Document Number: DSP1004

Date: 2010-04-22

Version: 1.0.1

Base Server Profile

Document Type: Specification

Document Status: DMTF Standard

Document Language: US-en

CONTENTS

Figures

Tables

Foreword

The *Base Server Profile* (DSP1004) 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. For information about the DMTF, see http://www.dmtf.org.

Acknowledgments

The DMTF acknowledges the following individuals for their contributions to this document:

Editors:

- Christina Shaw – HP
- Aaron Merkin – IBM
- Jeff Hilland – HP

Contributors:

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

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 basic server and subsystems that are modeled 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 Server Profile

1 Scope

The *Base Server Profile* is the autonomous profile that defines the classes used to describe basic server hardware and its related software. The scope of this profile is limited to simple servers that are directly realized in physical components. The profiles referenced by the *Base Server Profile* extend the management capabilities by adding the capability to represent server configuration, boot control, provisioning, and hardware.

2 Normative References

The following referenced documents are indispensable for the application of this document. For dated or versioned references, only the edition cited (including any corrigenda or DMTF update versions) applies. For references without a date or version, the latest published edition of the referenced document (including any corrigenda or DMTF update versions) applies.

Advanced Configuration and Power Interface Specification (ACPI Specification), revision 3.0, www.acpi.info/Downloads/ACPIspec30.pdf

DMTF DSP0004, *CIM Infrastructure Specification 2.5*, http://www.dmtf.org/standards/published_documents/DSP0004_2.5.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 DSP1005, *CLP Service Profile 1.0*, http://www.dmtf.org/standards/published_documents/DSP1005_1.0.pdf

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

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

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

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

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

DMTF DSP1013, *Fan Profile 1.0,* http://www.dmtf.org/standards/published_documents/DSP1013_1.0.pdf

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

DMTF DSP1015, *Power Supply Profile 1.0*, http://www.dmtf.org/standards/published_documents/DSP1015_1.0.pdf

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

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

DMTF DSP1022, *CPU Profile 1.0*,
http://www.dmtf.org/standards/published_documents/DSP1022_1.0.pdf

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

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

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

DMTF DSP1026, *System Memory Profile 1.0*,
http://www.dmtf.org/standards/published_documents/DSP1026_1.0.pdf

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

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

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

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

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

DMTF DSP1052, *Computer System Profile 1.0*,
http://www.dmtf.org/standards/published_documents/DSP1052_1.0.pdf

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

3 Terms and Definitions

In this document, some terms have a specific meaning beyond the normal English meaning. Those terms are defined in this clause.

The terms "shall" ("required"), "shall not," "should" ("recommended"), "should not" ("not recommended"), "may," "need not" ("not required"), "can" and "cannot" in this document are to be interpreted as described in ISO/IEC Directives, Part 2, Annex H. The terms in parenthesis are alternatives for the preceding term, for use in exceptional cases when the preceding term cannot be used for linguistic reasons. Note that ISO/IEC Directives, Part 2, Annex H specifies additional alternatives. Occurrences of such additional alternatives shall be interpreted in their normal English meaning.

The terms "clause," "subclause," "paragraph," and "annex" in this document are to be interpreted as described in ISO/IEC Directives, Part 2, Clause 5.

The terms "normative" and "informative" in this document are to be interpreted as described in ISO/IEC Directives, Part 2, Clause 3. In this document, clauses, subclauses, or annexes labeled "(informative)" do not contain normative content. Notes and examples are always informative elements.

The terms defined in DSP0004, DSP0200, and DSP1001 apply to this document.

4 Symbols and Abbreviated Terms

The following abbreviations are used in this document.

4.1
ACPI
Advanced Configuration and Power Interface

5 Synopsis

Profile Name: Base Server

Version: 1.0.1

Organization: DMTF

CIM Schema Version: 2.13

Specializes: DMTF *Computer System Profile* 1.0

Central Class: CIM_ComputerSystem

Scoping Class: CIM_ComputerSystem

The *Base Server Profile* is an autonomous profile that provides the capability to manage simple server hardware and related software.

The Central Class of the *Base Server 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.

Table 1 – Referenced Profiles

Profile Name	Organization	Version	Relationship	Behavior
Computer System	DMTF	1.0	Specializes	None
Fan Profile	DMTF	1.0	Optional	See 7.2.1.
Physical Asset	DMTF	1.0	Mandatory	See 7.1.2.
Power State Management	DMTF	1.0	Optional	See 7.3.2.
Power Supply	DMTF	1.0	Optional	See 7.2.2.
Profile Registration	DMTF	1.0	Mandatory	None
Text Console Redirection	DMTF	1.0	Optional	See 7.4.

6 Description

The *Base Server Profile* is an autonomous profile that defines the minimum top-level object model needed to model simple server hardware and related software. Other profiles add additional management objects to this basic server model to provide system configuration, boot control, and other provisioning capabilities. CIM_ComputerSystem represents the server system. CIM_TimeService provides the ability to manage the system time.

Figure 1 presents the class schema for the *Base Server Profile*. For simplicity, the prefix CIM_ has been removed from the names of the classes.

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 *IP Interface Profile*, *Ethernet Port Profile*, and *Record Log Profile*.

