Document Number: DSP1040

Date: 2009-06-19

Version: 1.0.0

Platform Watchdog Profile

Document Type: Specification

Document Status: DMTF Standard

Document Language: E

CONTENTS

Figures

Tables

Foreword

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

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

Acknowledgments

The authors wish to acknowledge the following people.

Editor:

- John Leung – Intel

Contributors:

- Aaron Merkin – IBM
- Jon Hass – Dell
- Khachatur Papanyan – Dell
- Jeff Hilland – HP
- Christina Shaw – HP
- Joel Clark – Intel

Introduction

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

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

Platform Watchdog Profile

1 Scope

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

2 Normative References

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

2.1 Approved References

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

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

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

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

2.2 Other References

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

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

3 Terms and Definitions

For the purposes of this document, the following terms and definitions apply. For the purposes of this document, the terms and definitions given in DSP1033 and DSP1001 also apply.

3.1
can
used for statements of possibility and capability, whether material, physical, or causal

3.2
cannot
used for statements of possibility and capability, whether material, physical, or causal

3.3
conditional
indicates requirements to be followed strictly to conform to the document when the specified conditions are met

3.4
mandatory
indicates requirements to be followed strictly to conform to the document and from which no deviation is permitted

3.5
may
indicates a course of action permissible within the limits of the document

3.6
need not
indicates a course of action permissible within the limits of the document

3.7
optional
indicates a course of action permissible within the limits of the document

3.8
referencing profile
indicates a profile that owns the definition of this class and can include a reference to this profile in its "Referenced Profiles" table

3.9
shall
indicates requirements to be followed strictly to conform to the document and from which no deviation is permitted

3.10
shall not
indicates requirements to be followed strictly to conform to the document and from which no deviation is permitted

3.11
should
indicates that among several possibilities, one is recommended as particularly suitable, without mentioning or excluding others, or that a certain course of action is preferred but not necessarily required

3.12
should not
indicates that a certain possibility or course of action is deprecated but not prohibited

3.13
unspecified
indicates that this profile does not define any constraints for the referenced CIM element or operation

3.14
Watchdog
watchdog timer
a timer mechanism used to monitor the health of a software or hardware entity

4 Symbols and Abbreviated Terms

4.1
BIOS
basic input output system

4.2
OS
operating system

4.3
UTC
Coordinated Universal Time

5 Synopsis

Profile Name*:* Platform Watchdog

Version: 1.0.0

Organization: DMTF

CIM Schema Version: 2.22

Central Class: CIM_PlatformWatchdogService

Scoping Class: CIM_ComputerSystem

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

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

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

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

Table 1 – Referenced Profiles

Profile Name	Organization	Version	Relationship	Behavior
Profile Registration	DMTF	1.0	Mandatory	

6 Description (Informative)

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

A watchdog timer behaves as follows:

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

The countdown is performed with a specified timer resolution. When the timer reaches zero, the Watchdog expires.

Timeout values are represented in the data model using the interval format of the datetime type. The datetime format for intervals is ddddddddhhmmss.mmmmmm:000, where the meaning of each field is as follows:

- ddddddddd is the number of days.
- hh is the remaining number of hours.
- mm is the remaining number of minutes.
- ss is the remaining number of seconds.
- mmmmmm is the remaining number of microseconds.
- A colon (:) indicates that the value is an interval.
- 000 (the UTC offset field) is always zero for interval properties.

Fields that are not significant are replaced with asterisk (*) characters. Non-significant fields are those that are beyond the resolution of the data source.

A timeout value might start, for example, with a value of 00000000000001.000***:000, which represents a timeout of 1 second expressed with a 1 millisecond precision. After some time counting down, the remaining timeout value might be 00000000000000.125***.000, which indicates that 125 milliseconds remain before the timer expires.

