Document Number: DSP1030

Date: 2009-06-22

Version: 1.0.0

Battery Profile

Document Type: Specification

Document Status: DMTF Standard

Document Language: E

Table of Contents

List of Figures

List of Tables

Foreword

The *Battery Profile* (DSP1030) was prepared by the Desktop and Mobile Working Group.

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:

- Jon Hass – Dell

Contributors:

- Aaron Merkin – IBM
- Bob Blair – AMD
- David Hines – Intel
- Hemal Shah – Broadcom
- Jon Hass – Dell
- Khachatur Papanyan – Dell
- RadhaKrishna Dasari – Dell
- Scott Lenharth – Dell

Introduction

This document defines the classes used to describe the batteries in a managed system. Also included are descriptions of association classes that describe the relationship of the battery to the battery's physical aspects, such as FRU data, to sensors monitoring the battery, and to DMTF profile version information. The information in this specification is intended to be sufficient for a provider or consumer of this data to identify unambiguously the classes, properties, methods, and values that are mandatory to be instantiated and manipulated to represent and manage batteries of managed systems and subsystems modeled using the DMTF CIM core and extended model definitions.

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

Battery Profile

1 Scope

The *Battery Profile* extends the management capabilities of referencing profiles by adding the capability to represent batteries for manageability. The battery as a logical device is modeled as referencing the battery physical package for physical asset information, the sensor for sensor-reading information, and the profile registration for the schema implementation version information.

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 1.3,*
http://www.dmtf.org/standards/published_documents/DSP0200_1.3.pdf

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

DMTF DSP1009, *Sensors Profile1.0,*
http://www.dmtf.org/standards/published_documents/DSP1009_1.0.pdf

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

3 Terms and Definitions

For the purposes of this document, the following terms and definitions 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 strictly to be followed in order to conform to the document when the specified conditions are met

3.4
mandatory
indicates requirements strictly to be followed in order 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
a profile that owns the definition of this class and can include a reference to this profile in its “Approved References” section

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

3.10
shall not
indicates requirements strictly to be followed in order 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

4 Symbols and Abbreviated Terms

The following symbols and abbreviations are used in this document.

4.1
CIM
Common Information Model

4.2
FRU
Field Replaceable Unit

5 Synopsis

Profile Name: Battery

Version: 1.0.0

Organization: DMTF

CIM Schema Version: 2.22

Central Class: CIM_Battery

Scoping Class: CIM_ComputerSystem

The *Battery Profile* extends the management capability of the referencing profiles to describe and set the logical properties of the battery. Such properties include the description of the battery's charge status and the time it takes for the battery charge to be depleted. The profile also describes operations such as recharging the battery.

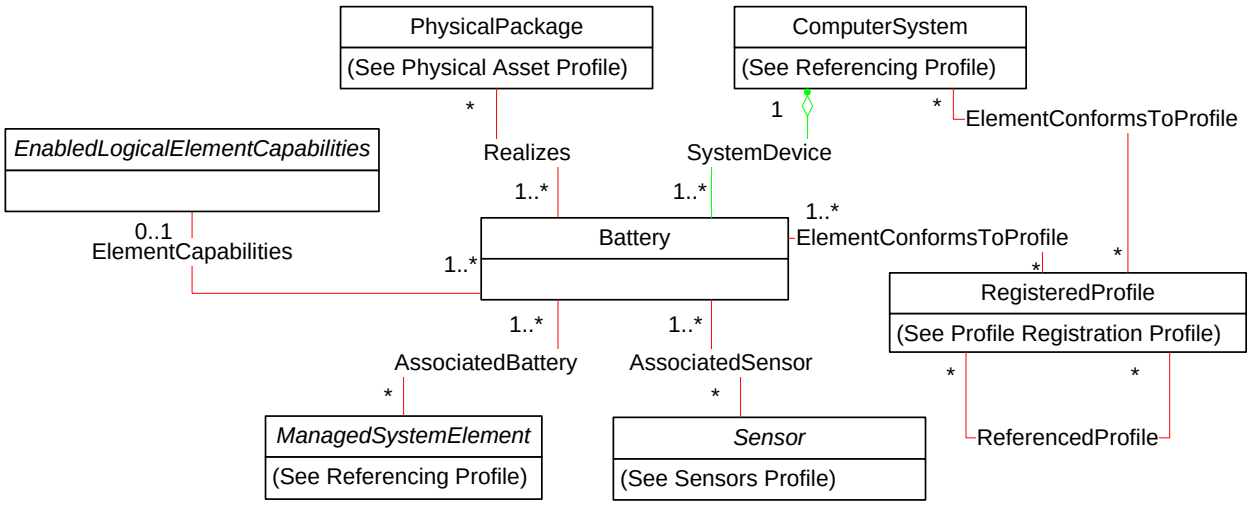
Table 1 – Related Profiles

Profile Name	Organization	Version	Relationship	Behavior
Physical Asset	DMTF	1.0	Optional	
Profile Registration	DMTF	1.0	Mandatory	
Sensor	DMTF	1.0	Optional	