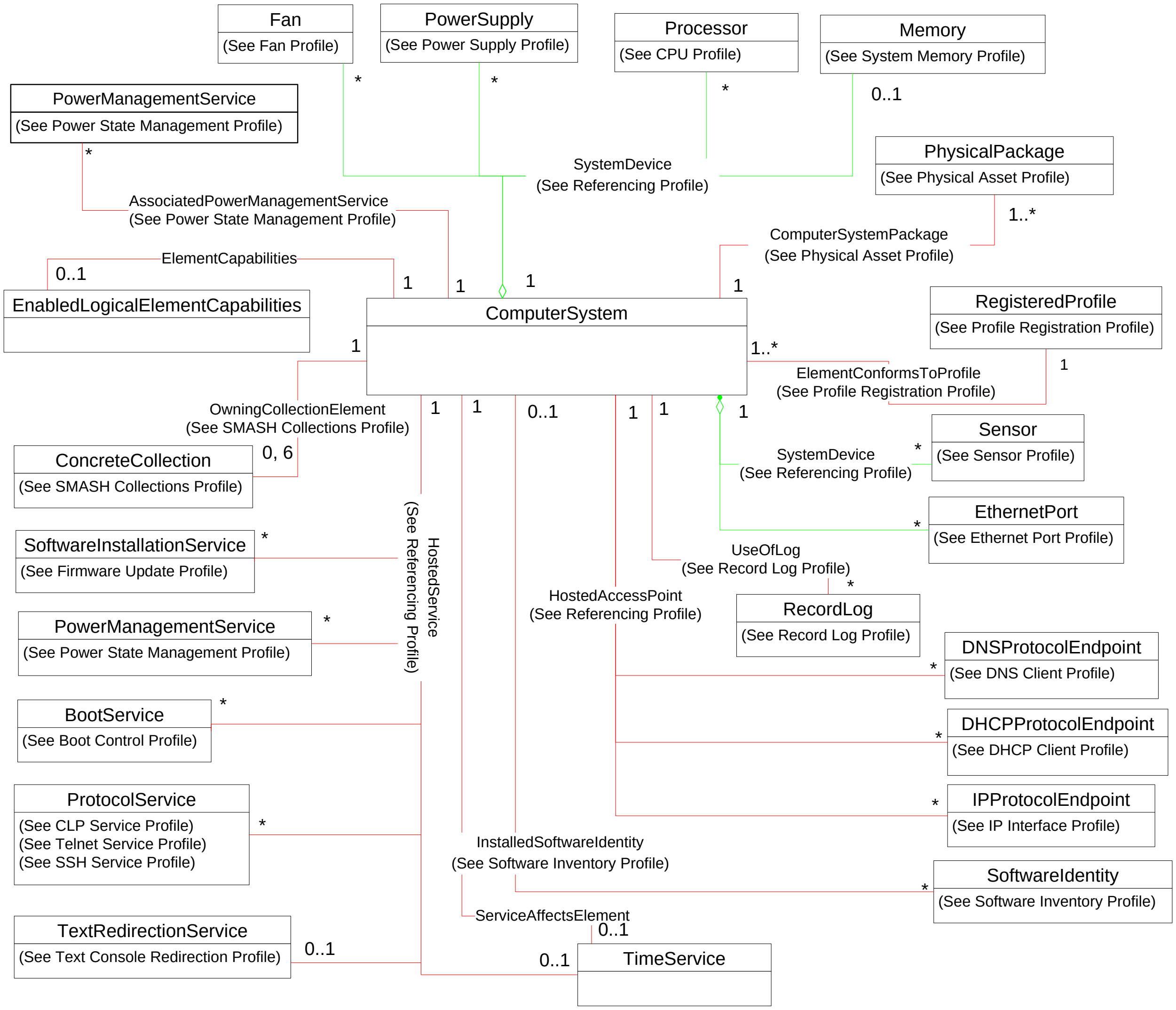

Fan
(See Fan Profile)
PowerSupply
(See Power Supply Profile)
Processor
(See CPU Profile)
Memory
(See System Memory Profile)
PowerManagementService
(See Power State Management Profile)
*
*
*
0..1
SystemDevice
(See Referencing Profile)
PhysicalPackage
(See Physical Asset Profile)
*
AssociatedPowerManagementService
(See Power State Management Profile)
1..*
ComputerSystemPackage
(See Physical Asset Profile)
ElementCapabilities
0..1
EnabledLogicalElementCapabilities
1
1
1
1
ComputerSystem
RegisteredProfile
(See Profile Registration Profile)
1
1..*
ElementConformsToProfile
(See Profile Registration Profile)
1
OwningCollectionElement
(See SMASH Collections Profile)
1
1
0..1
1
1
1
Sensor
(See Sensor Profile)
ConcreteCollection
(See SMASH Collections Profile)
0, 6
SystemDevice
(See Referencing Profile)
*
EthernetPort
(See Ethernet Port Profile)
*
SoftwareInstallationService
(See Firmware Update Profile)
*
HostedService
(See Referencing Profile)
UseOfLog
(See Record Log Profile)
*
HostedAccessPoint
(See Referencing Profile)
RecordLog
(See Record Log Profile)
PowerManagementService
(See Power State Management Profile)
*
DNSProtocolEndpoint
(See DNS Client Profile)
*
BootService
(See Boot Control Profile)
*
DHCPProtocolEndpoint
(See DHCP Client Profile)
*
IPProtocolEndpoint
(See IP Interface Profile)
*
ProtocolService
(See CLP Service Profile)
(See Telnet Service Profile)
(See SSH Service Profile)
*
InstalledSoftwareIdentity
(See Software Inventory Profile)
SoftwareIdentity
(See Software Inventory Profile)
*
ServiceAffectsElement
0..1
TextRedirectionService
(See Text Console Redirection Profile)
0..1
0..1
TimeService

Figure 1 – Base Server Profile: Class Diagram

6.1 Representation of System Power State

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

The *Base Server Profile* identifies two complementary approaches to representing the power state of a base server: simple on/off management through the RequestedState and EnabledState properties, and the RequestStateChange( ) method. Definitions are given for the 2 (Enabled) and 3 (Disabled) values for the EnabledState property in terms of 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. For example, the equivalency between the EnabledState and PowerState values results from their mapping to identical ACPI states rather than that they are defined in terms of each other. For the subset of values for the EnabledState and RequestedState properties for which ACPI states are defined, there is a one-to-one correspondence with a legal value for the PowerState and RequestedPowerState properties.

Defining the states expressible through the *Power State Management Profile* as a superset of those states possible with the EnabledState and RequestedState properties is contrasted with the discarded alternative 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 approach were taken, multiple values of PowerState and RequestedPowerState would be mapped to the less granular values for the EnabledState and RequestedState properties.

7 Implementation

The *Base Server Profile* consists of definitions for the CIM_ComputerSystem, CIM_PhysicalPackage, and CIM_TimeService classes, and their related EnabledLogicalElementCapabilities classes. 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 Server Profile* classes are discussed in this clause. The *Base Server Profile* is divided into two areas of functionality: the logical aspects of the server system and its physical aspects. This profile defines how to model the system’s logical aspects, and the *Physical Asset Profile* defines how to model its physical aspects.

Methods are described in clause 8 (“Methods”), and properties are described in clause 10 (“CIM Elements”).

7.1 Base Server System

The instrumentation shall create an instance of CIM_ComputerSystem to represent the system being modeled.

7.1.1 Identifying a Base Server

This clause details the 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

The value of the OtherIdentifyingInfo property shall match the value of the CIM_ComputerSystemPackage.PlatformGUID property for an instance of CIM_ComputerSystemPackage that references the Central Instance.

7.1.1.2 CIM:Model:SerialNumber

The value of the OtherIdentifyingInfo property shall match the value of the Model property of an instance of CIM_PhysicalPackage, concatenated with a single colon (:), concatenated with the value of the SerialNumber property of the same instance of CIM_PhysicalPackage.

7.1.1.3 CIM:Tag

The value of the OtherIdentifyingInfo property shall match the value of the Tag property of an instance of CIM_PhysicalPackage.

7.1.2 Representing the Physical Packaging

The physical packaging for a system shall be modeled according to 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 Server 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 (Optional)

A system can contain one or more fans that provide cooling for the system. When the fans of the system are instrumented, the instrumentation shall be conformant with the *Fan Profile*, and the Central Instance of the *Base Server Profile* shall be associated with the Central Instance of the *Fan Profile* through the CIM_SystemDevice association.

7.2.2 Instrumentation of Power Supplies (Optional)

A system can contain one or more power supplies that provide power to the system. When the power supplies of the system are instrumented, the instrumentation shall be conformant with the *Power Supply Profile*, and the Central Instance of the *Base Server Profile* shall be associated with the Central Instance of the *Power Supply Profile* through the CIM_SystemDevice association.

7.3 State Management

This clause 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 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. It is not necessary for the underlying modeled system to support the ACPI specification.

Table 2 – EnabledState and ACPI State Equivalence

CIM_ComputerSystem.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

The *Power State Management Profile* may be supported because the Central Instance either hosts an instance of CIM_PowerManagementService or has the functionality of one available to it.

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 the *Base Server Profile* shall be associated with the Central Instance of the *Power State Management Profile* through the CIM_AssociatedPowerManagementService 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 the *Base Server 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 clause is conditional on the implementation of the behavior in 7.3.2.1. When the optional behavior specified in 7.3.2.1 is supported, the state management behavior specified in clause "State Management Is Supported (Conditional)" of the *Computer System Profile* shall be supported.

