Document Number: DSP1058

Date: 2008-12-09

Version: 1.0.0

Base Desktop and Mobile Profile

Document Type: Specification

Document Status: Final Standard

Document Language: E

CONTENTS

Figures

Tables

Foreword

The *Base Desktop and Mobile Profile* (DSP1058) was prepared by the Desktop Mobile Working Group and 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.

Introduction

The information in this specification should be sufficient for a provider or consumer of this data to unambiguously identify the classes, properties, methods, and values that shall be instantiated and manipulated to represent and manage a monolithic desktop or mobile computer and its subsystems using the DMTF Common Information Model (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 that represent the components described in this document.

Base Desktop and Mobile Profile

1 Scope

The *Base Desktop and Mobile Profile* is an autonomous profile that defines the classes used to describe monolithic desktop or mobile computer hardware and related software. The scope of this profile is limited to monolithic desktop or mobile computer hardware and related software that are directly realized in physical components. The profiles referenced by the *Base Desktop and Mobile Profile* extend the management capabilities described in this profile.

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

Advanced Configuration and Power Interface Specification, revision 3.0, http://www.acpi.info/Downloads/ACPIspec30.pdf

DMTF DSP1012, *Boot Control Profile 1.0.0*

DMTF DSP0004, *CIM Infrastructure Specification 2.3.0*

DMTF DSP0200, *CIM Operations over HTTP 1.2.0*

DMTF DSP1052, *Computer System Profile 1.0.0*

DMTF DSP1022, *CPU Profile 1.0.0*

DMTF DSP1000, *Management Profile Specification Template*

DMTF DSP1001, *Management Profile Specification Usage Guide*

DMTF DSP1011, *Physical Asset Profile 1.0.0*

DMTF DSP1015, *Power Supply Profile 1.0.0*

DMTF DSP1027, *Power State Management Profile 1.0.0*

DMTF DSP1033, *Profile Registration Profile 1.0.0*

DMTF DSP1023, *Role Based Authorization Profile 1.0.0*

DMTF DSP1009, *Sensors Profile 1.0.0*

DMTF DSP1034, *Simple Identity Management Profile 1.0.0*

DMTF DSP1026, *System Memory Profile 1.0.0*

2.2 References under Development

2.3 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

Unified Modeling Language (UML) Specifications from the Open Management Group (OMG), http://www.omg.org/technology/documents/modeling_spec_catalog.htm#UML

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

4 Symbols and Abbreviated Terms

4.1
ACPI
Advanced Configuration and Power Interface specification

5 Synopsis

Profile Name: Base Desktop and Mobile

Version: 1.0.0

Organization: DMTF

CIM schema version: 2.20.0

Specializes: DMTF *Computer System Profile* 1.0.0 (DSP1052)

Central Class: CIM_ComputerSystem

Scoping Class: CIM_ComputerSystem

The *Base Desktop and Mobile Profile* is an autonomous profile that provides the capability to manage monolithic desktop or mobile computer hardware and related software.

The Central Class of the *Base Desktop and Mobile Profile* shall be CIM_ComputerSystem. The Central Instance shall be an instance of CIM_ComputerSystem. The Scoping Class shall be CIM_ComputerSystem. The Scoping Instance shall be the Central Instance. Table 1 lists profiles upon which this profile has a dependency.

Note that the behavioral definitions for many of the profiles are inherited from the abstract *Computer System Profile*. Therefore, they are not referenced in Table 1. Examples are the *System Memory Profile* and the *Sensors Profile.*

206 **Table 1 – Referenced Profiles**

Profile Name	Organization	Version	Relationship	Behavior
Computer System	DMTF	1.0	Specializes	None
Fan	DMTF	1.0	Optional	See section 7.2.1.
Physical Asset	DMTF	1.0	Mandatory	See section 7.1.2.
Power State Management	DMTF	1.0	Optional	See section 7.3.2.
Power Supply	DMTF	1.0	Optional	See section 7.2.2.
Profile Registration	DMTF	1.0	Mandatory	None
Simple Identity Management	DMTF	1.0	Optional	See section 7.4.
Role Based Authorization	DMTF	1.0	Optional	See section 7.5.

207 # 6 Description

208 The *Base Desktop and Mobile Profile* is an autonomous profile that defines the minimum top-level object
209 model needed to model monolithic desktop or mobile computer hardware and related software. Other
210 profiles add additional management objects to this basic desktop mobile model to provide system
211 configuration, boot control, and other provisioning capabilities. CIM_ComputerSystem represents the
212 desktop mobile system. CIM_TimeService provides the ability to manage the system time.

213 Figure 1 presents the class schema for the *Base Desktop and Mobile Profile*. For simplicity, the prefix
214 *CIM_* has been removed from the names of the classes.