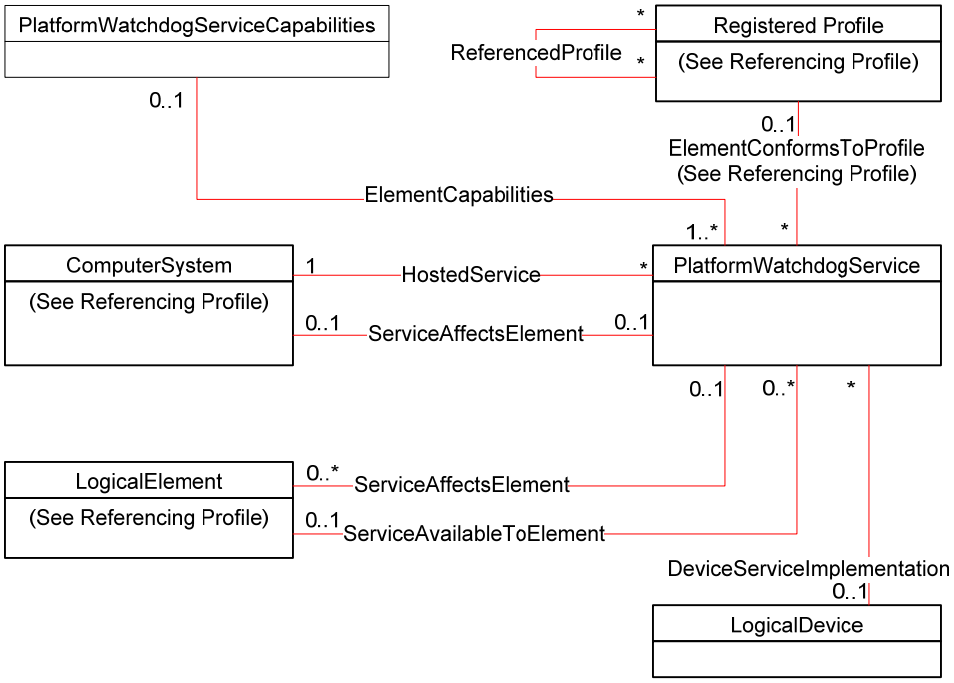
The watchdog timer is continuously prevented from expiring if the monitored entity is operational. This is accomplished by either stopping the timer within a specified timeout interval or periodically resetting the value of the timeout interval.

The expiration of the watchdog timer can cause a specific action to be performed. The action can be performed on the monitored entity or on another element on the platform.

One example is a Watchdog that monitors the operating system and resets the computer system if the watchdog timer expires. Another example is a Watchdog that monitors an application and generates a non-maskable interrupt if the watchdog timer expires.

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.