Management of the power state 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, it is necessary to establish constraints on the interaction of power state management and state management when power state management is supported. This clause details these constraints.

The CIM_ComputerSystem.RequestStateChange( ) method defined in the *Computer System Profile* causes the values for the CIM_ComputerSystem.EnabledState and CIM_ComputerSystem.RequestedState properties to change. Due to the equivalence requirements stated in 7.3.3.1, 7.3.3.2, and 7.3.3.3, the possible invocation of the method will 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. Due to the equivalence requirements stated in 7.3.3.1, 7.3.3.2, and 7.3.3.3, it is possible that this will 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 detail the 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, 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. 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. Power states expressible through the PowerState property that are not expressible through the EnabledState property are mapped to 5 (Not Applicable).

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)

Table 4 – EnabledState and PowerState 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)

7.3.3.2 Relationship between RequestedState and RequestedPowerState

Table 5 details equivalency requirements for the values of the CIM_ComputerSystem.RequestedState property and the CIM_AssociatedPowerManagementService.RequestedPowerState property for the instance of CIM_AssociatedPowerManagementService that references the CIM_ComputerSystem instance. When the CIM_AssociatedPowerManagementService.RequestedPowerState property has the value listed in the first column, the CIM_ComputerSystem.RequestedState property shall have the value listed in the second column. The set of power states that can be represented by the RequestedPowerState property is a superset of those power states that are expressible through the RequestedState property. Power states expressible through the RequestedPowerState property that are not expressible through the RequestedState property are mapped to 12 (Not Applicable).

Table 5 – RequestedState and RequestedPowerState Values

RequestedPowerState Value	Corresponding RequestedState Value
2 (On)	2 (Enabled)
3 (Sleep-Light)	12 (Not Applicable)
4 (Sleep-Deep)	12 (Not Applicable)
5 (Power Cycle (Off-Soft))	11 (Reset)
6 (Power Cycle (Off-Hard))	12 (Not Applicable)
7 (Hibernate (Off-Soft))	12 (Not Applicable)
8 (Off – Hard)	12 (Not Applicable)
9 (Off – Soft)	3 (Disabled)
10 (Master Bus Reset)	12 (Not Applicable)
11 (Diagnostic Interrupt (NMI))	12 (Not Applicable)

7.3.3.3 Relationship between RequestedStatesSupported and PowerStatesSupported

Table 6 details equivalency requirements for values of the following properties:

- the CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported property for the instance of CIM_EnabledLogicalElementCapabilities that is associated with the CIM_ComputerSystem instance
- the CIM_PowerManagementCapabilities.PowerStatesSupported property for the instance of CIM_PowerManagementCapabilities that is associated through CIM_ElementCapabilities with the instance of CIM_PowerManagementService that is associated with the CIM_ComputerSystem instance through the CIM_AssociatedPowerManagementService association

When the CIM_PowerManagementCapabilities.PowerStatesSupported property contains the value listed in the first column, the CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported property shall contain the value listed in the second column. The RequestedStatesSupported property may contain additional values that correspond to supported states. The PowerStatesSupported property may contain other values; however, corresponding values for the RequestedStatesSupported property are not defined.

The purpose of the PowerStatesSupported and RequestedStatesSupported properties is to indicate the power state changes that can be initiated through the RequestPowerStateChange( ) method and the RequestStateChange( ) method, respectively. The absence of a value from the array indicates the absence of support for that power state change. For those power state changes that can be initiated through the RequestPowerStateChange( ) method and not through the RequestStateChange( ) method, no mapping is defined because the absence of a value in the RequestedStatesSupported property implicitly indicates a lack of support for initiating the corresponding power state change.

Table 6 – RequestedStatesSupported and PowerStatesSupported Values

PowerStatesSupported Value	RequestedStatesSupported Value
0 (On)	2 (Enabled)
4 (Power Cycle (Off-Soft))	11 (Reset)
3 (Off – Soft)	3 (Disabled)

7.4 Text Console Redirection (Optional)

This clause details requirements for the implementation of the *Text Console Redirection Profile*.

7.4.1 Text Console Redirection Available to the System

Redirection of a text console may be supported for the system. When the redirection of a text console is supported, the requirements specified in this clause shall be met.

The *Text Console Redirection Profile* shall be implemented. The Central Instance of the *Base Server Profile* shall be associated with the CIM_TextRedirectionSAP instance of the *Text Console Redirection Profile* through the CIM_SAPAvailableForElement association. The Central Instance of the *Base Server Profile* shall be associated with the Central Instance of the *Text Console Redirection Profile* through the CIM_ServiceAffectsElement association.

7.4.2 Text Console Redirection Provided by the System

The system may provide support for the redirection of a text console for itself or other systems. When the system provides this support, the requirements specified in this clause shall be met.

The *Text Console Redirection Profile* shall be implemented. The Central Instance of the *Base Server Profile* shall be associated with the Central Instance of the *Text Console Redirection Profile* through the CIM_HostedService association. The Central Instance of the *Base Server Profile* shall be associated with one or more instances of CIM_TextRedirectionSAP implemented conformant with the *Text Console Redirection Profile* through the CIM_HostedAccessPoint association.

8 Methods

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

9 Use Cases

The following use cases are based on the implementation conforming to the DMTF *Base Server Profile*.

9.1 Object Diagrams

Figure 2 shows two systems conformant with the *Base Server Profile*. rp3 and rp1 both advertise the instrumentation of the *Base Server Profile*. rp2 advertises the existence of the *Power State Management Profile* and is associated with rp3, which is an instance of CIM_RegisteredProfile that advertises the *Base Server Profile*. system1 provides power control over itself and system2. The ability to provide power control is modeled by svc2. The *Power State Management Profile* is advertised as supported on system1 because that is where the functionality is hosted.

