830 defined by the [Profile Registration Profile](#). With the exception of the mandatory values specified for the
 831 properties in Table 13, the behavior of the CIM_RegisteredProfile instance is per the [Profile Registration](#)
 832 [Profile](#).

833

Table 13 – Class: CIM_RegisteredProfile

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

834 **10.2 CIM_DeviceServiceImplementation**

835 The CIM_DeviceServiceImplementation association is used to relate an instance of a concrete subclass
 836 of CIM_LogicalDevice with the instance of CIM_PlatformWatchdogService. Table 14 contains the
 837 requirements for elements of this class.

838

Table 14 – Class: CIM_DeviceServiceImplementation

Elements	Requirement	Notes
Antecedent	Mandatory	This property shall be a reference to an instance of a concrete subclass of CIM_LogicalDevice. See 7.2. Cardinality is "**".
Dependent	Mandatory	This property shall be a reference to an instance of CIM_PlatformWatchdogService. See 7.2. Cardinality is "**".

839 10.3 CIM_ElementCapabilities

840 The CIM_ElementCapabilities association is used to relate an instance of
 841 CIM_PlatformWatchdogServiceCapabilities with the instance of CIM_PlatformWatchdogService. Table 15
 842 contains the requirements for elements of this class.

843 **Table 15 – Class: CIM_ElementCapabilities**

Elements	Requirement	Notes
ManagedElement	Mandatory	This property shall be a reference to an instance of CIM_PlatformWatchdogService. See 7.1.1. Cardinality is "1..*".
Capabilities	Mandatory	This property shall be a reference to an instance of CIM_PlatformWatchdogServiceCapabilities. See 7.1.1. Cardinality is "0..1".

844 10.4 CIM_HostedService

845 The CIM_HostedService association is used to relate the CIM_PlatformWatchdogService to the
 846 CIM_ComputerSystem on which it is hosted. Table 16 contains the requirements for elements of this
 847 class.

848 **Table 16 – Class: CIM_HostedService**

Elements	Requirement	Notes
Antecedent	Mandatory	This property shall be a reference to an instance of CIM_ComputerSystem. See 7.1.2. Cardinality is "1".
Dependent	Mandatory	This property shall be a reference to an instance of CIM_PlatformWatchdogService. See 7.1.2. Cardinality is "**".

849 10.5 CIM_ServiceAffectsElement, Relating CIM_PlatformWatchdogService to 850 CIM_ComputerSystem

851 The CIM_ServiceAffectsElement association is used to relate the instance of
 852 CIM_PlatformWatchdogService to the instance of CIM_ComputerSystem that is affected by an action
 853 upon the expiration of the watchdog timer. Table 17 contains the requirements for elements of this class.

854 **Table 17 – Class: CIM_ServiceAffectsElement Referencing CIM_ComputerSystem**

Elements	Requirement	Notes
AffectingElement	Mandatory	This property shall be a reference to an instance of CIM_PlatformWatchdogService. See 7.4. Cardinality is "0..1".
AffectedElement	Mandatory	This property shall be a reference to an instance of CIM_ComputerSystem. See 7.4. Cardinality is "0..1".

855 **10.6 CIM_ServiceAffectsElement, Relating CIM_PlatformWatchdogService to a**
 856 **Concrete Subclass of CIM_LogicalElement**

857 The CIM_ServiceAffectsElement association is used to relate the instance of
 858 CIM_PlatformWatchdogService to the instance of a concrete subclass of CIM_LogicalElement that
 859 represents the logical element on which action is taken upon expiration. Table 18 contains the
 860 requirements for elements of this class.

861 **Table 18 – Class: CIM_ServiceAffectsElement Referencing CIM_LogicalElement**

Elements	Requirement	Notes
AffectingElement	Mandatory	This property shall be a reference to an instance of CIM_PlatformWatchdogService. See 7.4. Cardinality is "0..1".
AffectedElement	Mandatory	This property shall be a reference to an instance of a concrete subclass of CIM_LogicalElement. See 7.4. Cardinality is "0..*".

862 **10.7 CIM_ServiceAvailableToElement**

863 The CIM_ServiceAvailableToElement association is used to relate the instance of
 864 CIM_PlatformWatchdogService to the instance of a concrete subclass of CIM_LogicalElement that
 865 represents the monitored entity. Table 19 contains the requirements for elements of this class.

866 **Table 19 – Class: CIM_ServiceAvailableToElement**

Elements	Requirement	Notes
ServiceProvided	Mandatory	This property shall be a reference to an instance of CIM_PlatformWatchdogService. See 7.3. Cardinality is "0..1".
UserOfService	Mandatory	This property shall be a reference to an instance of a concrete subclass of CIM_LogicalElement. See 7.3. Cardinality is "0..1".

867 **10.8 CIM_LogicalDevice**

868 The concrete subclass of the CIM_LogicalDevice class represents the device on the computer system
 869 that instantiates the Watchdog. Table 20 contains the requirements for elements of this class.

870 **Table 20 – Class: CIM_LogicalDevice**

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

871 **10.9 CIM_PlatformWatchdogService**

872 The CIM_PlatformWatchdogService class represents the ability to provide a Watchdog to the monitored
 873 entity on a system. Table 21 contains the requirements for elements of this class.

874 **Table 21 – Class: CIM_PlatformWatchdogService**

Elements	Requirement	Notes
SystemCreationClassName	Mandatory	Key
SystemName	Mandatory	Key
CreationClassName	Mandatory	Key
Name	Mandatory	Key
ElementName	Mandatory	See 7.1.11.
EnabledState	Mandatory	See 7.5.
CurrentTimerValue	Conditional	See 7.1.8.
MonitoredEntityType	Mandatory	See 7.3.
OtherMonitoredEntityType	Conditional	See 7.3.
RequestedState	Mandatory	See 7.5.
TimeOfLastExpiration	Conditional	See 7.1.9.
TimeoutInterval	Mandatory	See 7.1.4.
TimerExpired	Mandatory	See 7.1.5.
TimerResolution	Conditional	See 7.1.6.
ActionOnExpiration	Conditional	See 7.1.10.
RequestStateChange()	Conditional	See 8.1.

875 **10.10 CIM_PlatformWatchdogServiceCapabilities**

876 The CIM_PlatformWatchdogServiceCapabilities class represents the capabilities supported by a
 877 Watchdog. Table 22 contains the requirements for elements of this class.

878 **Table 22 – Class: CIM_PlatformWatchdogServiceCapabilities**

Elements	Requirement	Notes
InstanceID	Mandatory	Key
ElementName	Mandatory	Key
RequestedStatesSupported	Mandatory	See 7.5.
ElementNameEditSupported	Mandatory	See 7.1.11.
MaxElementNameLen	Conditional	See 7.1.11.
ActionOnExpirationSupported	Mandatory	See 7.1.10.
LastExpirationDataSupported	Mandatory	See 7.1.9.
WatchdogTimerDataSupported	Mandatory	See 7.1.6 and 7.1.8.

879
880
881
882

ANNEX A
(Informative)
Change Log

Version	Date	Description
1.0.0	2009-06-19	DMTF Standard Release

883
884