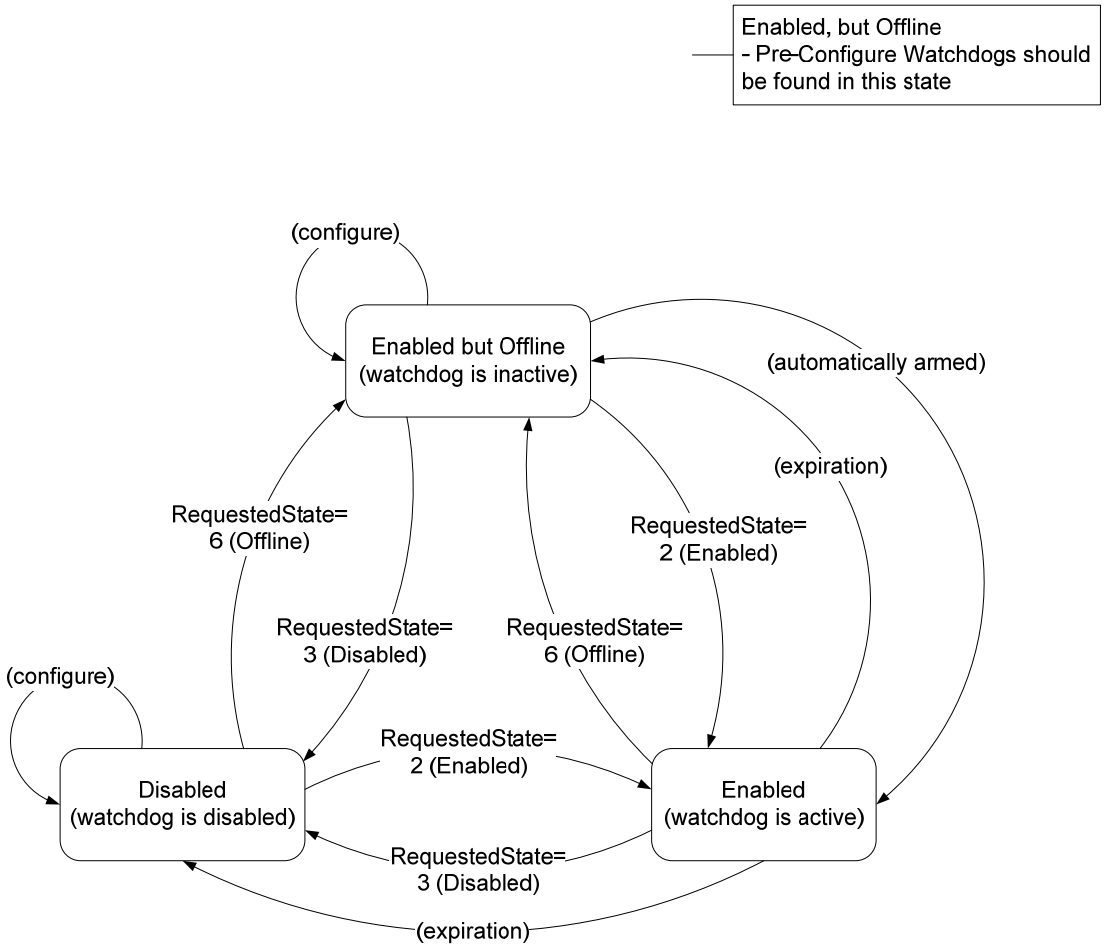


Figure 2 – CIM_PlatformWatchdogService State Chart

7 Implementation

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

7.1 Representing a Watchdog

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

7.1.1 CIM_PlatformWatchdogServiceCapabilities

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

7.1.2 Relationship to the Hosting Computer System

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

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

7.1.3 States of a Watchdog

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

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

7.1.4 Timeout Interval

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

7.1.5 Timer Expired

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

7.1.6 Timer Resolution (Optional)

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

7.1.6.1 TimerResolution Is Supported — Conditional

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

7.1.6.1.1 CIM_PlatformWatchdogServiceCapabilities

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

7.1.6.1.1.1 CIM_PlatformWatchdogServiceCapabilities.WatchdogTimerDataSupported

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

7.1.6.1.2 CIM_PlatformWatchdogService.TimerResolution

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

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

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

7.1.6.2 TimerResolution Is Not Supported

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

7.1.6.2.1 CIM_PlatformWatchdogServiceCapabilities

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

7.1.6.2.1.1 CIM_PlatformWatchdogServiceCapabilities.WatchdogTimerDataSupported

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

7.1.6.2.2 CIM_PlatformWatchdogService.TimerResolution

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

7.1.7 Type of Monitored Entity

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

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

7.1.8 Current Timer Value (Optional)

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

7.1.8.1 CurrentTimerValue Is Supported — Conditional

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

7.1.8.1.1 CIM_PlatformWatchdogServiceCapabilities

When the CurrentTimerValue property is supported, exactly one instance of CIM_PlatformWatchdogServiceCapabilities shall be associated with the instance of 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).

7.1.9.1.2 CIM_PlatformWatchdogService.TimeOfLastExpiration

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

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

7.1.9.2 TimeOfLastExpiration Is Not Supported

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

7.1.9.2.1 CIM_PlatformWatchdogServiceCapabilities

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

7.1.9.2.1.1 CIM_PlatformWatchdogServiceCapabilities.LastExpirationDataSupported

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

7.1.9.2.2 CIM_PlatformWatchdogService.TimeOfLastExpiration

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

7.1.10 Action on Expiration of the Watchdog Timer (Optional)

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

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

7.1.10.1 Action on Expiration Is Supported — Conditional

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

7.1.10.1.1 CIM_PlatformWatchdogServiceCapabilities

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

7.1.10.1.1.1 CIM_PlatformWatchdogServiceCapabilities.ActionOnExpirationSupported

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

7.1.10.1.2 CIM_PlatformWatchdogService.ActionOnExpiration

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

7.1.10.2 ActionOnExpiration Is Not Supported

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

7.1.10.2.1 CIM_PlatformWatchdogServiceCapabilities

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

7.1.10.2.1.1 CIM_PlatformWatchdogServiceCapabilities.ActionOnExpirationSupported

The ActionOnExpirationSupported array property shall not contain any values.