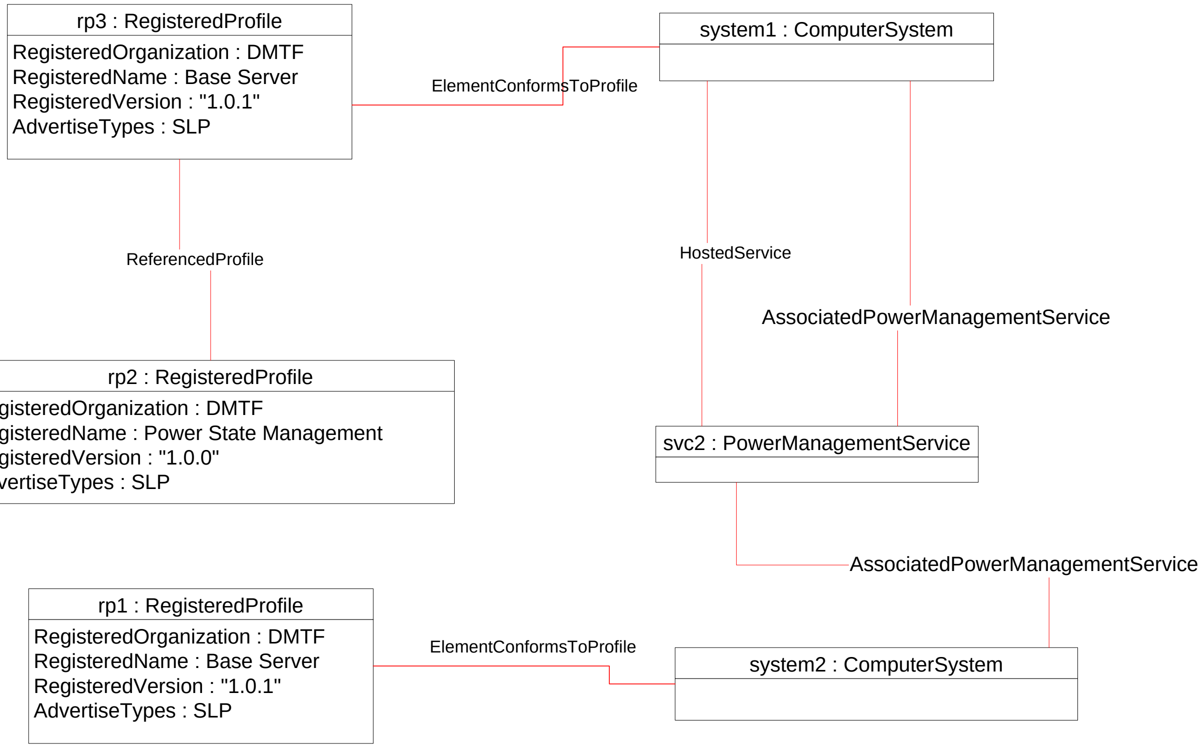
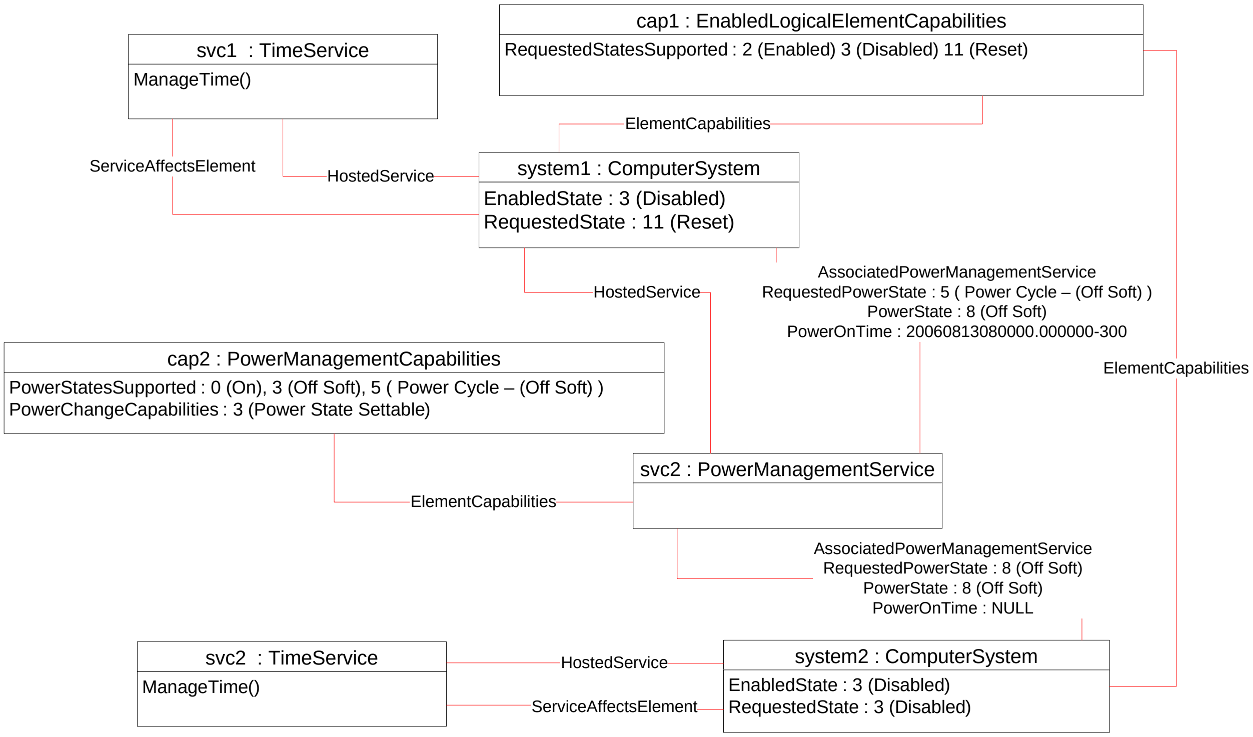

rp3 : RegisteredProfile
RegisteredOrganization : DMTF
RegisteredName : Base Server
RegisteredVersion : "1.0.1"
AdvertiseTypes : SLP
ElementConformsToProfile
system1 : ComputerSystem
ReferencedProfile
HostedService
AssociatedPowerManagementService
rp2 : RegisteredProfile
RegisteredOrganization : DMTF
RegisteredName : Power State Management
RegisteredVersion : "1.0.0"
AdvertiseTypes : SLP
svc2 : PowerManagementService
AssociatedPowerManagementService
rp1 : RegisteredProfile
RegisteredOrganization : DMTF
RegisteredName : Base Server
RegisteredVersion : "1.0.1"
AdvertiseTypes : SLP
ElementConformsToProfile
system2 : ComputerSystem

Figure 2 – Profile Registration

Figure 3 shows the power management functionality available for system1 and system2. Each system hosts an instance of CIM_TimeService for managing the system's time. system1 has been configured to

425 power on at 8 A.M. EST on August 13, 2006, as indicated by the value of the PowerOnTime property of
426 the instance of CIM_AssociatedPowerManagementService that references system1. This value is relative
427 to the system time as returned by a call to the ManageTime( ) method of svc1. State management is
428 supported with functional equivalence to the supported power state management.

429 
svc1 : TimeService
ManageTime()
cap1 : EnabledLogicalElementCapabilities
RequestedStatesSupported : 2 (Enabled) 3 (Disabled) 11 (Reset)
ElementCapabilities
ServiceAffectsElement
HostedService
system1 : ComputerSystem
EnabledState : 3 (Disabled)
RequestedState : 11 (Reset)
HostedService
AssociatedPowerManagementService
RequestedPowerState : 5 (Power Cycle – (Off Soft))
PowerState : 8 (Off Soft)
PowerOnTime : 20060813080000.000000-300
ElementCapabilities
cap2 : PowerManagementCapabilities
PowerStatesSupported : 0 (On), 3 (Off Soft), 5 (Power Cycle – (Off Soft))
PowerChangeCapabilities : 3 (Power State Settable)
svc2 : PowerManagementService
ElementCapabilities
AssociatedPowerManagementService
RequestedPowerState : 8 (Off Soft)
PowerState : 8 (Off Soft)
PowerOnTime : NULL
svc2 : TimeService
ManageTime()
HostedService
ServiceAffectsElement
system2 : ComputerSystem
EnabledState : 3 (Disabled)
RequestedState : 3 (Disabled)

430 **Figure 3 – Power Management and Time Service**

Figure 4 shows a system where 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 not expressible through the CIM_ComputerSystem.EnabledState property. Thus the CIM_ComputerSystem.EnabledState property has the value 5 (Not Applicable). The actual power state of the system is expressed through the CIM_AssociatedPowerManagementService.PowerState property.