6 Description

The *Battery Profile* describes the necessary set of logical values for managing battery devices. The properties in the Battery class provide the status of the battery as well as the estimation for its charge duration. Additionally, Battery class properties describe the physical characteristics of the battery such as its chemistry, voltages, and lifetime. The capabilities to disable, enable, recharge, and establish charge thresholds for a battery are advertised through the CIM_EnabledLogicalElementCapabilities instance. When a battery has an associated sensor, a CIM_Battery instance is associated to a CIM_Sensor instance through the CIM_AssociatedSensor class. The battery's physical description, such as FRU information, is represented by an instance of the CIM_PhysicalPackage class through the CIM_Realizes association. The managed system element that the battery is providing power for is represented by the CIM_Battery instance associated to a CIM_ManagedSystemElement sub-class instance through the CIM_AssociatedBattery association. The version of the *Battery Profile* implemented is represented through the CIM_RegisteredProfile class.

236 Figure 1 represents the class schema for the *Battery Profile*. For simplicity, the prefix CIM_ has been
237 removed from the names of the classes.



238

239 **Figure 1 – Class Diagram**

240 # 7 Implementation Requirements

241 This section details the requirements related to the arrangement of instances and properties of instances
242 for implementations of this profile.

243 ## 7.1 CIM_Battery

244 At least one instance of CIM_Battery shall be instantiated.

245 ## 7.2 CIM_Battery.BatteryStatus Value Formulation

246 Table 2 describes the mapping between the values of the CIM_Battery.BatteryStatus property and the
247 corresponding description of the status of the battery. The CIM_Battery.BatteryStatus property shall
248 match one of the values that are specified in Table 2.

249 **Table 2 – BatteryStatus Value Description**

Value	Description	Extended Description
2	Unknown	The battery status is not known.
3	Fully Charged	The battery is fully charged.
4	Low	The battery charge is low.
5	Critical	The battery charge is nearly empty.
6	Charging	The battery is being charged.
10	Undefined	There is no information on the battery status.
12	Learning	The battery is undergoing a learning cycle to recalibrate low- and high-charge capacity thresholds.
13	Overcharged	The battery is more than fully charged.

7.3 CIM_Battery.Chemistry Value Formulation (Optional)

The CIM_Battery.Chemistry property shall have one of the following values: 1 (Other), 2 (Unknown), 3 (Lead Acid), 4 (Nickel Cadmium), 5 (Nickel Metal Hydride), 6 (Lithium-ion), 7 (Zinc air), or 8 (Lithium Polymer).

7.4 CIM_AssociatedBattery (Optional)

The CIM_AssociatedBattery class is used to associate the battery to the component that uses or requires the battery. If a component that uses or requires the battery is represented by the instance of a subclass of CIM_ManagedSystemElement, then the CIM_AssociatedBattery class shall be used.

When no instance of CIM_AssociatedBattery references the instance of CIM_Battery, the battery represented by CIM_Battery supplies power to the whole managed system. In this case, the CIM_ComputerSystem instance and the CIM_Battery instance shall be associated only through an instance of CIM_SystemDevice.

7.5 Modeling a Battery Temperature Sensor (Optional)

This section describes constraints for representing a battery temperature sensor. These constraints are in addition to the constraints described in the *Sensors Profile*.

A battery discrete temperature sensor shall be represented by the CIM_Sensor class associated to the CIM_Battery instance using the CIM_AssociatedSensor class. The CIM_Sensor.Type property shall match 2 (Temperature).

