Document Number: DSP1026

Date: 2008-01-29

Version: 1.0.0

System Memory Profile

Document Type: Specification

Document Status: Final Standard

Document Language: E

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patent owner or claimant, and shall have no liability or responsibility for costs or losses incurred if a standard is withdrawn or modified after publication, and shall be indemnified and held harmless by any party implementing the standard from any and all claims of infringement by a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

CONTENTS

Figures

Tables

Foreword

The *System Memory Profile* (DSP1026) was prepared by the Server Management 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

This document defines the classes used to describe the system memory. Also included are descriptions of association classes that describe the relationship of the system memory with the memory's physical aspects (such as FRU data), with the managed system, and with 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 shall be instantiated and manipulated to represent the system memory of managed systems 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 component described in this document.

System Memory Profile

1 Scope

The *System Memory Profile* extends the management capabilities of referencing profiles by adding the capability to represent the total memory available to the system. The memory's relationship with the memory's physical aspects, the managed system that uses the memory, and the profile's registration for the schema implementation version information are also described.

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 DSP0200, *CIM Operations over HTTP 1.2.0*

DMTF DSP0004, *CIM Infrastructure Specification 2.3.0*

DMTF DSP1000, *Management Profile Specification Template*

DMTF DSP1001, *Management Profile Specification Usage Guide*

DMTF DSP1011, *Physical Asset Profile*

DMTF DSP1033, *Profile Registration Profile*

2.2 Other References

ISO/IEC Directives, Part 2, *Rules for the structure and drafting of International Standards*,

Common Information Model (CIM) Specification v2.2, June 14, 1999

Unified Modeling Language (UML) from the Open Management Group (OMG)

DMTF DSP0215, *Server Management Managed Element Addressing Specification* (SM ME Addressing)

DMTF DSP0134, *System Management BIOS Reference Specification*, v2.5 Final

3 Terms and Definitions

For the purposes of this document, the terms and definitions in DSP1033 and DSP1001 and 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 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

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: *System Memory*

Version: 1.0.0

Organization: DMTF

CIM Schema version: 2.10.0

Central Class: CIM_Memory

Scoping Class: CIM_ComputerSystem

The *System Memory Profile* extends the management capability of the referencing profiles by adding the capability to describe the total memory available to a managed system. The *System Memory Profile* is a component profile.

Table 1 identifies profiles that are related to this profile.

Table 1 – Referenced Profiles

Profile Name	Organization	Version	Relationship	Behavior
Physical Asset	DMTF	1.0.0	Optional	See section 7.4.
Profile Registration	DMTF	1.0.0	Mandatory	None

6 Description

The *System Memory Profile* extends the management capability of the referencing profiles by adding the capability to describe memory devices associated with a system. The CIM_Memory class describes the system's total memory that includes both available and physical memory, and corresponds to collective representation of the SMBIOS Type 16, Type 17, Type 19, and Type 20 memory structures (see section **Error! Reference source not found.**). The CIM_PhysicalMemory class describes the physical aspects of the system's memory and every instance corresponds to the individual SMBIOS Type 17 memory structure.

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

The CIM_Memory class describes the total system memory. The physical aspects of the memory are described with the CIM_PhysicalMemory class, which is associated with the CIM_Memory class through the CIM_Realizes association. The ownership of the memory is represented through the CIM_SystemDevice association to the managed system, which is represented by the CIM_ComputerSystem class. The DMTF version of the *System Memory Profile* that is implemented is represented through the CIM_RegisteredProfile class.