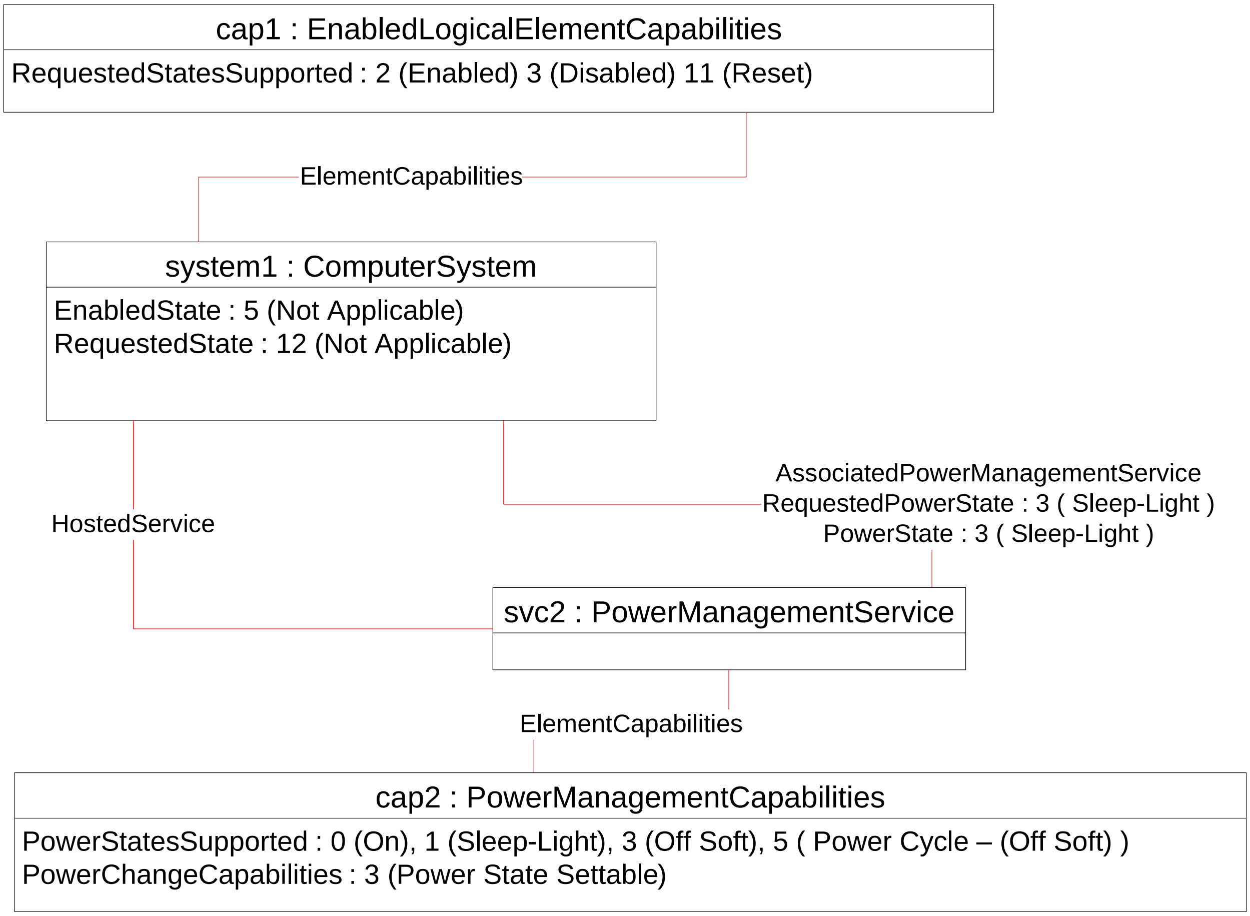

cap1 : EnabledLogicalElementCapabilities
RequestedStatesSupported : 2 (Enabled) 3 (Disabled) 11 (Reset)
ElementCapabilities
system1 : ComputerSystem
EnabledState : 5 (Not Applicable)
RequestedState : 12 (Not Applicable)
AssociatedPowerManagementService
RequestedPowerState : 3 (Sleep-Light)
PowerState : 3 (Sleep-Light)
HostedService
svc2 : PowerManagementService
ElementCapabilities
cap2 : PowerManagementCapabilities
PowerStatesSupported : 0 (On), 1 (Sleep-Light), 3 (Off Soft), 5 (Power Cycle – (Off Soft))
PowerChangeCapabilities : 3 (Power State Settable)

Figure 4 – Advanced Power Management

Figure 5 illustrates the network interfaces of the system. The *Ethernet Port Profile*, *IP Interface Profile*, *DHCP Client Profile*, and *DNS Client Profile* are implemented. The system has a single network interface.