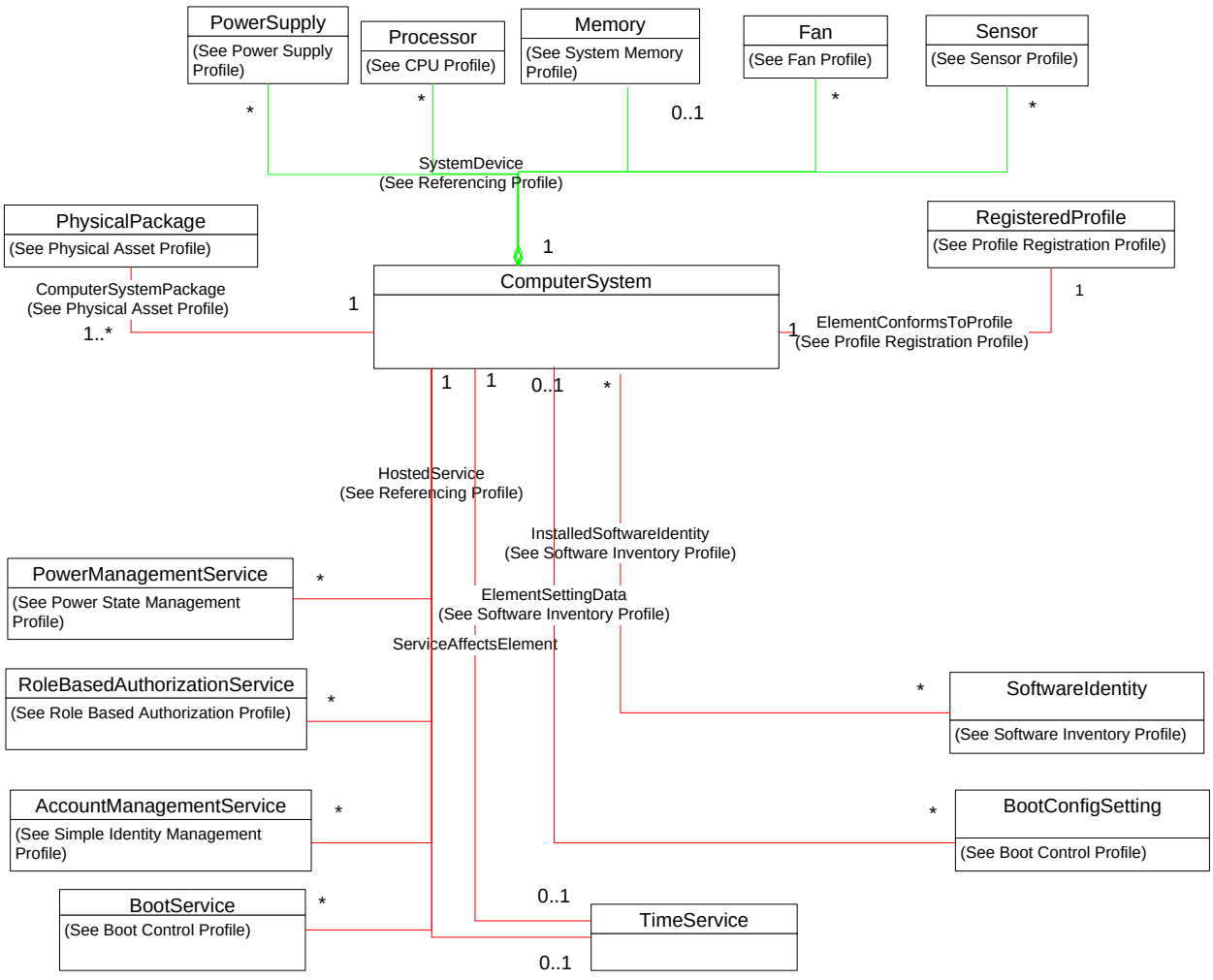


Figure 1 – Base Desktop and Mobile Profile: Class Diagram

Note that the behavioral constraints for many of the profiles identified in Figure 1 are inherited from the specialized *Computer System Profile*. Therefore, although they are shown in Figure 1, they are not referenced in this specification. Examples are the *System Memory Profile* and the *Sensors Profile.*

6.1 Representation of System Power State

Normative requirements for the representation of the power state of a system are expressed in section 7.3. The following informative text provides background on the approach taken to modeling the power state of a system.

The *Base Desktop and Mobile Profile* identifies two complementary approaches to representing the power state of a base desktop and mobile system: simple on/off management through the RequestedState and EnabledState properties, and the RequestStateChange() method. Definitions for the 2 (Enabled) and
3 (Disabled) values for EnabledState use industry-standard ACPI definitions. Alternately, if an implementation wants to support more granular or complex power management behavior; the *Power State Management Profile* can be implemented.

The power management behavior and system power states specified in the *Power State Management Profile* are a superset of the function and states that are represented using the EnabledState and RequestedState properties of CIM_ComputerSystem. That is, the EnabledState and RequestedState

properties are sufficient to represent ACPI states S0 and S5. Implementing the *Power State Management Profile* provides the ability to represent additional ACPI states. Although some of the values of EnabledState and PowerState are equivalent, this equivalency results from them being mapped to identical ACPI states rather than being defined in terms of each other. With this method, for the subset of values for EnabledState and RequestedState for which ACPI states are defined, there is a one-to-one correspondence with a legal value for the PowerState and RequestedPowerState properties.

The method of defining the states that are expressible through the *Power State Management Profile* as a superset of those possible with EnabledState and RequestedState is contrasted with the discarded alternative method of using the implementation of the *Power State Management Profile* to provide a refinement of the interpretation of the EnabledState and RequestedState values. If this latter, discarded method had been used, multiple values of PowerState and RequestedPowerState would have been mapped onto the less granular values for the EnabledState and RequestedState properties.

7 Implementation Requirements

The *Base Desktop and Mobile Profile* consists of definitions for CIM_ComputerSystem, CIM_PhysicalPackage, CIM_TimeService, and their related EnabledLogicalElementCapabilities. Other related subsystem classes such as CIM_LogicalDevice, CIM_Collection, and CIM_RecordLog are defined in their respective profiles.

Requirements for propagating and formulating certain properties of the *Base Desktop and Mobile Profile* classes are discussed in this section. The *Base Desktop and Mobile Profile* defines how to model the system's logical aspects, and the *Physical Asset Profile* defines how to model the system's physical aspects.

The list of all methods can be found in section 8 ("Methods"), and the list of properties can be found in section 10 ("CIM Elements").

7.1 Base Desktop and Mobile System

There shall be an instance of CIM_ComputerSystem to represent the system being modeled.

7.1.1 Identifying a Base Desktop Mobile

This section details constraints beyond those specified in the *Computer System Profile* for using the IdentifyingDescriptions and OtherIdentifyingInfo properties to identify a computer system.

7.1.1.1 CIM:GUID

For each unique value of the PlatformGUID property of an instance of CIM_ComputerSystemPackage that references the CIM_ComputerSystem instance, the IdentifyingDescriptions property shall contain the value "CIM:GUID", and the corresponding array position of the OtherIdentifyingInfo property shall contain the value of the PlatformGUID proprety.

7.1.1.2 CIM:Model:SerialNumber

For each unique combination of the values of the ModelNumber and SerialNumber properties of an associated instance of CIM_PhysicalPackage, the IdentifyingDescriptions property of CIM_ComputerSystem instance shall contain the value " CIM:Model:SerialNumber ". The value of the corresponding OtherIdentifyInfo array index shall be of the form specified in the *Computer System Profile*. The <Model Number> portion of this value shall match the value of the Model property of the CIM_PhysicalPackage instance. The <Serial Number> portion of this value shall match the value of the SerialNumber property of the same CIM_PhysicalPackage instance.

7.1.1.3 CIM:Tag

For each unique value of the Tag property of an associated instance of CIM_PhysicalPackage, the IdentifyingDescriptions property of the CIM_ComputerSystem instance shall contain the value “CIM:Tag”, and the corresponding array position of the OtherIdentifyingInfo property shall contain the value of the Tag property of the CIM_PhysicalPackage instance.

7.1.2 Representing the Physical Packaging

The physical packaging for a system shall be modeled in a way that is compliant with the requirements specified in the *Physical Asset Profile*. At least one instance of CIM_PhysicalPackage shall be associated with the Central Instance through the CIM_ComputerSystemPackage association.

7.2 Management of Base Desktop Mobile Components

The following subclauses detail the requirements for management of components of the system in addition to those specified in the *Computer System Profile*.

7.2.1 Instrumentation of Fans

A system can contain one or more fans that provide cooling for the system. If the fans of the system are instrumented, the instrumentation should be conformant with the *Fan Profile.* If the fans of the system are instrumented in conformance with the *Fan Profile*, and the Central Instance of this profile shall be associated with one of more instances of the Central Class of the *Fan Profile* through the CIM_SystemDevice association.

7.2.2 Instrumentation of Power Supplies

A system can contain one or more power supplies that provide power to the system. If the power supplies of the system are instrumented, the instrumentation should be conformant with the *Power Supply Profile*. If the power supplies of the system are instrumented in conformance with the *Power Supply Profile*, the Central Instance of this profile shall be associated with one or more instances of the Central Class of the *Power Supply Profile* through the CIM_SystemDevice association.

7.3 State Management

This section details further constraints related to state management beyond those specified in the *Computer System Profile*.