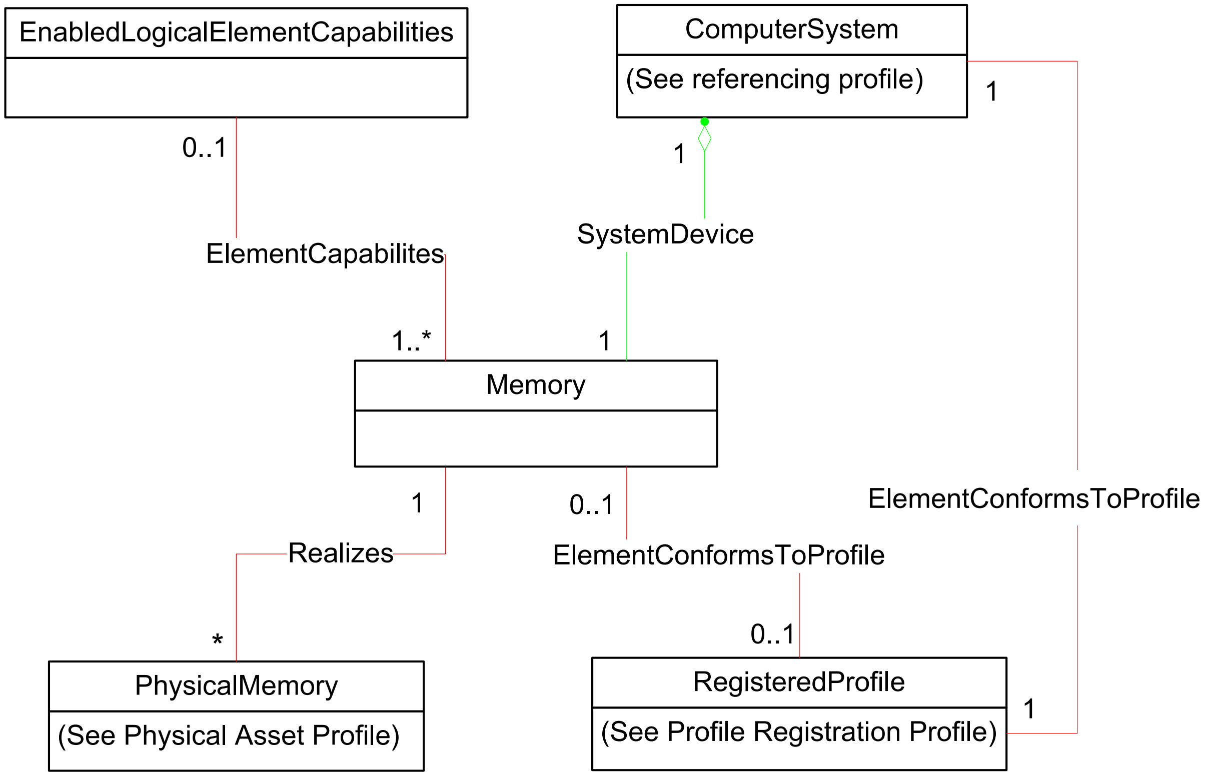

Figure 1 – System Memory Profile: Profile Class Diagram

7 Implementation

This section details the requirements related to the arrangement of instances and their properties for implementations of this profile. Methods are listed in section 8 (“Methods”), and properties are listed in section 10 (“CIM Elements”).

7.1 CIM_Memory

A single instance of CIM_Memory shall exist in accordance with the profile instantiation and shall be associated with the CIM_ComputerSystem instance that represents the managed system through an instance of CIM_SystemDevice. This CIM_Memory instance is the logical representation of multiple physical memories in a managed system and represents the total memory installed and available to the system..

7.2 Representation of the System Memory Size

This section describes the method of total system memory size calculation. Additionally this section describes the corresponding SMBIOS structures defined in DSP0134 that may be used for the calculation. Note that the underlying represented system does not need to support DSP0134.

7.2.1 Total Available System Memory

The ConsumableBlocks property represents the number of total available system memory blocks. When the number of available system memory blocks is unknown, the value of ConsumableBlocks shall be set to Null.

The total available memory of the system shall be the product of multiplying the value of the CIM_Memory.ConsumableBlocks property by the value of the CIM_Memory.BlockSize property.

When the DSP0134 is implemented, the product of multiplying the value of the CIM_Memory.ConsumableBlocks property by the value of the CIM_Memory.BlockSize property shall equal the sum of the values of the products of 1 KByte with the subtraction of value of StartingAddress property from the value of EndingAddress property (Σ [1KB x (EndingAddress – StartingAddress]) of either Memory Array Mapped Device (Type 19) structures or Memory Device Mapped Address (Type 20) structures of SMBIOS specification DSP0134.

7.2.2 Total Physical System Memory

The NumberOfBlocks property represents the number of total physical system memory blocks.

The total physical memory of the system shall be the product of multiplying the value of the CIM_Memory.NumberOfBlocks property by the value of the CIM_Memory.BlockSize property.

When the DSP0134 is implemented, the product of multiplying the value of the CIM_Memory.NumberOfBlocks property by the value of the CIM_Memory.BlockSize property shall equal the sum of the values of the Size property of Memory Device (Type 17) structure of SMBIOS specification DSP0134.

When the optional behavior of modeling the physical aspects of the system memory specified in section 7.4 is implemented, the product of multiplying the value of the CIM_Memory.NumberOfBlocks property by the value of the CIM_Memory.BlockSize property shall equal the sum of the values of the CIM_PhysicalMemory.Capacity property for each instance of the CIM_PhysicalMemory to which the CIM_Memory instance is associated through the CIM_Realizes association.