A battery analog temperature sensor shall be represented by the CIM_NumericSensor class associated to the CIM_Battery instance using the CIM_AssociatedSensor class. The CIM_NumericSensor instance shall comply with the following requirements:

- CIM_NumericSensor.Type matches 2 (Temperature).
- CIM_NumericSensor.BaseUnits matches 2 (Degrees C), 3 (Degrees F), or 4 (Degrees K)
- CIM_NumericSensor.RateUnits matches 0 (None)

An implementation may instantiate other CIM_Sensor or CIM_NumericSensor instances that are associated to the CIM_Battery instance, which represents other types of sensors. For example, a battery analog power production sensor may be associated to the CIM_Battery instance for the purposes of representing the power output of the battery.

7.6 Managing the Battery's State (Optional)

This section describes the management of the battery's state. Battery state management consists of the CIM_Battery.RequestStateChange() method being supported (see section 8.1) and the CIM_Battery.RequestedState property having a value other than 12 (Not Applicable).

7.6.1 Battery State Management Support

When no instance of CIM_EnabledLogicalElementCapabilities is associated with the CIM_Battery instance, battery state management shall not be supported.

When a CIM_EnabledLogicalElementCapabilities instance is associated with the CIM_Battery instance but the value of the CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported property is an empty array, battery state management shall not be supported.

When a CIM_EnabledLogicalElementCapabilities instance is associated with the CIM_ Battery instance and the value of the CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported property is not an empty array, battery state management shall be supported.

7.6.2 CIM_Battery.RequestedState

When state management is supported, the RequestedState property shall be supported. When state management is unspecified, the RequestedState property may be supported.

The CIM_Battery.RequestedState property shall be set to the value of the RequestedState parameter of the CIM_Battery.RequestStateChange( ) method, if the method is executed and is supported. The CIM_Battery.RequestedState property shall match one of the values that are specified in Table 3. After the RequestStateChange( ) method has successfully executed, RequestedState and EnabledState shall have equal values, with the exception of the transitional requested states 7 (Test) and 11 (Reset). The value of the RequestedState property may also change as a result of a request for change to the battery's enabled state by a non-CIM implementation.

Table 3 – RequestedState Values

Value	Description	Extended Description
0	Unknown	The battery state is indeterminate.
2	Enabled	The battery shall be made ready to provide power.
3	Disabled	The battery shall be made not ready to provide power.
7	Test	The battery shall begin undergoing the learning cycle process; low and high voltage thresholds will be calculated.
11	Reset	The battery shall begin recharging.
12	Not Applicable	State management is not supported.

7.6.3 CIM_Battery.EnabledState

Table 4 describes the mapping between the values of the CIM_Battery.EnabledState property and the corresponding description of the state of the battery. The CIM_Battery.EnabledState property shall match one of the values that are specified in Table 4. When the RequestStateChange( ) method executes but does not complete successfully, and the battery is in an indeterminate state, the CIM_Battery.EnabledState property shall have value of 5 (Not Applicable). The value of the EnabledState property may also change as a result of a change to the battery's enabled state by a non-CIM implementation.

Table 4 – EnabledState Values

Value	Description	Extended Description
0	Unknown	The battery state is indeterminate.
2	Enabled	The battery shall be available to provide power.
3	Disabled	The battery shall not be available to provide power.
5	Not Applicable	State management is not supported.

7.6.4 CIM_EnabledLogicalElementCapabilities

The CIM_EnabledLogicalElementCapabilities class is used for advertising the capabilities of the CIM_Battery instance.

7.6.4.1 CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported

The CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported property is an array that contains the supported requested states for the instance of CIM_Battery. The value of the CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported property shall be an empty array or contain any combination of the following values: 2 (Enabled), 3 (Disabled), 7 (Test), 11 (Reset), and 12 (Not Applicable).

7.6.4.2 CIM_EnabledLogicalElementCapabilities.ElementNameEditSupported

The CIM_EnabledLogicalElementCapabilities.ElementName property may be modifiable by a client, or it may have a fixed value.

The CIM_EnabledLogicalElementCapabilities.ElementNameEditSupported property shall have a value of TRUE when the implementation supports client modification of the associated instance of the CIM_Battery.ElementName property.

7.6.4.3 CIM_EnabledLogicalElementCapabilities.ElementNameMask

The CIM_EnabledLogicalElementCapabilities.ElementNameMask property provides the regular expression that expresses the limits of the CIM_EnabledLogicalElementCapabilities.ElementName property. The CIM_EnabledLogicalElementCapabilities.ElementNameMask property shall have a value as described in DSP1001.