7.1.10.2.2 CIM_PlatformWatchdogService.ActionOnExpiration

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

7.1.11 CIM_PlatformWatchdogService.ElementName

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

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

7.1.11.1 Modifying ElementName Is Supported — Conditional

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

7.1.11.1.1 CIM_PlatformWatchdogServiceCapabilities

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

7.1.11.1.1.1 CIM_PlatformWatchdogServiceCapabilities.ElementNameSupported

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

7.1.11.1.1.2 CIM_PlatformWatchdogServiceCapabilities.MaxElementNameLen

The CIM_PlatformWatchdogServiceCapabilities.MaxElementNameLen property shall be implemented.

7.1.11.2 Modifying ElementName Is Not Supported

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

7.1.11.2.1 CIM_PlatformWatchdogServiceCapabilities

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

7.5.1.1 CIM_PlatformWatchdogServiceCapabilities

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

7.5.1.2 CIM_PlatformWatchdogServiceCapabilities.RequestedStatesSupported

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

7.5.1.3 CIM_PlatformWatchdogService.RequestedState

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

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

7.5.1.4 CIM_PlatformWatchdogService.EnabledState

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

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

7.5.2 Watchdog State Management Is Not Supported

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

7.5.2.1.1 CIM_PlatformWatchdogServiceCapabilities

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

7.5.2.2 CIM_PlatformWatchdogServiceCapabilities.RequestedStatesSupported

The CIM_PlatformWatchdogServiceCapabilities.RequestedStatesSupported property shall not contain any values.

7.5.2.3 CIM_PlatformWatchdogService.RequestedState

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

7.5.2.4 CIM_PlatformWatchdogService.EnabledState

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

8 Methods

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

8.1 CIM_PlatformWatchdogService.RequestStateChange( )

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

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

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

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

No standard messages are defined for this method.

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

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

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

8.8 CIM_LogicalDevice Operations

Compliant implementations of this profile shall support the operations listed in Table 9 for CIM_LogicalDevice. Each operation shall be supported as defined in DSP0200.

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

Table 9 – Operations: CIM_LogicalDevice

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

8.9 CIM_PlatformWatchdogService Operations

Compliant implementations of this profile shall support the operations listed in Table 10 for CIM_PlatformWatchdogService. Each operation shall be supported as defined in DSP0200.

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

Table 10 – Operations: CIM_PlatformWatchdogService

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

8.9.1 CIM_PlatformWatchdogService — ModifyInstance Operation

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

8.9.1.1 CIM_PlatformWatchdogService.ElementName Property

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

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

8.10 CIM_PlatformWatchdogServiceCapabilities Operations

Compliant implementations of this profile shall support the operations listed in Table 11 for the CIM_PlatformWatchdogServiceCapabilities class. Each operation shall be supported as defined in DSP0200.

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.