7.3.1 Correspondence of System States and ACPI States

The EnabledState property of CIM_ComputerSystem is defined in terms of ACPI values in order to provide meaningful context for the interpretation of values for a computer system realized in hardware. The mappings specified in Table 2 shall be used. Note that the underlying modeled system does not need to support the ACPI specification.

Table 2 – EnabledState and ACPI State Equivalence

EnabledState Value	Corresponding ACPI State
2 (Enabled)	G0 or S0 Working
3 (Disabled)	G2 or S5
9 (Quiesce)	G1, S1, S2, S3, or S4

7.3.2 Power State Management

When the Scoping Instance of the *Power State Management Profile* is the Scoping Instance of this profile, the requirements defined in this section are applicable.

7.3.2.1 Power Management Available to System

Management of the power state of the system may be supported for the system. When the management of the power state is supported, the *Power State Management Profile* shall be implemented and the Central Instance of this profile shall be associated with the Central Instance of the *Power State Management Profile* through the CIM_AssociatedPowerManagement association.

7.3.2.2 Power Management Hosted on System

The system may provide the ability to manage the power state of itself or other systems. When the system provides this ability, the *Power State Management Profile* shall be implemented and the Central Instance of this profile shall be associated with the Central Instance of the *Power State Management Profile* through the CIM_HostedService association.

7.3.3 Relationship between State Management and Power State Management

The behavior in this section is conditional on the implementation of the behavior in section 7.3.2.1. When the optional behavior specified in section 7.3.2.1 is supported, the state management behavior specified in the *Computer System Profile* shall be supported.

Power state management may be supported for a system. One reason for supporting power state management is the need to provide more granular management beyond that available through state management. To ensure consistent semantics for state management regardless of whether power state management is supported, constraints on the interaction of power state management and state management must be established when power state management is supported. This section details these constraints.

Note: The CIM_ComputerSystem.RequestStateChange() method defined in the *Computer System Profile* will cause the values for the CIM_ComputerSystem.EnabledState and CIM_ComputerSystem.RequestedState properties to change. Because of the equivalence requirements stated in the following sections, this change might result in changes to the values of the CIM_AssociatedPowerManagementService.RequestedPowerState and CIM_AssociatedPowerManagementService.PowerState properties. Likewise, the CIM_PowerManagementService.RequestPowerStateChange() method defined in the *Power State Management Profile* will cause the CIM_AssociatedPowerManagementService.RequestedPowerState and CIM_AssociatedPowerManagementService.PowerState properties to change. Because of the equivalence requirements stated in the following sections, this change might result in changes to the values of the CIM_ComputerSystem.EnabledState and CIM_ComputerSystem.RequestedState properties.

7.3.3.1 Relationship between EnabledState and PowerState

Table 3 and Table 4 list equivalency requirements for values of the CIM_ComputerSystem.EnabledState property and the CIM_AssociatedPowerManagementService.PowerState property for the instance of CIM_AssociatedPowerManagementService that references the CIM_ComputerSystem instance. When the CIM_AssociatedPowerManagementService.PowerState property has the value listed in the first column of Table 3, the CIM_ComputerSystem.EnabledState property shall have the value listed in the second column. When the CIM_AssociatedPowerManagementService.PowerState property has the value listed in the first column of Table 4, the CIM_ComputerSystem.EnabledState property should have the value listed in the second column. Note that the set of power states that can be represented by the PowerState property is a superset of those power states that are expressible through the EnabledState property alone. Power states expressible through the PowerState property that are not expressible through the EnabledState property are mapped to 5 (Not Applicable).

357 **Table 3 – PowerState and EnabledState Values (Required Equivalence)**

PowerState Value	Corresponding EnabledState Value
2 (On)	2 (Enabled)
8 (Off – Soft)	3 (Disabled)
12 (Off – Soft Graceful)	3 (Disabled)

358

359 **Table 4 – PowerState and EnabledState Values (Recommended Equivalence)**

PowerState Value	Corresponding EnabledState Value
3 (Sleep – Light)	9 (Quiesce)
4 (Sleep – Deep)	9 (Quiesce)
5 (Power Cycle (Off – Soft))	5 (Not Applicable)
6 (Off – Hard)	3 (Disabled)
7 (Hibernate (Off – Soft))	9 (Quiesce)
9 (Power Cycle (Off – Hard))	5 (Not Applicable)
10 (Master Bus Reset)	5 (Not Applicable)
11 (Diagnostic Interrupt (NMI))	5 (Not Applicable)
13 (Off – Hard Graceful)	3 (Disabled)
14 (Master Bus Reset Graceful)	5 (Not Applicable)
15 (Power Cycle (Off – Soft) Graceful)	5 (Not Applicable)
16 (Power Cycle (Off – Hard) Graceful)	5 (Not Applicable)

360 #### 7.3.3.2 Relationship between RequestedState and RequestedPowerState

361 Table 5 and Table 6 list equivalency requirements for values of the
362 CIM_ComputerSystem.RequestedState property and the
363 CIM_AssociatedPowerManagementService.RequestedPowerState property for the instance of
364 CIM_AssociatedPowerManagementService that references the CIM_ComputerSystem instance. When
365 the CIM_AssociatedPowerManagementService.RequestedPowerState property has the value listed in the
366 first column of Table 5, the CIM_ComputerSystem.RequestedState property shall have the value listed in
367 the second column. When the CIM_AssociatedPowerManagementService.RequestedPowerState
368 property has the value listed in the first column of Table 6, the CIM_ComputerSystem.RequestedState
369 property should have the value listed in the second column. Note that the set of power states that can be
370 represented by the RequestedPowerState property is a superset of those power states that are
371 expressible through the RequestedState property alone. Power states expressible through the
372 RequestedPowerState property that are not expressible through the RequestedState property are
373 mapped to 12 (Not Applicable).

374 **Table 5 – RequestedPowerState and RequestedState Values (Required Equivalence)**

RequestedPowerState Value	Corresponding RequestedState Value
2 (On)	2 (Enabled)
8 (Off – Soft)	3 (Disabled)
13 (Off – Soft Graceful)	3 (Disabled)

375

376 **Table 6 – RequestedPowerState and RequestedState Values (Recommended Equivalence)**

RequestedPowerState Value	Corresponding RequestedState Value
3 (Sleep – Light)	9 (Quiesce)
4 (Sleep – Deep)	9 (Quiesce)
5 (Power Cycle (Off – Soft))	11 (Reset)
6 (Off – Hard)	3 (Disabled)
7 (Hibernate (Off – Soft))	9 (Quiesce)
9 (Power Cycle (Off – Hard))	12 (Not Applicable)
10 (Master Bus Reset)	11 (Reset)
11 (Diagnostic Interrupt (NMI))	12 (Not Applicable)
12 (Not Applicable)	12 (Not Applicable)
14 (Off – Hard Graceful)	3 (Disabled)
15 (Master Bus Reset Graceful)	11 (Reset)
16 (Power Cycle (Off – Soft) Graceful)	11 (Reset)
17 (Power Cycle (Off – Hard) Graceful)	12 (Not Applicable)

377 #### 7.3.3.3 Relationship between RequestedStatesSupported and PowerStatesSupported