7.6.4.4 CIM_EnabledLogicalElementCapabilities.MaxElementNameLen

The CIM_EnabledLogicalElementCapabilities.MaxElementNameLen property shall be implemented when the CIM_EnabledLogicalElementCapabilities.ElementNameEditSupported property has a value of TRUE. The CIM_EnabledLogicalElementCapabilities.MaxElementNameLen property shall indicate the maximum length of a string that the implementation will accept as a value for the CIM_EnabledLogicalElementCapabilities.ElementName property of the associated CIM_Battery instance.

8 Methods

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

8.1 Method: CIM_Battery.RequestStateChange( )

Invocation of the CIM_Battery.RequestStateChange( ) method will change the battery status and/or state to the value specified in the RequestedState parameter. The CIM_Battery.RequestedState property and CIM_Battery.EnabledState property shall be affected by the invocation of the RequestStateChange( ) method as specified in sections 7.6.3 and 7.6.2. The “Enabled”/”Disabled” values of the RequestedState parameter correspond to turning on/off the battery represented by the instance of CIM_Battery. The “Reset” value of the RequestedState parameter corresponds to requesting the battery to perform a recharge operation. The “Test” value of the RequestedState parameter corresponds to requesting the battery to perform a recalculation of charge thresholds.

The RequestStateChange( ) method’s detailed requirements are specified in Table 5 and Table 6. The return code values specified in Table 5 shall be returned by the RequestStateChange( ) method when the execution behavior of the method matches the description in Table 5. Table 6 specifies RequestStateChange( ) method parameters. The RequestedState parameter shall be required by the RequestStateChange( ) method.

Table 5 – CIM_Battery.RequestStateChange( ) Method: Return Code Values

Value	Description
0	Initiation of the state change request was successful.
1	Method is not supported in the implementation.
2	Error Occurred
4096	Job started: REF returned to started CIM_ConcreteJob

355 **Table 6 – CIM_Battery.RequestStateChange( ) Method: Parameters**

Qualifiers	Name	Type	Description/Values
IN, REQ	RequestedState	uint16	Valid state values: 2 (Enabled) 3 (Disabled) 7 (Test) 11 (Reset)
OUT	Job	CIM_ConcreteJob REF	Return if job started
IN, REQ	TimeoutPeriod	Datetime	Client-specified maximum amount of time the transition to a new state is supposed to take: 0 or NULL – No time requirements <interval> – Maximum time allowed

356 ### 8.1.1 CIM_Battery.RequestStateChange( ) Conditional Support

357 When the CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported property contains at least
358 one value, the CIM_Battery.RequestStateChange( ) method shall be implemented and supported. The
359 CIM_Battery.RequestStateChange( ) method shall not return a value of 1 (Unsupported).

360 ## 8.2 Profile Conventions for Operations

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

363 The default list of operations is as follows:

364 • GetInstance
365 • Associators
366 • AssociatorNames
367 • References
368 • ReferenceNames
369 • EnumerateInstances
370 • EnumerateInstanceNames

371 ## 8.3 CIM_AssociatedBattery

372 Table 7 lists implementation requirements for operations. If implemented, these operations shall be
373 implemented as defined in DSP0200. In addition, and unless otherwise stated in Table 7, all operations in
374 the default list in 8.2 shall be implemented as defined in DSP0200.

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

376 **Table 7 – Operations: CIM_AssociatedBattery**

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

377 8.4 CIM_ElementCapabilities

378 Table 8 lists implementation requirements for operations. If implemented, these operations shall be
379 implemented as defined in DSP0200. In addition, and unless otherwise stated in Table 8, all operations in
380 the default list in 8.2 shall be implemented as defined in DSP0200.

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

382 **Table 8 – Operations: CIM_ElementCapabilities**

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

383 8.5 CIM_EnabledLogicalElementCapabilities

384 Table 9 lists implementation requirements for operations. If implemented, these operations shall be
385 implemented as defined in DSP0200. In addition, and unless otherwise stated in Table 9, all operations in
386 the default list in 8.2 shall be implemented as defined in DSP0200.

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

388 **Table 9 – Operations: CIM_EnabledLogicalElementCapabilities**

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

8.6 CIM_Battery

Table 10 lists implementation requirements for operations. If implemented, these operations shall be implemented as defined in DSP0200. In addition, and unless otherwise stated in Table 10, all operations in the default list in 8.2 shall be implemented as defined in DSP0200.