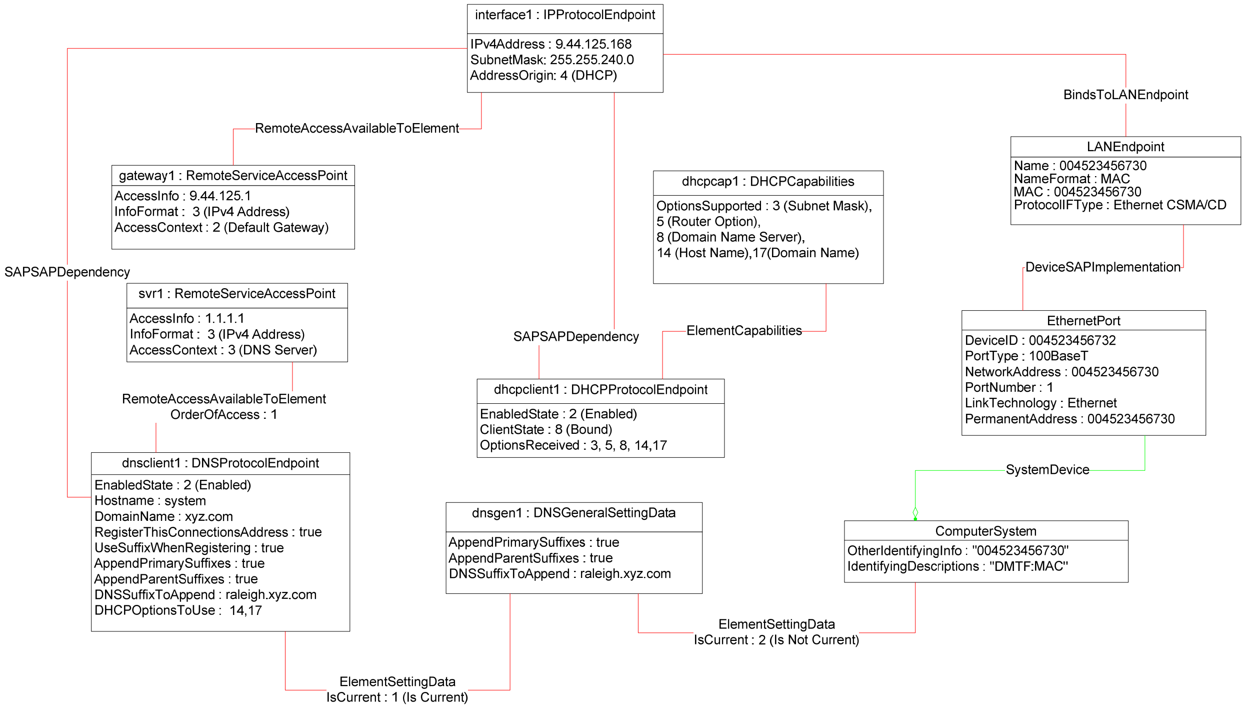

interface1 : IPProtocolEndpoint
IPv4Address : 9.44.125.168
SubnetMask: 255.255.240.0
AddressOrigin: 4 (DHCP)
BindsToLANEndpoint
RemoteAccessAvailableToElement
LANEndpoint
Name : 004523456730
NameFormat : MAC
MAC : 004523456730
ProtocolIFType : Ethernet CSMA/CD
gateway1 : RemoteServiceAccessPoint
AccessInfo : 9.44.125.1
InfoFormat : 3 (IPv4 Address)
AccessContext : 2 (Default Gateway)
dhcpcap1 : DHCPCapabilities
OptionsSupported : 3 (Subnet Mask),
5 (Router Option),
8 (Domain Name Server),
14 (Host Name),17(Domain Name)
DeviceSAPImplementation
SAPSAPDependency
svr1 : RemoteServiceAccessPoint
AccessInfo : 1.1.1.1
InfoFormat : 3 (IPv4 Address)
AccessContext : 3 (DNS Server)
EthernetPort
DeviceID : 004523456732
PortType : 100BaseT
NetworkAddress : 004523456730
PortNumber : 1
LinkTechnology : Ethernet
PermanentAddress : 004523456730
SAPSAPDependency
ElementCapabilities
RemoteAccessAvailableToElement
OrderOfAccess : 1
dhcpclient1 : DHCPProtocolEndpoint
EnabledState : 2 (Enabled)
ClientState : 8 (Bound)
OptionsReceived : 3, 5, 8, 14,17
SystemDevice
dnsclient1 : DNSProtocolEndpoint
EnabledState : 2 (Enabled)
Hostname : system
DomainName : xyz.com
RegisterThisConnectionsAddress : true
UseSuffixWhenRegistering : true
AppendPrimarySuffixes : true
AppendParentSuffixes : true
DNSSuffixToAppend : raleigh.xyz.com
DHCPOptionsToUse : 14,17
dnsgen1 : DNSGeneralSettingData
AppendPrimarySuffixes : true
AppendParentSuffixes : true
DNSSuffixToAppend : raleigh.xyz.com
ComputerSystem
OtherIdentifyingInfo : "004523456730"
IdentifyingDescriptions : "DMTF:MAC"
ElementSettingData
IsCurrent : 2 (Is Not Current)
ElementSettingData
IsCurrent : 1 (Is Current)

Figure 5 – Network Interfaces

Figure 6, Figure 7, and Figure 8 illustrate the logical and physical containment hierarchy of a single system.

Figure 6 illustrates the logical hierarchy of components contained in the system. The optional CPU Profile, Fan Profile, Power Supply Profile, Sensors Profile, System Memory Profile, and SMASH Collections Profile have been implemented. The system has four processors; each processor has a dedicated voltage sensor and a dedicated temperature sensor. The total system memory available is modeled. The system has two power supplies installed; each power supply has a dedicated voltage sensor. Four fans are installed in the system; each fan has a dedicated tachometer associated with it.

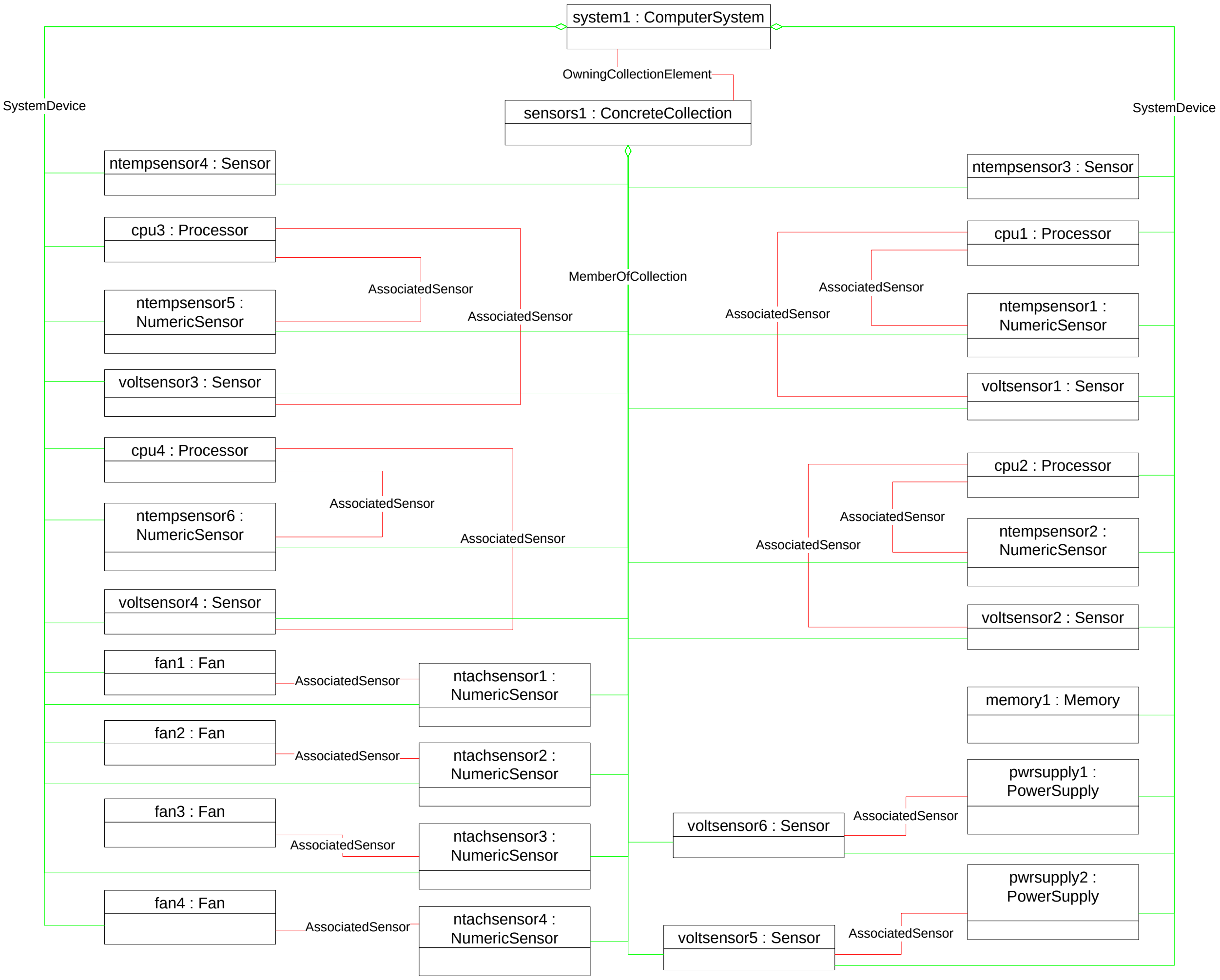


Figure 6 – Logical Topology

Figure 7 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 the fans 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.