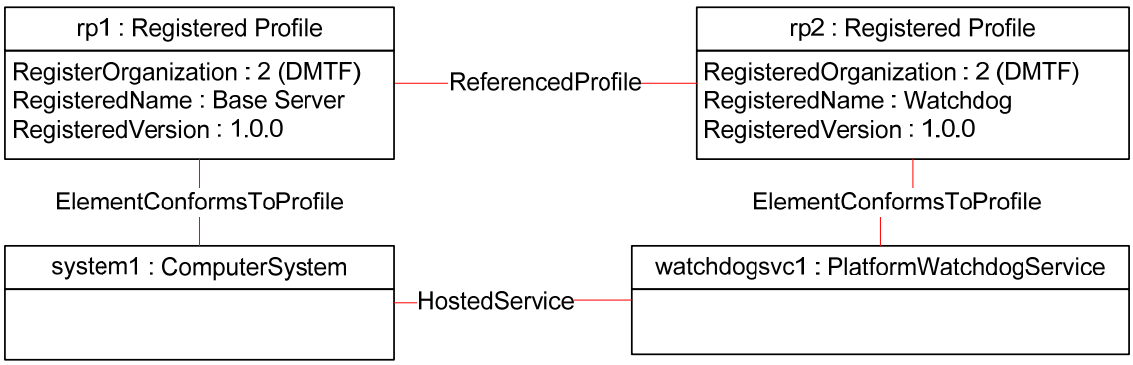
665 ## 9.1 Advertising the Profile Conformance

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.

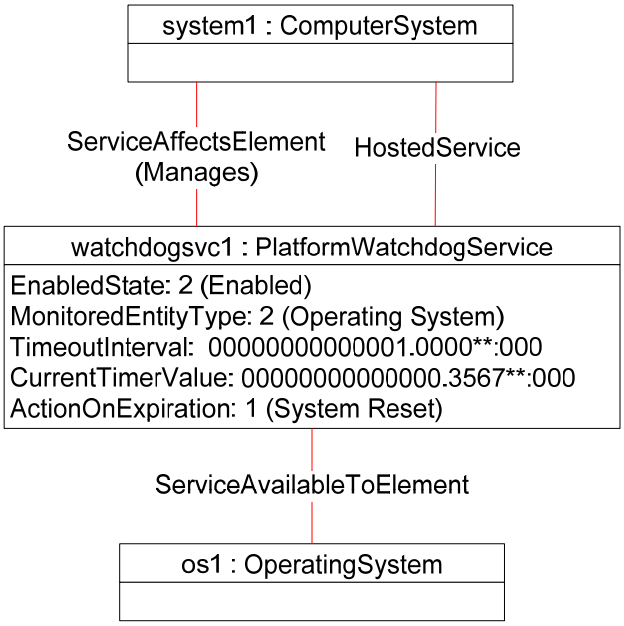


679

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.