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

Table 10 – Operations: CIM_Battery

Operation	Requirement	Messages
ModifyInstance	Optional. See section 8.6.1.	None

8.6.1 CIM_Battery — ModifyInstance

This section details the requirements for the ModifyInstance operation applied to an instance of CIM_Battery. The ModifyInstance operation may be supported. See section 8.6.1.1.

8.6.1.1 CIM_Battery.ElementName

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

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

8.7 CIM_SystemDevice

Table 11 lists implementation requirements for operations. If implemented, these operations shall be implemented as defined in DSP0200. In addition, and unless otherwise stated in Table 11, all operations in the default list in 8.2 shall be implemented as defined in DSP0200.

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

Table 11 – Operations: CIM_SystemDevice

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

9 Use Cases

This section contains object diagrams and use cases for the *Battery Profile*.

9.1 Object Diagrams

Figure 2 represents a possible instantiation of the *Battery Profile*. In this instantiation, a battery instance, battery1, is associated with a computer system, system1. The battery is operating but is only partially charged. battery1's physical package information is represented as well.

Because battery1 does not have the CIM_AssociatedBattery association reference, battery1 is available to supply power to system1, which is denoted by the CIM_SystemDevice association. system1 is also the scoping instance for battery1. Thus, following the CIM_ElementConformsToProfile association to profile1 and then the referenced CIM_ReferencedProfile association to a CIM_RegisteredProfile instance with the RegisteredName property set to "Battery", the client can retrieve profile2. profile2 shows the version of the current *Battery Profile* implementation.

For simplicity, the prefix CIM_ has been removed from the class names in the figure.