378 Table 7 and Table 8 detail equivalency requirements for values of the following properties:

379 • the CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported property for the
380 instance of CIM_EnabledLogicalElementCapabilities that is associated with the
381 CIM_ComputerSystem instance

382 • the CIM_PowerManagementCapabilities.PowerStatesSupported property for the instance of
383 CIM_PowerManagementCapabilities that is associated through CIM_ElementCapabilities with the
384 instance of CIM_PowerManagementService that is associated with the CIM_ComputerSystem
385 instance through the CIM_AssociatedPowerManagementService association

386 When the PowerStatesSupported property contains the value listed in the first column of Table 7, the
387 RequestedStatesSupported property shall contain the value listed in the second column. When the
388 PowerStatesSupported property contains the value listed in the first column of Table 8, the
389 RequestedStatesSupported property should contain the value listed in the second column. The
390 RequestedStatesSupported property may contain additional values that correspond to supported states.
391 The PowerStatesSupported property may contain other values; however, corresponding values for
392 RequestedStatesSupported are not defined. The purpose of the PowerStatesSupported property and
393 RequestedStatesSupported property is to indicate the power state changes that can be initiated through
394 the RequestPowerStateChange() method and the RequestStateChange() method, respectively. The
395 absence of a value from the array indicates the absence of support for that power state change. For those
396 power state changes that can be initiated through the RequestPowerStateChange() method but not
397 through the RequestStateChange() method, no mapping is defined because the absence of a value in the
398 RequestedStatesSupported property implicitly indicates a lack of support for initiating the corresponding
399 power state change.

400 **Table 7 – PowerStatesSupported and RequestedStatesSupported Values (Required Equivalence)**

PowerStatesSupported Value	RequestedStatesSupported Value
2 (On)	2 (Enabled)
8 (Off – Soft)	3 (Disabled)
12 (Off – Soft Graceful)	3 (Disabled)

401

402 **Table 8 – PowerStatesSupported and RequestedStatesSupported Values (Recommended**
403 **Equivalence)**

PowerStatesSupported Value	RequestedStatesSupported Value
3 (Sleep – Light)	9 (Quiesce)
4 (Sleep – Deep)	9 (Quiesce)
5 (Power Cycle (Off – Soft))	11 (Reset)
6 (Off – Hard)	3 (Disabled)
7 (Hibernate (Off – Soft))	9 (Quiesce)
9 (Power Cycle (Off – Hard))	-
10 (Master Bus Reset)	11 (Reset)
11 (Diagnostic Interrupt (NMI))	-
13 (Off – Hard Graceful)	3 (Disabled)
14 (Master Bus Reset Graceful)	11 (Reset)
15 (Power Cycle (Off – Soft) Graceful)	11 (Reset)
16 (Power Cycle (Off – Hard) Graceful)	-

404 ## 7.4 Simple Identity Management

405 A system can represent Account, AcountManagementService, Group, and Identity. If these entities are
406 represented for the system, the instrumentation should be conformant with the *Simple Identity*
407 *Management Profile*. If these entities are instrumented in conformance with the *Simple Identity*
408 *Management Profile,* the Central Instance of the *Base Desktop and Mobile Profile* shall be associated
409 with the Central Instance of the *Simple Identity Management Profile* through the CIM_HostedService
410 association.

411

412 ## 7.5 Role Based Authorization

413 A system can represent Role, RoleBasedAuthenticationService and Privilege. If these entities are
414 represented for the system, the instrumentation should be conformant with the *Role Based Authorization*
415 *Profile*. If these entities are instrumented in conformance with the *Role Based Authorization Profile*, the
416 Central Instance of the *Base Desktop and Mobile Profile* shall be associated with the Central Instance of
417 the *Role Based Authorization Profile* through the CIM_HostedService association.

418 # 8 Methods

419 All intrinsic and extrinsic methods are supported as defined in the *Computer System Profile*.

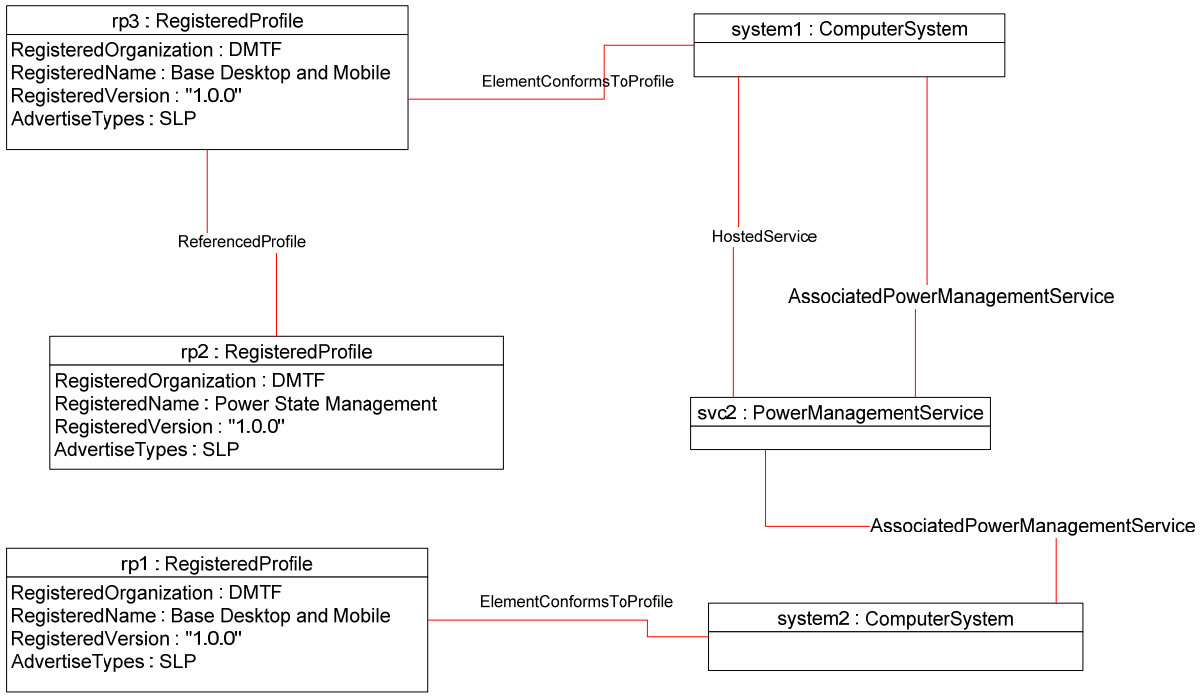
420 # 9 Use Cases

421 The following object diagrams and use cases are based on the implementation conforming to the *Base*
422 *Desktop and Mobile Profile*.

423 ## 9.1 Object Diagrams

424 Figure 2 shows two systems conformant with the *Base Desktop and Mobile Profile*. Both rp3 and rp1
425 advertise the instrumentation of the *Base Desktop and Mobile Profile*. rp2 advertises the existence of the
426 *Power State Management Profile*. rp2 is associated with rp3, which is an instance of
427 CIM_RegisteredProfile that advertises the *Base Desktop and Mobile Profile*. System1 provides power

428 control over itself and system2. The ability to provide power control is modeled by svc2. The *Power State*
429 *Management Profile* is advertised as supported on system1 because that is where the functionality is
430 accessible.