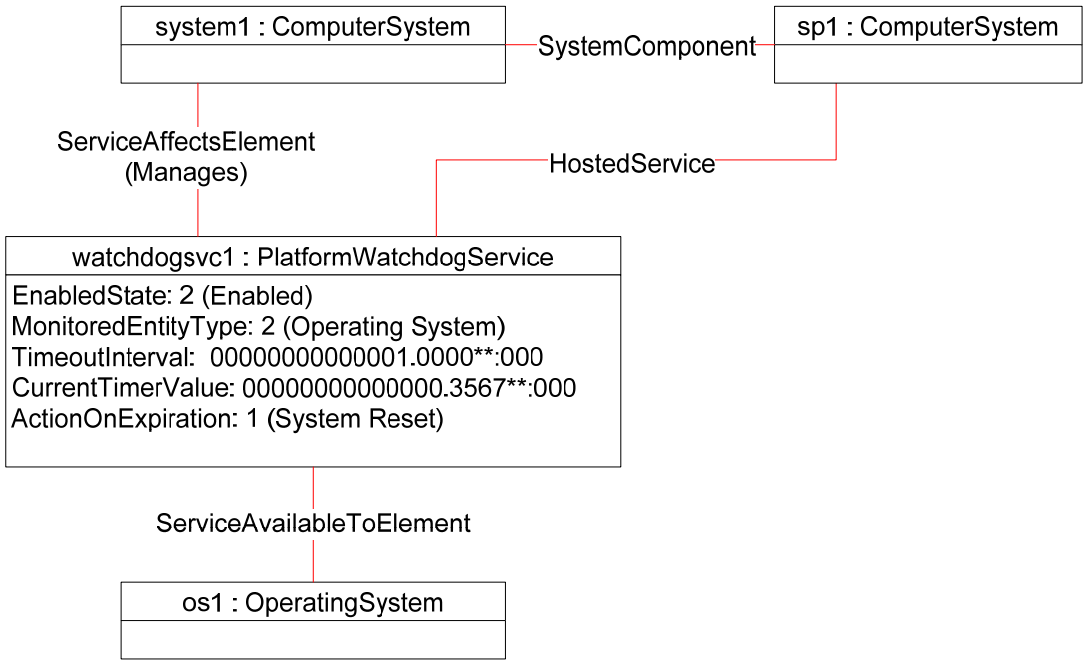


685

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**

9.4 Object Diagram for a Monolithic System with a Watchdog Device

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

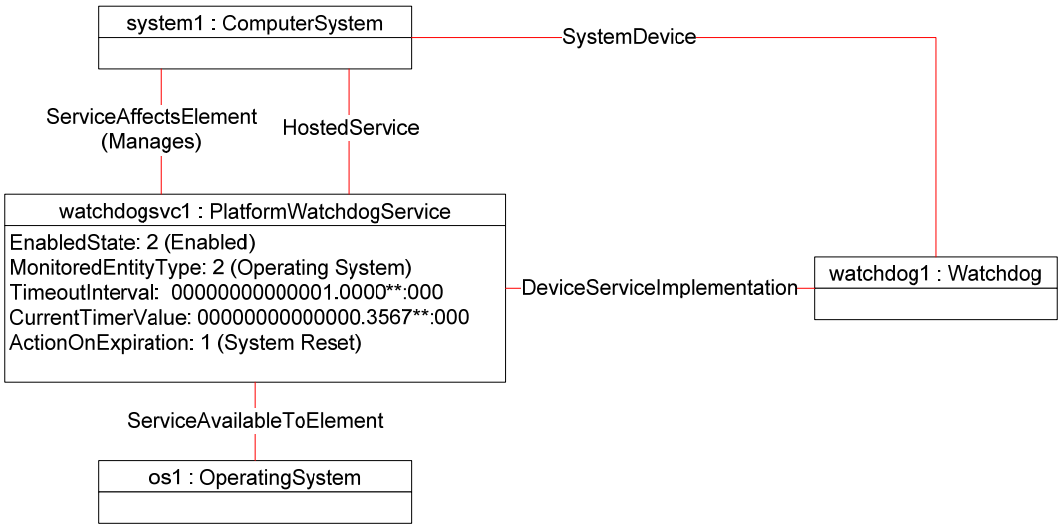


Figure 6 – Object Diagram: Reset System When a Watchdog Expires

9.5 Object Diagram for OS Reset When a Watchdog Timer Expires

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

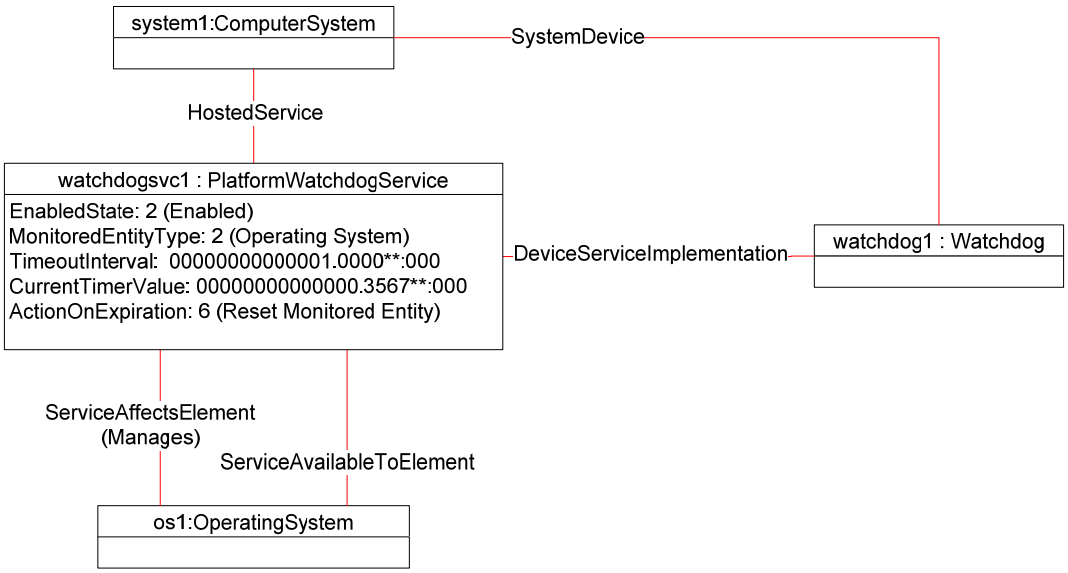


Figure 7 – Object Diagram: Reset OS When a Watchdog Timer Expires

9.6 Object Diagram for System with Multiple Watchdogs

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

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