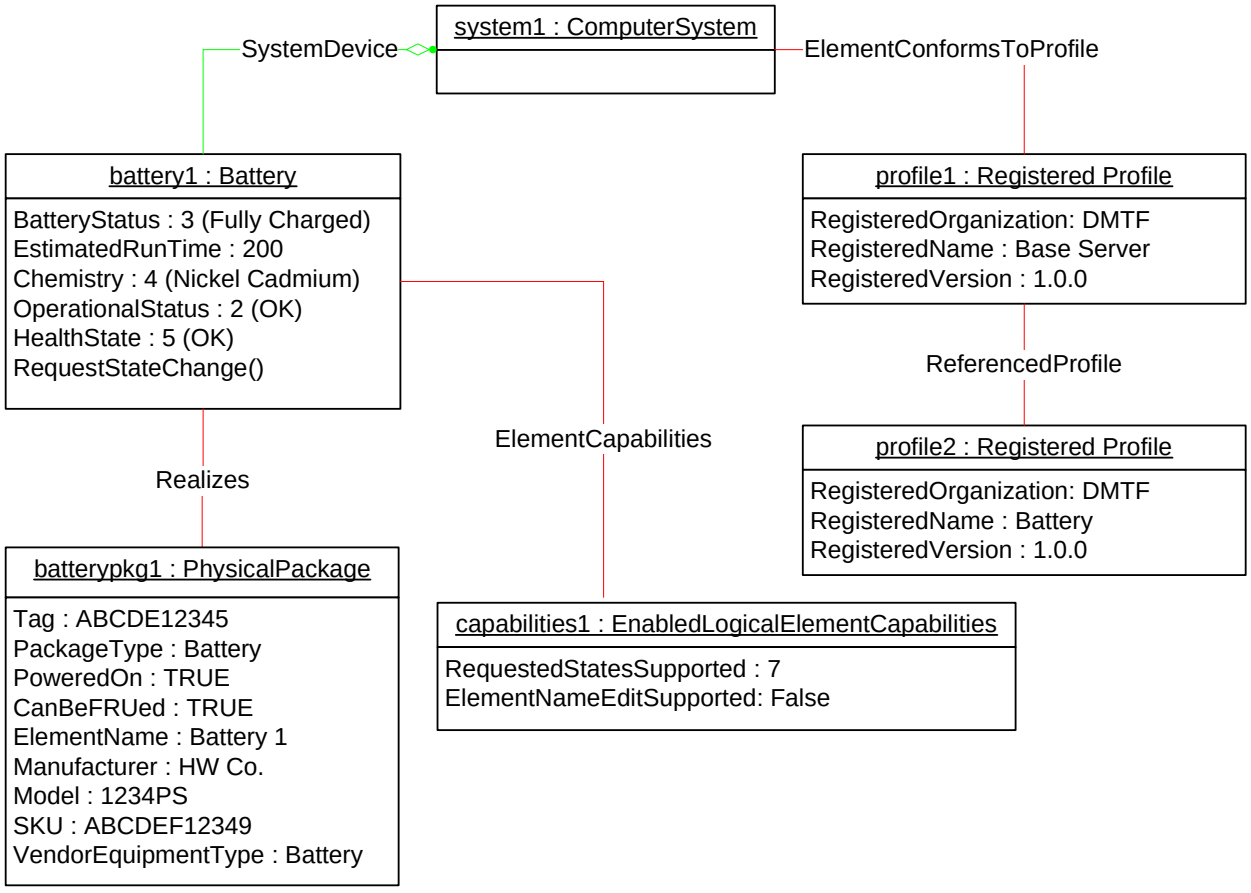


Figure 2 – Instance Diagram 1

Figure 3 represents a possible instantiation of the *Battery Profile*. In this instantiation, a battery instance, battery1, is associated with a sensor, sensor1. The battery is fully charged but is degraded. Following the CIM_AssociatedSensor association to sensor1, the CurrentState property value indicates that the temperature sensor is in a bad state.

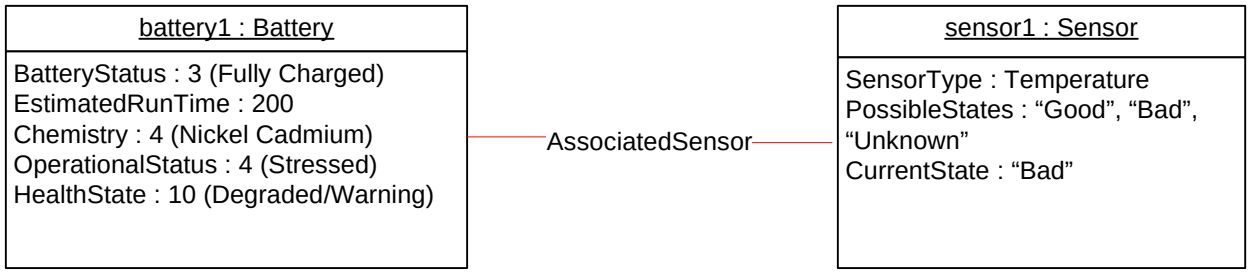


Figure 3 – Instance Diagram 2

Figure 4 represents a possible instantiation of the *Battery Profile*. In this instantiation, a battery instance, battery1, is associated with a sensor, sensor1. The battery is fully charged and is operating. Following the CIM_AssociatedSensor association to sensor1, the CurrentState property value indicates that the voltage sensor is in a good state.

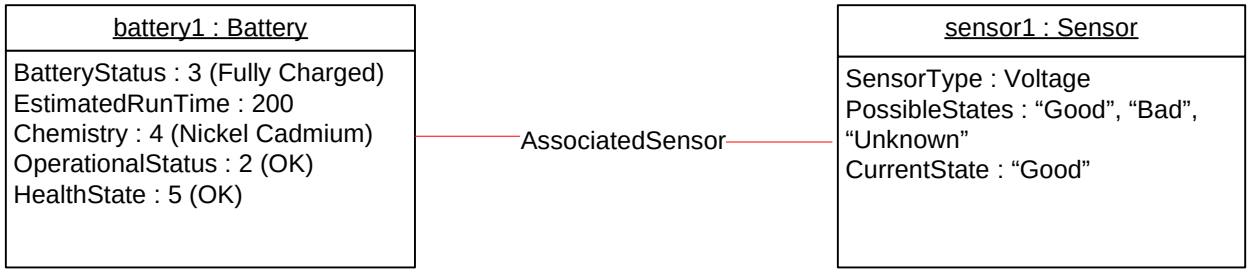


Figure 4 – Instance Diagram 3

Figure 5 represents a possible instantiation of the *Battery Profile*. In this instantiation, a battery instance, battery1, is associated with capabilities, capabilities1. battery1 is charging. Its RequestedState property has a value of 11 (Reset), which indicates that a recharge was initiated. The BatteryStatus property has a value of 6 (Charging), which indicates that the recharge is still active. The battery can be recharged, using the RequestStateChange( ) method, because the value of the RequestedStatesSupported property of capabilities1 is 11 (Reset).