431

432 **Figure 2 – Profile Registration**

433 Figure 3 shows the power management functionality available to system1 and system2. Each system
434 hosts an instance of CIM_TimeService for managing the system's time. System1 has been configured to
435 power on at 8 A.M. EST on August 13, 2006, as indicated by the value of the PowerOnTime property of
436 the instance of CIM_AssociatedPowerManagementService that references system1. This value is relative
437 to the system time as returned by a call to the ManageTime() method of svc1. Note that state
438 management is supported with functional equivalence to the supported power state management.
439 System2 is off and is not configured to come back on.

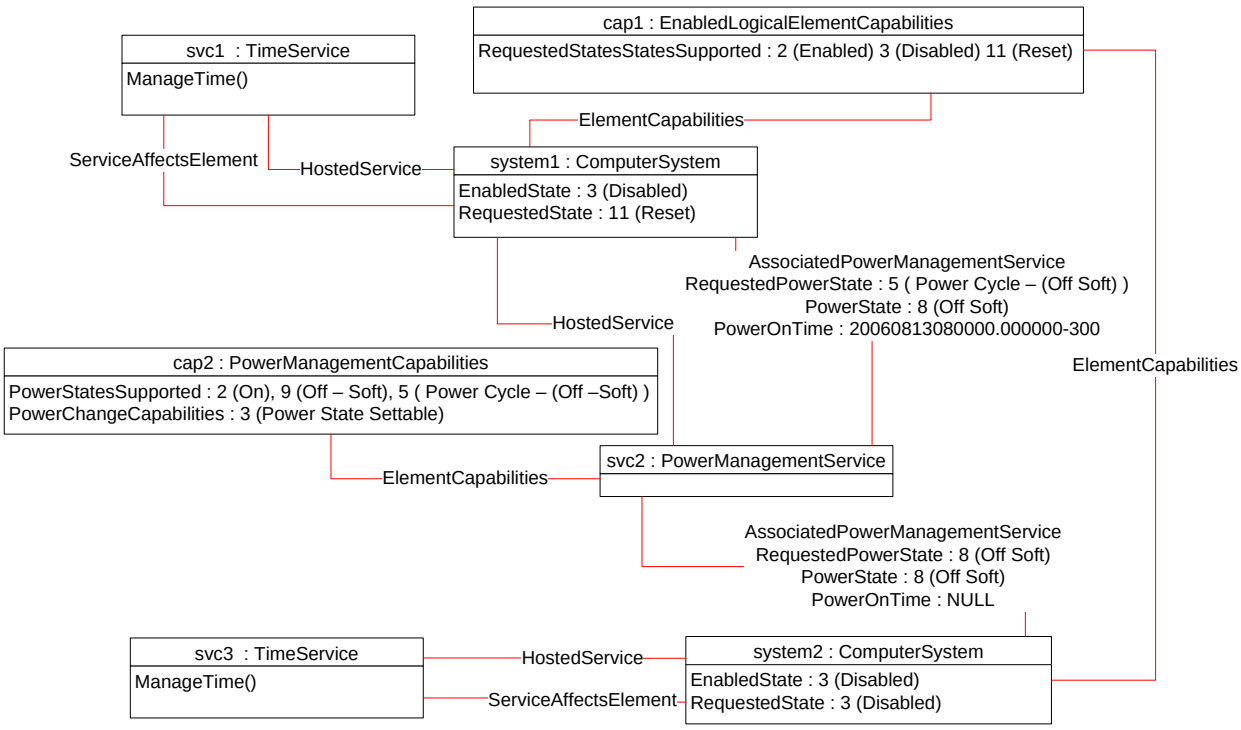
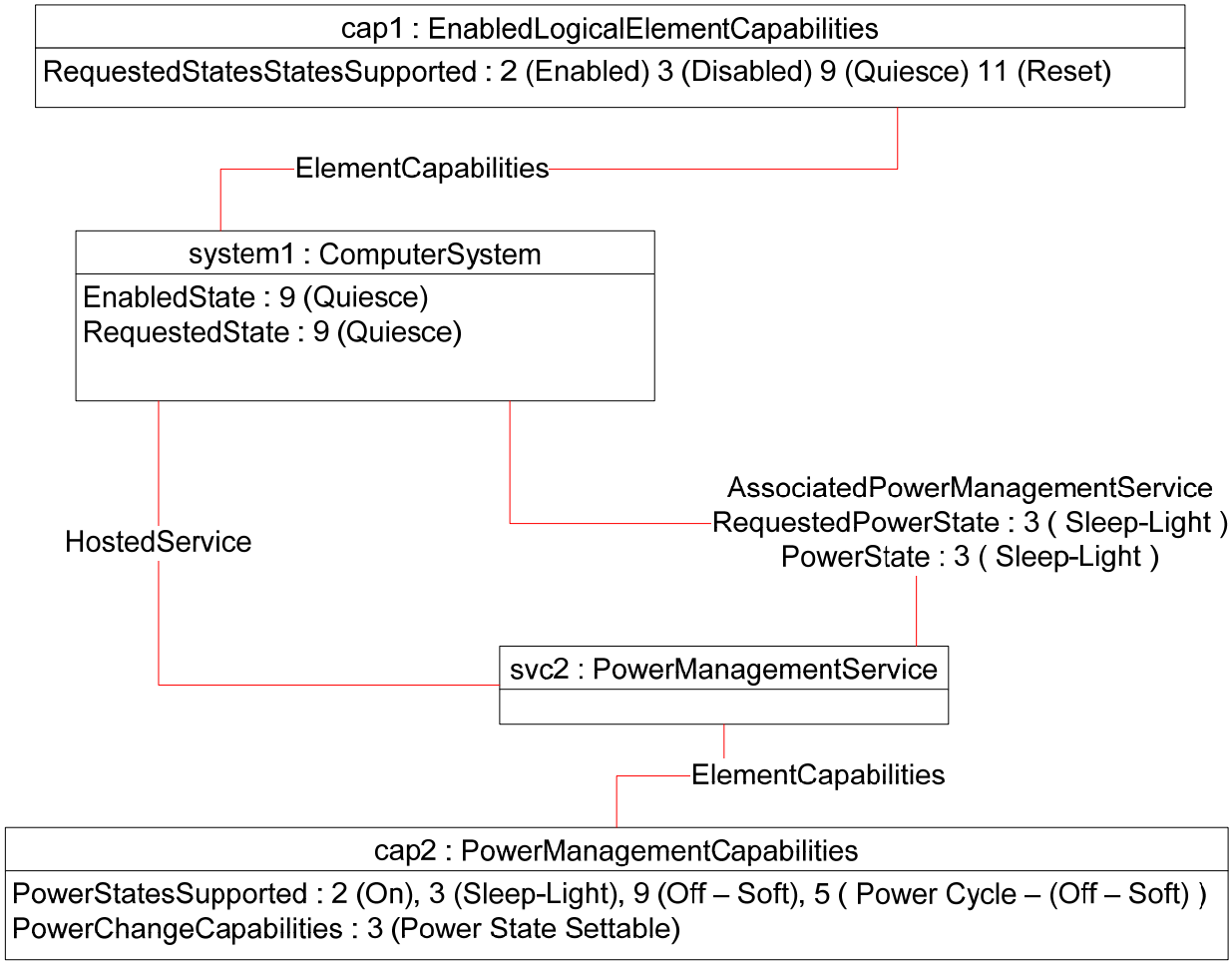


Figure 3 – Power Management and Time Service

Figure 4 shows a system in which the ability to put the system into a sleep-light power state is supported. The sleep-light state is an extended power state that is expressible through the CIM_ComputerSystem.EnabledState property. The CIM_ComputerSystem.EnabledState property has the value 9 (Quiesce) because the current power state of the system is sleep light. If the power state was not sleep light and the current power state mapped to another valid EnabledState value, the EnabledState property would be that value. The actual power state of the system is expressed through the CIM_AssociatedPowerManagementService.PowerState property.