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

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

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

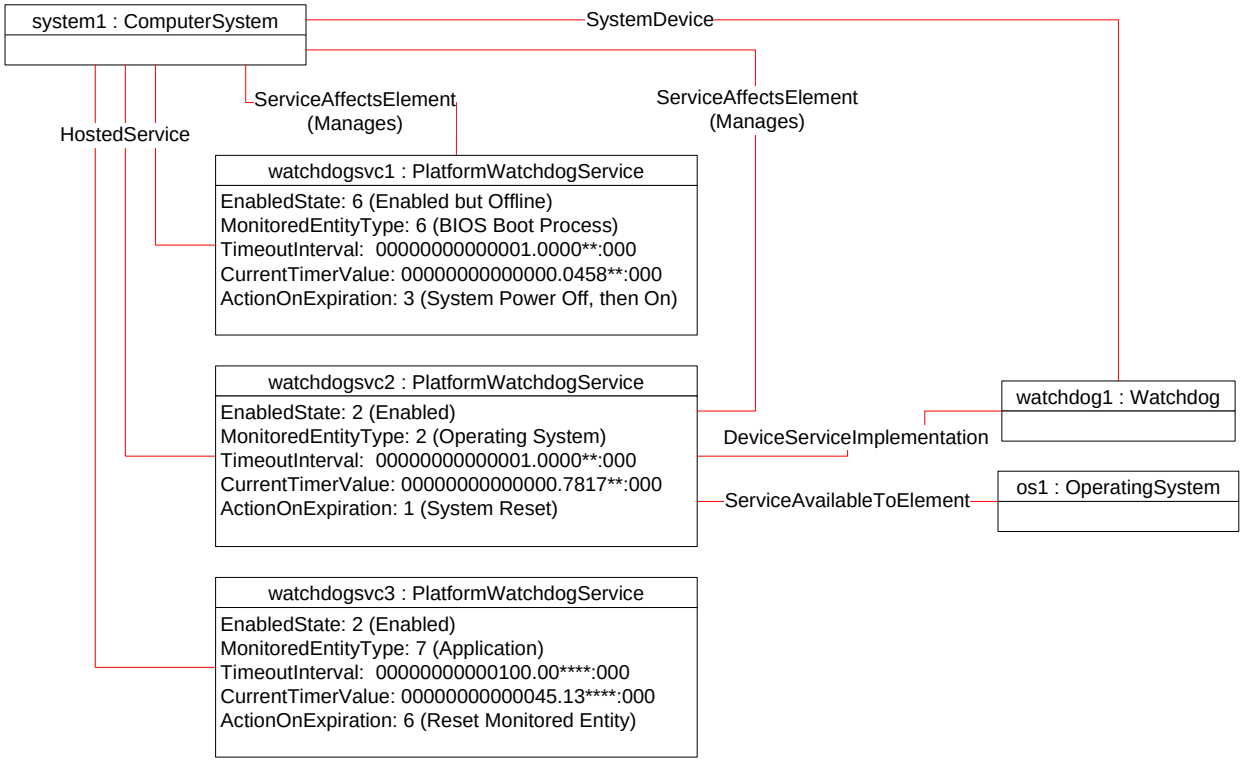


Figure 8 – Object Diagram: System with Multiple Watchdogs

9.7 Representing the Watchdog States

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

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

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

The CIM_PlatformWatchdogService instance at the lower right represents a Watchdog some time after it has been activated. The CurrentTimerValue property shall have a value between the TimeoutInterval value and the 0 (00000000000000.000000:000).