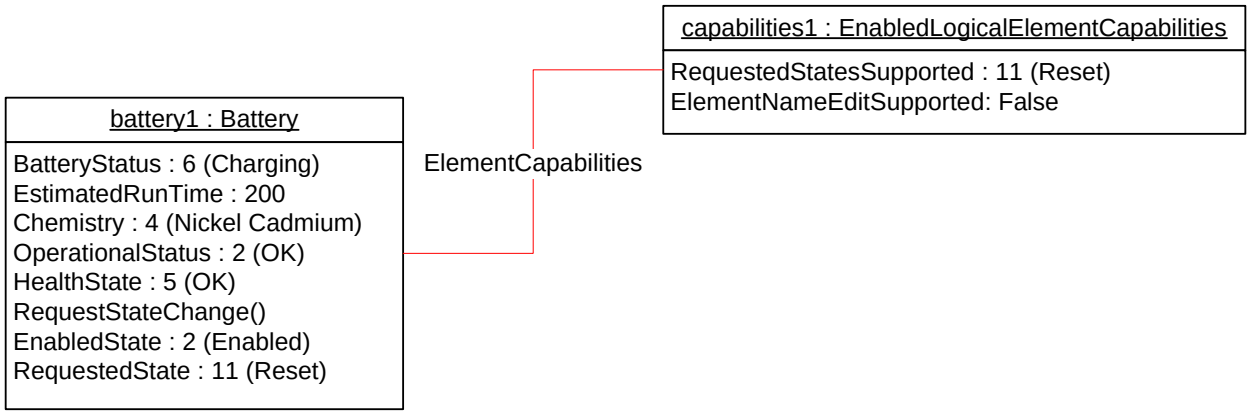


Figure 5 – Instance Diagram 4

9.2 Recharge a Battery

A client can recharge a battery as follows:

1) Find the instance of CIM_EnabledLogicalElementCapabilities that is associated with the instance of CIM_Battery through an instance of CIM_ElementCapabilities. If the instance is not found, the battery cannot be recharged.
2) Retrieve the value of the CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported property. If the property is a non-empty array that contains the value 11 (Reset), execute the RequestStateChange( ) method with the value of the RequestedState parameter set to 11 (Reset), which will begin a recharge of the battery represented by this instance.

9.3 Recalibrate Battery Thresholds

A client can recalibrate battery thresholds as follows:

1) Find the instance of CIM_EnabledLogicalElementCapabilities that is associated with the instance of CIM_Battery through an instance of CIM_ElementCapabilities. If the instance is not found, the battery thresholds cannot be recalibrated.
2) Retrieve the value of the CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported property. If the property is a non-empty array that contains the value 7 (Test), execute the RequestStateChange( ) method with the value of the RequestedState parameter set to 7 (Test), which will begin a recalibration of the battery represented by this instance.

9.4 Determine Additional Information for the Health of a Battery Using a Sensor

A client can determine the health of a battery using a sensor as follows:

1) Find the instance of CIM_Sensor or CIM_NumericSensor that is associated with the instance of CIM_Battery through an instance of CIM_AssociatedSensor. If the instance is not found, the battery properties CIM_Battery.BatteryStatus, CIM_Battery.HealthState, and CIM_Battery.OperationalStatus provide the only health information for the battery.
2) Retrieve the value of the CIM_Sensor.CurrentState property or the CIM_NumericSensor.CurrentState property.

10 CIM Elements

Table 12 shows the instances of CIM Elements for this profile. Instances of these CIM Elements shall be implemented as described in Table 12.

Table 12 – CIM Elements: Battery Profile

Element Name	Requirement	Description
Classes		
CIM_AssociatedBattery	Optional	See sections 7.4 and 10.1.
CIM_Battery	Mandatory	See sections 7.1 and 10.2.
CIM_ElementCapabilities	Optional	See section 10.3.
CIM_EnabledLogicalElementCapabilities	Optional	See sections 7.6.4 and 10.4.
CIM_NumericSensor	Optional	See sections 7.5 and 10.5.
CIM_RegisteredProfile	Mandatory	See section 10.6.
CIM_Sensor	Optional	See sections 7.5 and 10.7.
CIM_SystemDevice	Mandatory	See section 10.8.

10.1 CIM_AssociatedBattery

The CIM_AssociatedBattery class is used to associate an instance of CIM_Battery with a device represented by an instance of the CIM_ManagedSystemElement subclass that uses or requires the battery. Table 13 contains the requirements for elements of this class.