449



450

451 **Figure 4 – Advanced Power Management**

452 Figure 5, Figure 6, and Figure 7 illustrate the logical and physical containment hierarchy of a single
453 system.

454 Figure 5 illustrates the logical hierarchy of components contained in the system. The optional *CPU Profile*,
455 *Fan Profile*, *Power Supply Profile*, *Sensors Profile*, and *System Memory Profile* have been implemented. The
456 system has four processors. Each processor has a dedicated voltage sensor and a dedicated
457 temperature sensor. The system has two power supplies. Each power supply has a dedicated voltage
458 sensor. The system has four fans. Each fan has a dedicated tachometer associated with it. The total
459 system memory available is modeled as well.

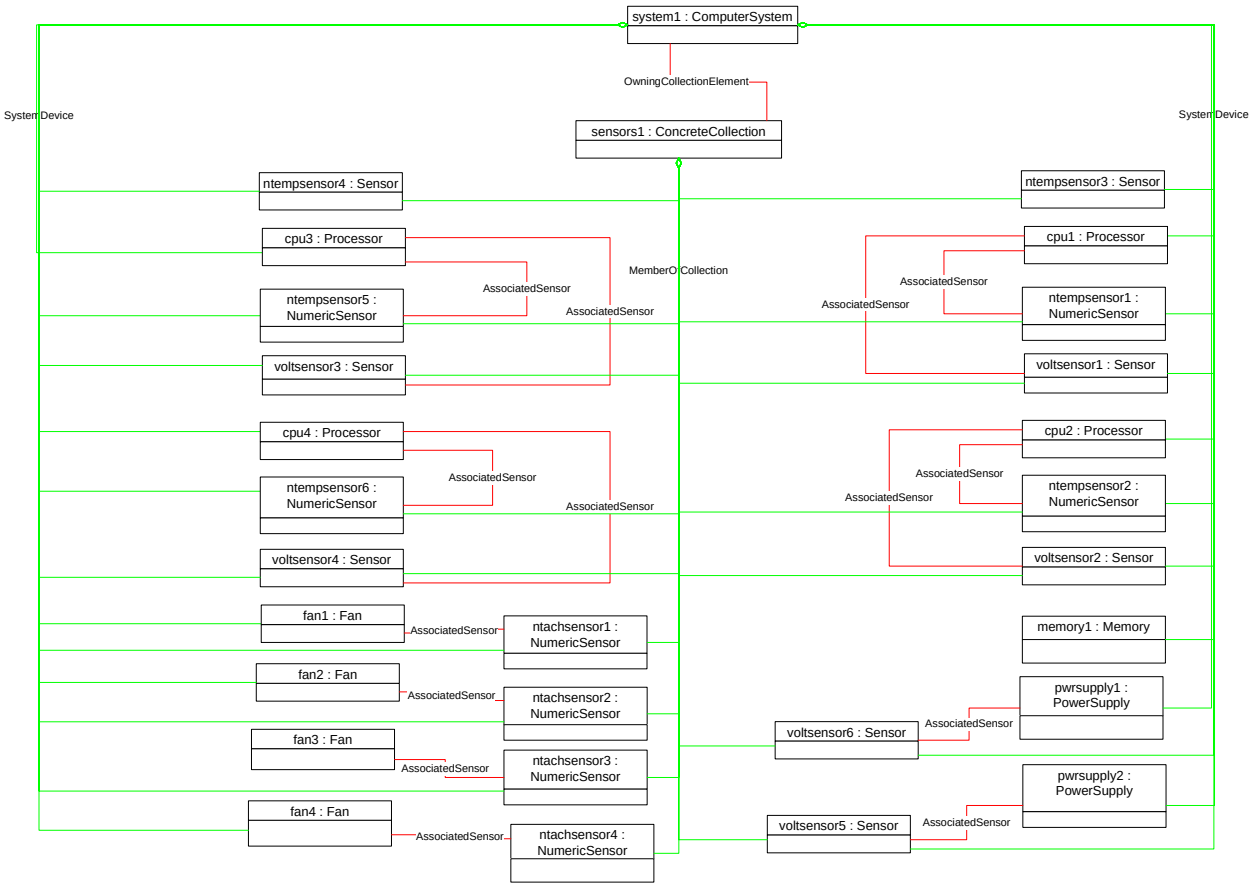


Figure 5 – Logical Topology

Figure 6 shows the physical containment hierarchy for the managed system. The *Physical Asset Profile* has been implemented. The location of the fans within the system is not modeled; instead, they are modeled as being directly contained in the main system chassis. The slots or bays in the main chassis that can contain a power supply are separately modeled (slot5 and slot2). The optional slot and package compatibility behavior of the *Physical Asset Profile* has been implemented for the power supply slots. The system memory is installed in four slots on the main system board (card1). The processors (proc1 – proc4) are installed in pairs on separate cards on the main system card. The capacity of the system for processors, fans, power supplies, and memory is indicated through instances of CIM_ConfigurationCapacity.