7.3 CIM_EnabledLogicalElementCapabilities

When the CIM_EnabledLogicalElementCapabilities class is instantiated, the instance of CIM_EnabledLogicalElementCapabilities shall be associated with the CIM_Memory instance through an instance of CIM_ElementCapabilities and used for advertising the capabilities of the CIM_Memory instance.

At most one instance of CIM_EnabledLogicalElementCapabilities shall be associated with a given instance of CIM_Memory.

7.3.1 CIM_EnabledLogicalElementCapabilities.ElementNameEditSupported

The ElementNameEditSupported property shall have a value of TRUE when the implementation supports client modification of the CIM_Memory.ElementName property.

7.3.2 CIM_EnabledLogicalElementCapabilities.MaxElementNameLen

The MaxElementNameLen property shall be implemented when the ElementNameEditSupported property has a value of TRUE.

7.4 Physical Asset Profile

The *Physical Asset Profile* may be implemented to model the physical aspects of the system memory, including the asset information.

When the system memory’s physical aspects are represented, each CIM_PhysicalMemory instance(s) shall be instantiated and associated with the instance of CIM_Memory through instance(s) of CIM_Realizes.

8 Methods

This section details the requirements for supporting intrinsic operations for the CIM elements defined by this profile. No extrinsic methods are defined for this profile.

8.1 Profile Conventions for Operations

Support for operations for each profile class (including associations) is specified in the following subclauses. Each subclause includes either the statement “All operations in the default list in section 8.1 are supported as described by DSP0200 version 1.2” or a table listing all of the operations that are not supported by this profile or where the profile requires behavior other than that described by DSP0200 version 1.2.

The default list of operations is as follows:

- GetInstance
- EnumerateInstances
- EnumerateInstanceNames
- Associators
- AssociatorNames
- References
- ReferenceNames

A compliant implementation shall support all of the operations in the default list for each class, unless the “Requirement” column states something other than *Mandatory*.

8.2 CIM_ElementCapabilities

Table 2 lists operations that either have special requirements beyond those from DSP0200 version 1.2 or shall not be supported.

Table 2 – Operations: CIM_ElementCapabilities

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

8.3 CIM_EnabledLogicalElementCapabilities

All operations in the default list in section 8.1 are supported as described by DSP0200 version 1.2.

8.4 CIM_Memory

Table 3 lists operations that either have special requirements beyond those from DSP0200 version 1.2 or shall not be supported.

Table 3 – Operations: CIM_Memory

Operation	Requirement	Messages
ModifyInstance	Conditional. See section 8.4.1.	None

8.4.1 CIM_Memory—ModifyInstance

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

The ModifyInstance operation shall be supported and CIM_Memory.ElementName shall be modifiable when an instance of CIM_EnabledLogicalElementCapabilities is associated to the instance of CIM_Memory, and the ElementNameEditSupported property of the CIM_EnabledLogicalElementCapabilities instance has a value of TRUE. See section 8.4.1.1.

8.4.1.1 CIM_Memory.ElementName

When an instance of CIM_EnabledLogicalElementCapabilities is associated to the instance of CIM_Memory, and the ElementNameEditSupported property of the CIM_EnabledLogicalElementCapabilities instance has a value of TRUE, the implementation shall allow the ModifyInstance operation to change the value of the ElementName property of the CIM_Memory 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 or when there is no instance of CIM_EnabledLogicalElementCapabilities associated with the instance of CIM_Memory, the implementation shall not allow the ModifyInstance operation to change the value of the ElementName property of the CIM_Memory instance.

8.5 CIM_SystemDevice

Table 4 lists operations that either have special requirements beyond those from DSP0200 version 1.2 or shall not be supported.

Table 4 – Operations: CIM_SystemDevice

Operation	Requirement	Messages
EnumerateInstances	Unspecified	None
EnumerateInstanceNames	Unspecified	None
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 *System Memory Profile*.

9.1 Object Diagrams

Figure 2 represents a possible instantiation of the *System Memory Profile*. In this case, the computer system represented by the system1 instance of CIM_ComputerSystem owns and uses memory represented by the memory1 instance of CIM_Memory. memory1 contains information such as the volatility, accessibility, total physical and available size, and operational and health status of the memory. The property ConsumableBlocks represents the memory blocks available for consumption by the system and is equal to the total physical system memory blocks represented by the NumberOfBlocks property. The physical aspects such as memory type, physical size, form factor, and other FRU data are described by the synchdram1 instance of CIM_PhysicalMemory. profile1 shows the version of the current *System Memory Profile* implementation.