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

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

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

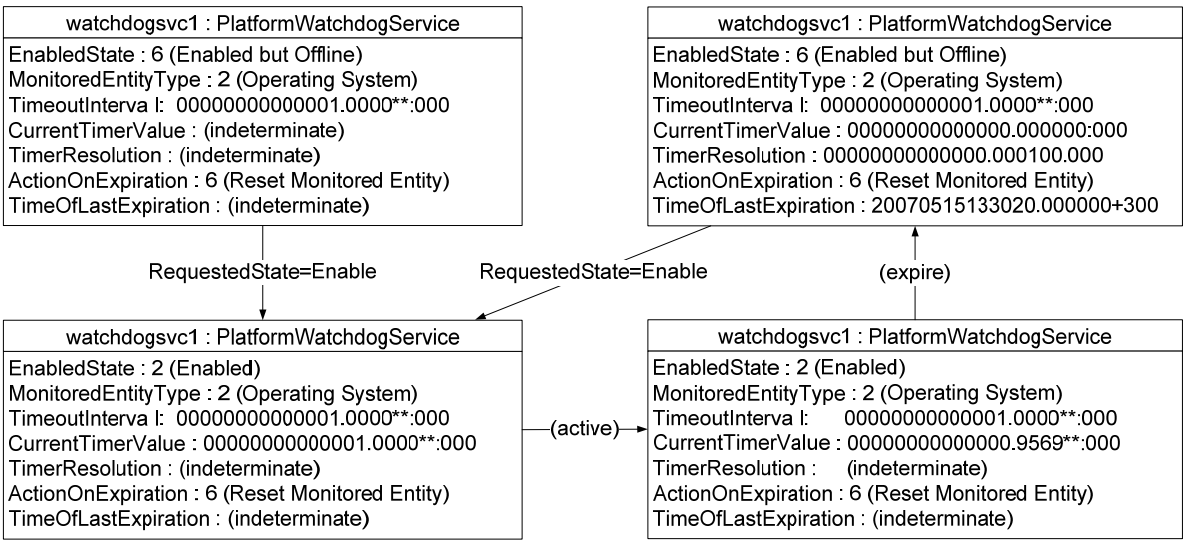


Figure 9 – CIM_PlatformWatchdogService Property Values per Watchdog State

9.8 Finding the Watchdogs Hosted on a Computer System

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

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

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

9.9 Finding the Watchdogs Monitoring an Entity Type

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

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

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

9.10 Finding the Watchdogs Monitoring an Entity Instance

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

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

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

9.11 Determining Whether a Watchdog Supports State Management

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

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

   If no CIM_PlatformWatchdogServiceCapabilities instance is returned, state management is not supported.
3) Query the value of the RequestedStatesSupported property array.

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

9.12 Activating a Watchdog

A client can activate a Watchdog of interest as follows:

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

   The Watchdog should now be active.

9.13 Obtaining Information Regarding the Last Watchdog Expiration

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

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

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

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

9.14 Determining Whether CIM_PlatformWatchdogService.ElementName Can Be Modified

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

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

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

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

10 CIM Elements

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

Table 12 lists the CIM elements that are required for this profile. The subsequent subclauses describe 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 "*".

10.3 CIM_ElementCapabilities

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

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".

10.4 CIM_HostedService

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

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 "*".

10.5 CIM_ServiceAffectsElement, Relating CIM_PlatformWatchdogService to CIM_ComputerSystem

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

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".

10.6 CIM_ServiceAffectsElement, Relating CIM_PlatformWatchdogService to a Concrete Subclass of CIM_LogicalElement

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

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..*".

10.7 CIM_ServiceAvailableToElement

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

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".

10.8 CIM_LogicalDevice

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

Table 20 – Class: CIM_LogicalDevice

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

10.9 CIM_PlatformWatchdogService

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

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.

10.10 CIM_PlatformWatchdogServiceCapabilities

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

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.

ANNEX A
(Informative)

Change Log

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