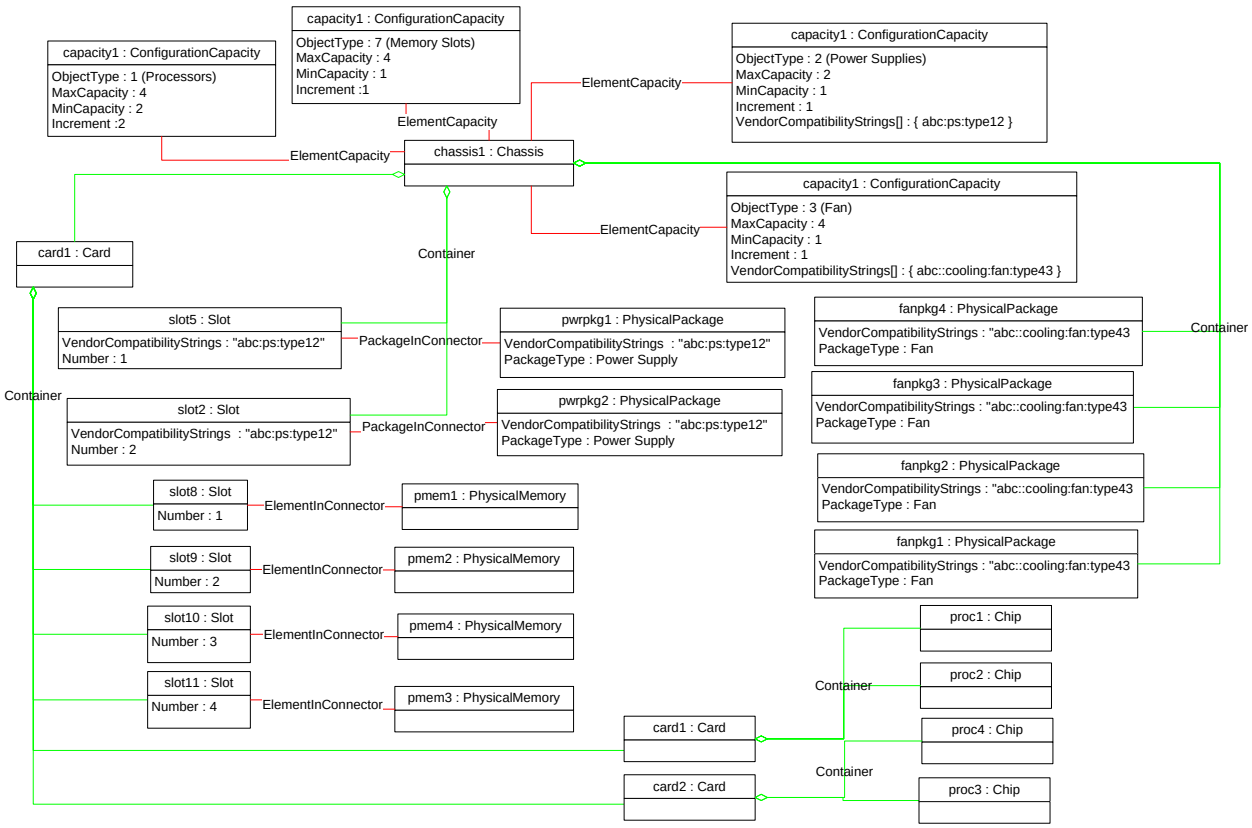


Figure 6 – Physical Topology

Figure 7 shows the relationship between the logical components and their underlying physical packaging. Each fan, power supply, and processor has a dedicated package. The system memory is realized with four physical components. The system itself is packaged in a single chassis. To minimize clutter in the diagram, the CIM_SystemDevice associations have been elided.

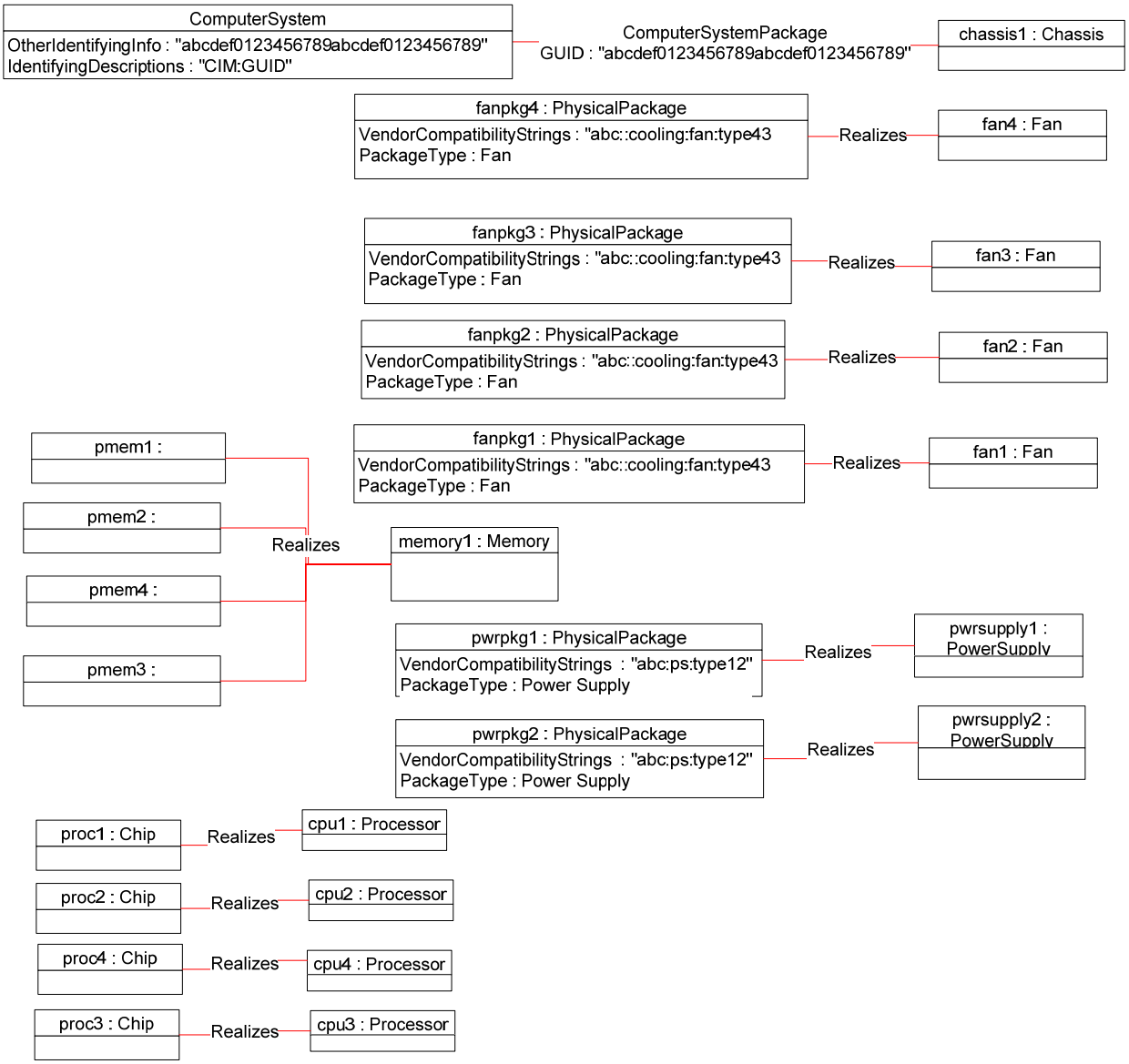


Figure 7 – Logical to Physical Mapping

9.2 Determine the System Model and Serial Number

When the *Physical Asset Profile* and optional asset management have been implemented for the system, a client can determine the system model and serial number as follows:

1) Find an instance of CIM_PhysicalPackage that is associated with the Central Instance through the CIM_ComputerSystemPackage association.
2) Refer to the Model and SerialNumber properties of the instance.

9.3 Power On a System

A client can power on a system as follows:

1) Look for an instance of CIM_EnabledLogicalElementCapabilities that is associated with the target instance through the CIM_ElementCapabilities association.
2) Verify that the CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported property contains the value 2 (Enabled).
3) Invoke the RequestStateChange() method on the target instance, specifying 2 (Enabled) for the RequestedState parameter.