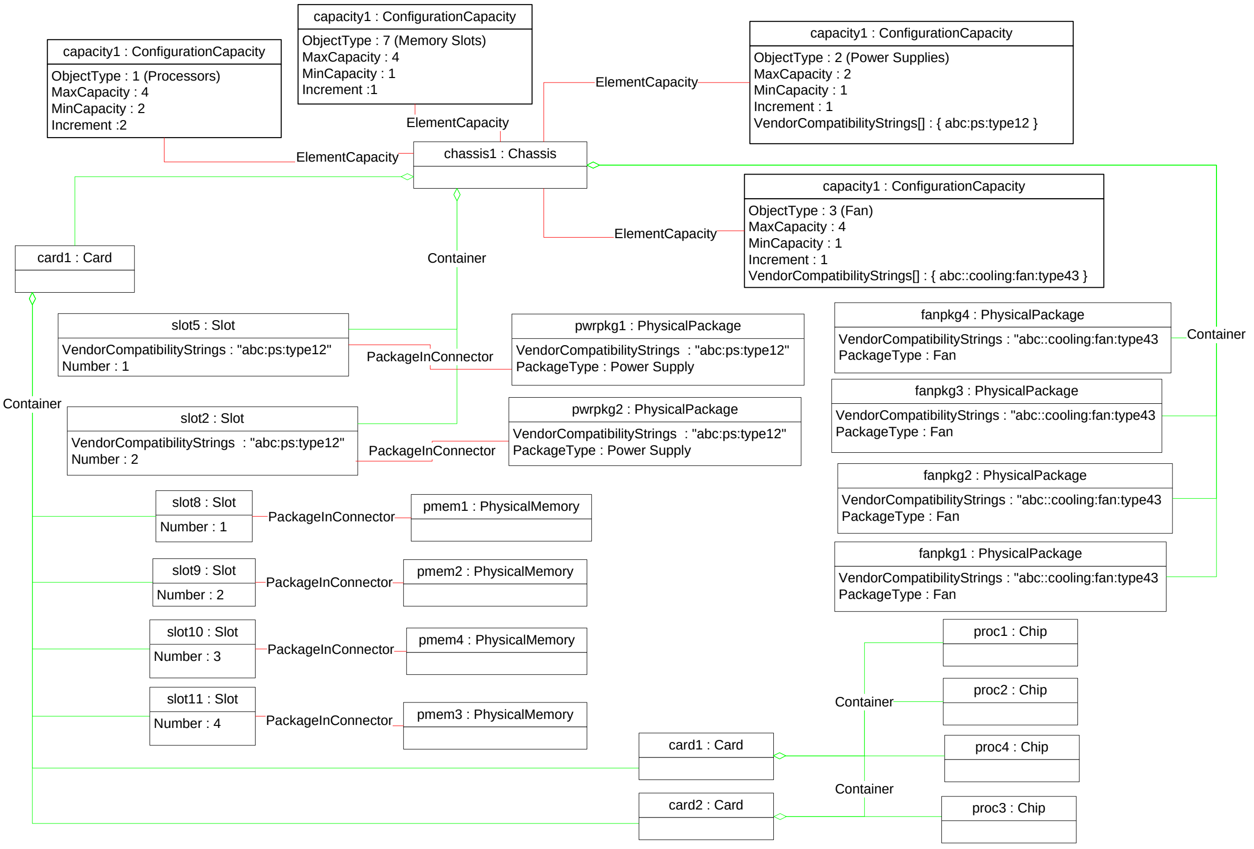

capacity1 : ConfigurationCapacity
ObjectType : 7 (Memory Slots)
MaxCapacity : 4
MinCapacity : 1
Increment :1
capacity1 : ConfigurationCapacity
ObjectType : 1 (Processors)
MaxCapacity : 4
MinCapacity : 2
Increment :2
capacity1 : ConfigurationCapacity
ObjectType : 2 (Power Supplies)
MaxCapacity : 2
MinCapacity : 1
Increment : 1
VendorCompatibilityStrings[] : { abc:ps:type12 }
ElementCapacity
ElementCapacity
ElementCapacity
chassis1 : Chassis
capacity1 : ConfigurationCapacity
ObjectType : 3 (Fan)
MaxCapacity : 4
MinCapacity : 1
Increment : 1
VendorCompatibilityStrings[] : { abc::cooling:fan:type43 }
ElementCapacity
card1 : Card
Container
slot5 : Slot
VendorCompatibilityStrings : "abc:ps:type12"
Number : 1
PackageInConnector
pwrpkg1 : PhysicalPackage
VendorCompatibilityStrings : "abc:ps:type12"
PackageType : Power Supply
fanpkg4 : PhysicalPackage
VendorCompatibilityStrings : "abc::cooling:fan:type43
PackageType : Fan
Container
Container
slot2 : Slot
VendorCompatibilityStrings : "abc:ps:type12"
Number : 2
PackageInConnector
pwrpkg2 : PhysicalPackage
VendorCompatibilityStrings : "abc:ps:type12"
PackageType : Power Supply
fanpkg3 : PhysicalPackage
VendorCompatibilityStrings : "abc::cooling:fan:type43
PackageType : Fan
fanpkg2 : PhysicalPackage
VendorCompatibilityStrings : "abc::cooling:fan:type43
PackageType : Fan
slot8 : Slot
Number : 1
PackageInConnector
pmem1 : PhysicalMemory
slot9 : Slot
Number : 2
PackageInConnector
pmem2 : PhysicalMemory
fanpkg1 : PhysicalPackage
VendorCompatibilityStrings : "abc::cooling:fan:type43
PackageType : Fan
slot10 : Slot
Number : 3
PackageInConnector
pmem4 : PhysicalMemory
proc1 : Chip
proc2 : Chip
Container
slot11 : Slot
Number : 4
PackageInConnector
pmem3 : PhysicalMemory
card1 : Card
proc4 : Chip
Container
card2 : Card
proc3 : Chip

Figure 7 – Physical Topology

Figure 8 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 keep the diagram uncluttered, the CIM_SystemDevice associations have been elided.