Table 13 – Class: CIM_AssociatedBattery

Properties	Requirement	Description
Antecedent	Mandatory	**Key** This property shall be a reference to an instance of CIM_Battery. Cardinality 1..*
Dependent	Mandatory	**Key** This property shall reference the instance of a subclass of CIM_ManagedSystemElement that represents the element that uses or requires the battery. Cardinality *

10.2 CIM_Battery

The CIM_Battery class is used to represent the battery. Table 14 contains the requirements for elements of this class.

Table 14 – Class: CIM_Battery

Properties	Requirement	Description
SystemCreationClassName	Mandatory	**Key**
SystemName	Mandatory	**Key**
CreationClassName	Mandatory	**Key**
DeviceID	Mandatory	**Key**
BatteryStatus	Mandatory	See section 7.2.
OperationalStatus	Mandatory	None
HealthState	Mandatory	None
EnabledState	Mandatory	See section 7.6.3.
RequestedState	Mandatory	See section 7.6.2.
ElementName	Mandatory	Pattern ".*"
Chemistry	Optional	See section 7.3.
MaxRechargeCount	Optional	None
RechargeCount	Optional	None
ExpectedLife	Optional	None
EstimatedRunTime	Optional	None
TimeToFullCharge	Optional	None
MaxRechargeTime	Optional	None

10.3 CIM_ElementCapabilities

The CIM_ElementCapabilities class is used to associate an instance of CIM_EnabledLogicalElementCapabilities with an instance of CIM_Battery. Table 15 contains the requirements for elements of this class.

Table 15 – Class: CIM_ElementCapabilities

Properties	Requirement	Notes
ManagedElement	Mandatory	**Key** This property shall be a reference to an instance of CIM_Battery. Cardinality 1..*
Capabilities	Mandatory	**Key** This property shall be a reference to the CIM_EnabledLogicalElementCapabilities instance. Cardinality 1

10.4 CIM_EnabledLogicalElementCapabilities

The CIM_EnabledLogicalElementCapabilities class represents the capabilities of the battery. Table 16 contains the requirements for elements of this class.

Table 16 – Class: CIM_EnabledLogicalElementCapabilities

Properties	Requirement	Notes
InstanceID	Mandatory	**Key**
RequestedStatesSupported	Mandatory	See section 7.6.4.1.
ElementNameEditSupported	Mandatory	See section 7.6.4.2.
ElementNameMask	Mandatory	See section 7.6.4.3.
MaxElementNameLen	Conditional	See section 7.6.4.4.

10.5 CIM_NumericSensor

The CIM_NumericSensor class is defined by the *Sensors Profile*. The requirements denoted in Table 17 are in addition to those mandated by the *Sensors Profile*. See section 2.1.

Table 17 – Class: CIM_NumericSensor

Properties	Requirement	Description
SensorType	Mandatory	Matches 2 (Temperature)
BaseUnits	Mandatory	Matches 2 (Degrees C), 3 (Degrees F), or 4 (Degrees K)
RateUnits	Mandatory	Matches 0 (None)

10.6 CIM_RegisteredProfile

The CIM_RegisteredProfile class is defined by the *Profile Registration Profile*. The requirements denoted in Table 18 are in addition to those mandated by the *Profile Registration Profile*. See section 2.1.

Table 18 – Class: CIM_RegisteredProfile

Properties	Requirement	Description
RegisteredName	Mandatory	Matches "Battery"
RegisteredVersion	Mandatory	Matches "1.0.0"
RegisteredOrganization	Mandatory	Matches 2 (DMTF)

10.7 CIM_Sensor

The CIM_Sensor class is defined by the *Sensors Profile*. The requirements denoted in Table 19 are in addition to those mandated by the *Sensors Profile*. See section 2.1.

Table 19 – Class: CIM_Sensor

Properties	Requirement	Description
SensorType	Mandatory	Matches 2 (Temperature)

10.8 CIM_SystemDevice

The CIM_SystemDevice class is used to associate an instance of CIM_Battery with the CIM_ComputerSystem instance of which the CIM_Battery instance is a member. Table 20 contains the requirements for elements of this class.

Table 20 – Class: CIM_SystemDevice

Properties	Requirement	Description
GroupComponent	Mandatory	**Key** This property shall be a reference to the CIM_ComputerSystem instance of which a CIM_Battery instance is a member. Cardinality 1
PartComponent	Mandatory	**Key** This property shall be a reference to an instance of CIM_Battery. Cardinality 1..*

513
514
515

ANNEX A
(informative)
Change Log

Version	Date	Description
1.0.0e	5/1/2007	Preliminary Standard
1.0.0	5/13/2009	DMTF Standard Release

516