9.4 Power Off a System

A client can power off a system as follows:

1) Look for an instance of CIM_EnabledLogicalElementCapabilities that is associated with the target instance through the CIM_ElementCapabilities association.
2) Verify that the CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported property contains the value 3 (Disabled).
3) Invoke the RequestStateChange() method on the target instance, specifying 3 (Disabled) for the RequestedState parameter.

9.5 Shut Down and Restart a System

A client can shut down and restart a system as follows:

1) Look for an instance of CIM_EnabledLogicalElementCapabilities that is associated with the target instance through the CIM_ElementCapabilities association.
2) Verify that the CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported property contains the value 11 (Reset).
3) Invoke the RequestStateChange() method on the target instance, specifying 11 (Reset) for the RequestedState parameter.

9.6 Perform System Power Control

A client might need to perform power control that is more granular than the functionality available through state management. This power control is done through power state management. A client can determine whether power state management is available for the system by searching for an instance of CIM_PowerManagementService that is associated with the Central Instance through the CIM_AssociatedPowerManagementService association. The specific use cases for performing power state management are documented in the *Power State Management Profile*.

9.7 Determining the System Power State

A client can determine the power state of the system as follows:

1) Query the CIM_ComputerSystem.EnabledState property.

   If the property has the value 2 (Enabled), the system is currently in ACPI state S0 (or equivalent in a non-ACPI system). If the property has the value 3 (Disabled), the system is currently in ACPI state S5 (or equivalent in a non-ACPI system).
2) If the CIM_ComputerSystem.EnabledState property has the value 5 (Not Applicable), find the instance of CIM_AssociatedPowerManagementService that references the CIM_ComputerSystem instance.

3) Query the value of the CIM_AssociatedPowerManagementService.PowerState property. The *Power State Management Profile* details the equivalent ACPI states for each value.

9.8 Determine the Number of Processors in the System

When the optional *CPU Profile* is implemented, the client can determine the number of processors in the system by querying for instances of CIM_Processor that are associated with the Central Instance through the CIM_SystemDevice association.

The client can use these same steps to find the fans and power supplies installed in the system, substituting the *Fan Profile* and CIM_Fan, and the *Power Supply Profile* and CIM_PowerSupply, appropriately.

9.9 Determine the Number of Processors That the System Can Hold

When the optional configuration capacity behavior from the *Physical Asset Profile* is implemented for processors for the system, a client can determine the number of processors that the system can hold as follows:

1) Find instances of CIM_PhysicalPackage that are associated with the Central Instance through the CIM_ComputerSystemPackage association.

2) For each instance of CIM_PhysicalPackage, find the instances of CIM_ConfigurationCapacity that are associated with the CIM_PhysicalPackage instance through the CIM_ElementCapacity association.

3) For each instance of CIM_ConfigurationCapacity, if the ObjectType property has the value 1 (Processors), query the MaximumCapacity property and add the value to the total number of processors that the system can hold.

The client can also apply these steps to find the total amount of physical memory, fans, and power supplies that the system can hold when the configuration capacity has been instrumented for objects of that type by substituting the appropriate value for 1 (Processors) in step 3.

10 CIM Elements

Table 9 shows the instances of CIM Elements for this profile. Instances of the CIM Elements shall be implemented as described in Table 9. Sections 7 (“Implementation Requirements”) and 8 (“Methods”) may impose additional requirements on these elements.

Table 9 – CIM Elements: Base Desktop and Mobile Profile

Element Name	Requirement	Description
Classes		
CIM_ComputerSystem	Mandatory	See section 10.1
CIM_ComputerSystemPackage	Mandatory	See section 10.2
CIM_EnabledLogicalElementCapabilities	Optional	See section 10.3
CIM_PhysicalPackage	Mandatory	See section 10.4
CIM_RegisteredProfile	Mandatory	See section 10.5
Indications		
None defined in this profile		

10.1 CIM_ComputerSystem

An instance of CIM_ComputerSystem is used to represent the system. Table 10 defines the requirements for elements of this class.

Table 10 – Class: CIM_ComputerSystem

Elements	Requirement	Notes
EnabledState	Mandatory	See sections 7.3.3.1 and 7.3.1.
RequestedState	Mandatory	See section 7.3.3.2.
Dedicated	Mandatory	This property shall have the value 32 (“Desktop”) or the value 33 (“Laptop”).

10.2 CIM_ComputerSystemPackage

One or more instances of CIM_ComputerSystemPackage are used to associate the CIM_ComputerSystem instance with the CIM_PhysicalPackage instances in which it resides. The constraints specified in Table 11 are in addition to those specified in the *Physical Asset Profile*.

Table 11 – Class: CIM_ComputerSystemPackage

Elements	Requirement	Notes
Dependent	Mandatory	Shall be a reference to the Central Instance Cardinality 1
Antecedent	Mandatory	Shall be a reference to CIM_PhysicalPackage Cardinality 1..*

10.3 CIM_EnabledLogicalElementCapabilities

CIM_EnabledLogicalElementCapabilities is used to indicate support for managing the state of the system. Table 12 defines the requirements for elements of this class.

Table 12 – Class: CIM_EnabledLogicalElementCapabilities

Elements	Requirement	Notes
RequestedStatesSupported	Mandatory	See section 7.3.3.3

10.4 CIM_PhysicalPackage

One or more instances of CIM_PhysicalPackage represent the physical packaging of the computer system. Other than the existence of at least one, this profile does not specify any constraints for CIM_PhysicalPackage beyond those specified in the *Physical Asset Profile*.

10.5 CIM_RegisteredProfile

CIM_RegisteredProfile identifies the *Base Desktop and Mobile Profile* so that a client can determine whether an instance of CIM_ComputerSystem is conformant with this profile. CIM_RegisteredProfile is defined by the *Profile Registration Profile*. With the exception of the mandatory values specified for the elements in Table 13, the behavior of the RegisteredProfile instance is per the *Profile Registration Profile*.

577 **Table 13 – Class: CIM_RegisteredProfile**

Elements	Requirement	Notes
RegisteredName	Mandatory	This property shall have a value of “Base Desktop and Mobile”.
RegisteredVersion	Mandatory	This property shall have a value of “1.0.0”.
RegisteredOrganization	Mandatory	This property shall have a value of 2 (DMTF).

ANNEX A
(Informative)

Change Log

Version	Date	Description
0.8.0	2006/07/13	Initial draft
1.0.0	2008/11/24	Final release.
1.0.0	2008/12/9	Final release after addressing Platform SC comments.

ANNEX B
(informative)

Acknowledgments

The authors wish to acknowledge the following people, including all the editors and contributors of the *Base Server Profile*, on which this profile is based.

Editor:

- Hemal Shah - Broadcom
- Jianwen Yin – Dell

Contributors:

- Jon Hass – Dell
- Khachatur Papanyan – Dell
- Jeff Hilland – HP
- Christina Shaw – HP
- Aaron Merkin – IBM
- Perry Vincent – Intel
- John Leung – Intel
- Jianwen Yin – Dell
- Hemal Shah - Broadcom