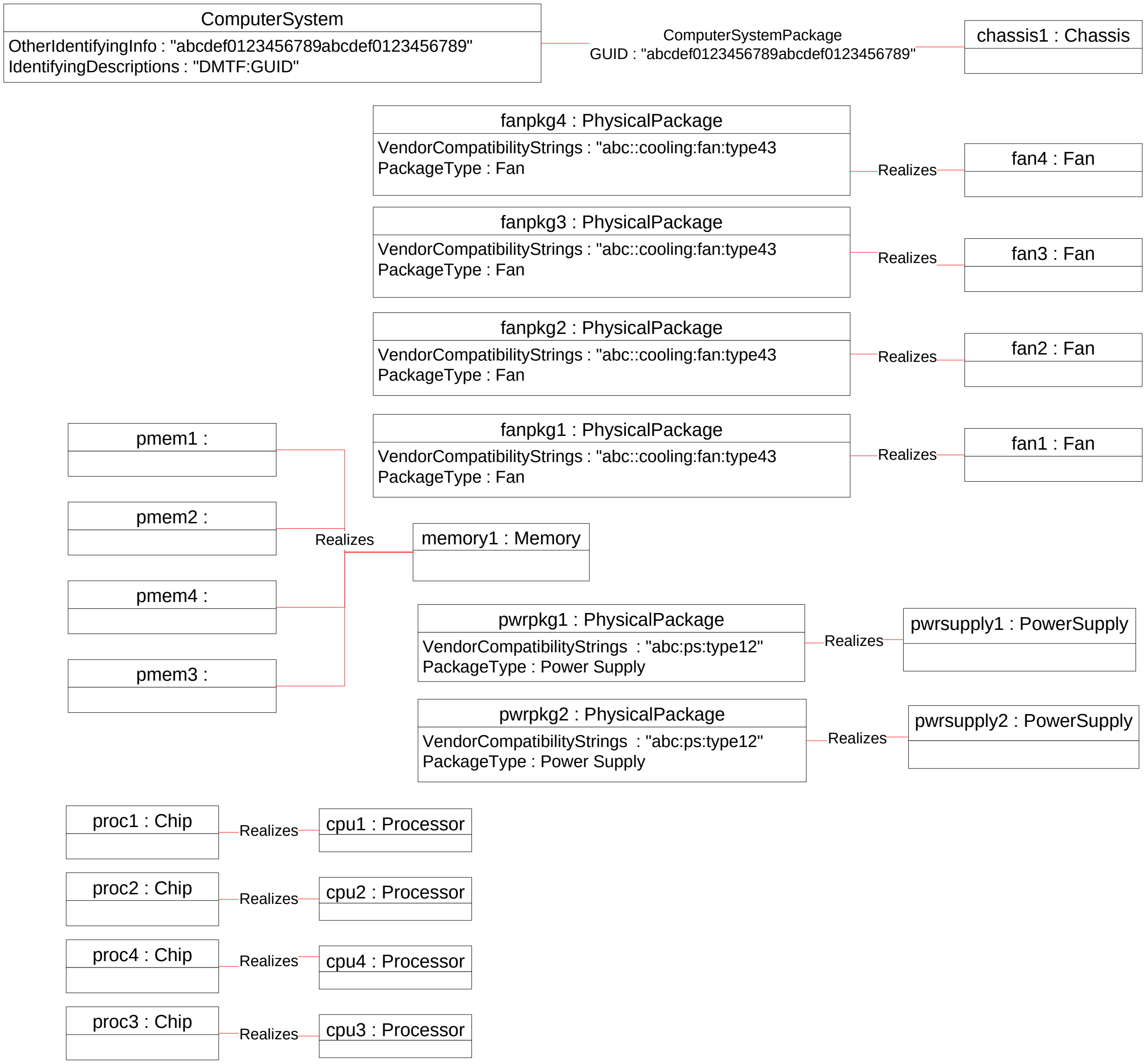

ComputerSystem
OtherIdentifyingInfo : "abcdef0123456789abcdef0123456789"
IdentifyingDescriptions : "DMTF:GUID"
ComputerSystemPackage
GUID : "abcdef0123456789abcdef0123456789"
chassis1 : Chassis
fanpkg4 : PhysicalPackage
VendorCompatibilityStrings : "abc::cooling:fan:type43
PackageType : Fan
Realizes
fan4 : Fan
fanpkg3 : PhysicalPackage
VendorCompatibilityStrings : "abc::cooling:fan:type43
PackageType : Fan
Realizes
fan3 : Fan
fanpkg2 : PhysicalPackage
VendorCompatibilityStrings : "abc::cooling:fan:type43
PackageType : Fan
Realizes
fan2 : Fan
pmem1 :
fanpkg1 : PhysicalPackage
VendorCompatibilityStrings : "abc::cooling:fan:type43
PackageType : Fan
Realizes
fan1 : Fan
pmem2 :
Realizes
memory1 : Memory
pmem4 :
pwrpkg1 : PhysicalPackage
VendorCompatibilityStrings : "abc:ps:type12"
PackageType : Power Supply
Realizes
pwrsupply1 : PowerSupply
pmem3 :
pwrpkg2 : PhysicalPackage
VendorCompatibilityStrings : "abc:ps:type12"
PackageType : Power Supply
Realizes
pwrsupply2 : PowerSupply
proc1 : Chip
Realizes
cpu1 : Processor
proc2 : Chip
Realizes
cpu2 : Processor
proc4 : Chip
Realizes
cpu4 : Processor
proc3 : Chip
Realizes
cpu3 : Processor

Figure 8 – Logical to Physical Mapping

9.2 Determine the System Model and Serial Number

When the optional asset management of the *Physical Asset Profile* has 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) Query 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 Central 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 Central 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 Shutdown 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 Central Instance through the CIM_ElementCapabilities association.

2) Verify that the CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported property contains the value 11 (Reset).

3) Invoke the RequestStateChange( ) parameter 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 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 if non-ACPI system). If the property has the value 3 (Disabled), the system is currently in ACPI state S0 (or equivalent if 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 also 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 the 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 and the total number of 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 7 shows the instances of CIM Elements for this profile. Instances of the CIM Elements shall be implemented as described in Table 7. Clauses 7 ("Implementation") and 8 ("Methods") may impose additional requirements on these elements.

Table 7 – CIM Elements: Base Server Profile

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

10.1 CIM_ComputerSystem

An instance of CIM_ComputerSystem is used to represent the system. Table 8 contains the requirements for elements of this class.

Table 8 – Class: CIM_ComputerSystem

Elements	Requirement	Notes
EnabledState	Mandatory	See 7.3.3.1 and 7.3.1.
RequestedState	Mandatory	See 7.3.3.2.
Dedicated	Mandatory	

10.2 CIM_ComputerSystemPackage

One or more instances of CIM_ComputerSystemPackage associate the CIM_ComputerSystem instance with the CIM_PhysicalPackage instances in which it resides. The constraints specified in Table 9 are in addition to those specified in the *Physical Asset Profile*.

Table 9 – Class: CIM_ComputerSystemPackage

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

10.3 CIM_EnabledLogicalElementCapabilities

CIM_EnabledLogicalElementCapabilities indicates support for managing the state of the system. Table 10 contains the requirements for elements of this class.

Table 10 – Class: CIM_EnabledLogicalElementCapabilities

Elements	Requirement	Notes
RequestedStatesSupported	Mandatory	See 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 instance of CIM_PhysicalPackage, 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 Server Profile* in order for a client to determine whether an instance of CIM_ComputerSystem is conformant with this profile. The CIM_RegisteredProfile class is defined by the *Profile Registration Profile*. With the exception of the mandatory values specified for the properties in Table 11, the behavior of the CIM_RegisteredProfile instance is in accordance with the *Profile Registration Profile*.

Table 11 – Class: CIM_RegisteredProfile

Elements	Requirement	Notes
RegisteredName	Mandatory	This property shall have a value of "Base Server".
RegisteredVersion	Mandatory	This property shall have a value of "1.0.1".
RegisteredOrganization	Mandatory	This property shall have a value of 2 (DMTF).

NOTE: Previous versions of this document included the suffix “Profile” for the RegisteredName value. If implementations querying for the RegisteredName value find the suffix “Profile”, they should ignore the suffix, with any surrounding white spaces, before any comparison is done with the value as specified in this document.

ANNEX A
(informative)

Change Log

Version	Date	Description
1.0.0	2009-06-16	DMTF Standard Release
1.0.1	2010-04-22	DMTF Standard Release – Corrected Figure 5