For simplicity, the prefix CIM_ has been removed from the names of the classes.

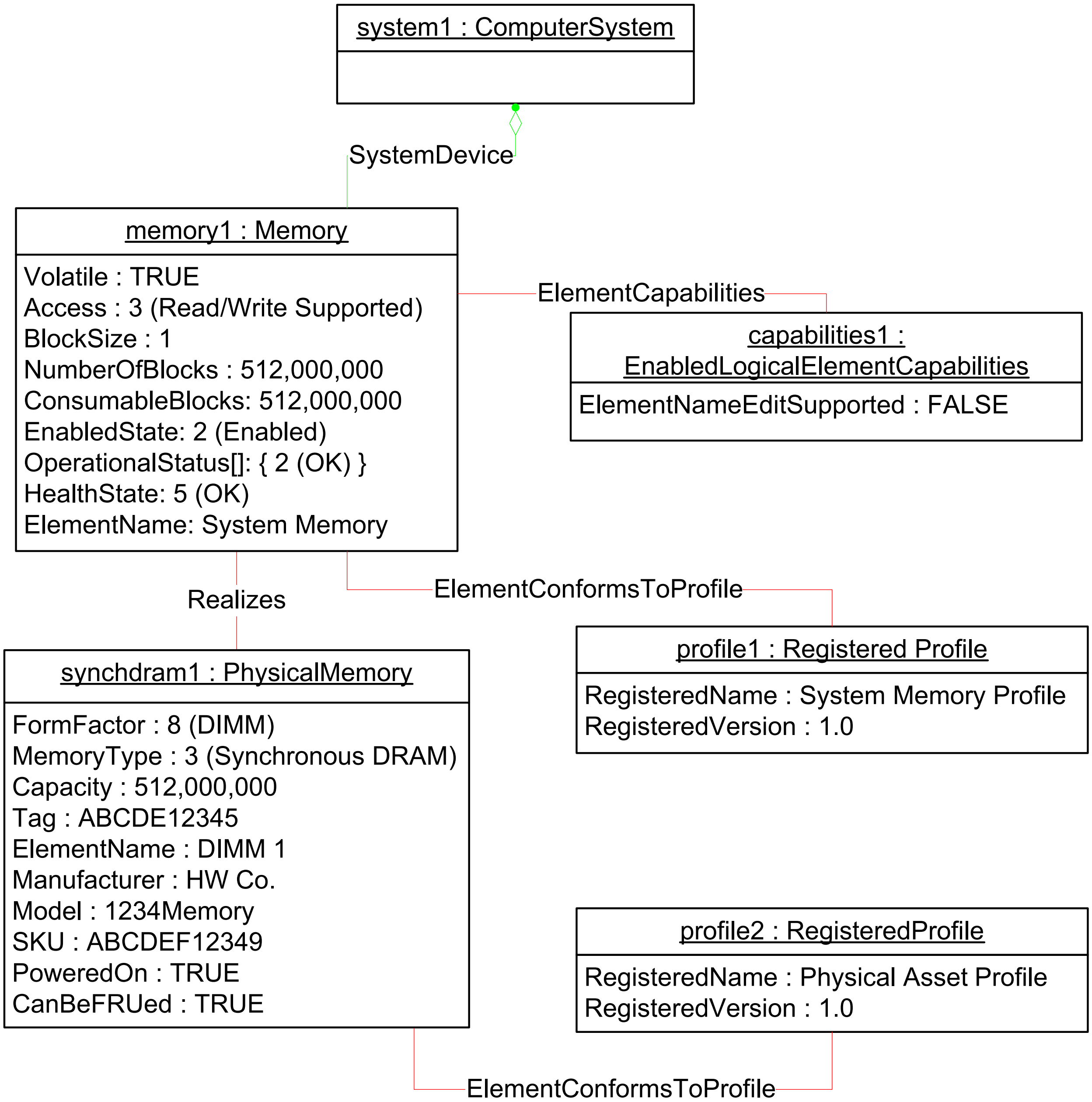
system1 : ComputerSystem
SystemDevice
memory1 : Memory
Volatile : TRUE
Access : 3 (Read/Write Supported)
BlockSize : 1
NumberOfBlocks : 512,000,000
ConsumableBlocks: 512,000,000
EnabledState: 2 (Enabled)
OperationalStatus[]: { 2 (OK) }
HealthState: 5 (OK)
ElementName: System Memory
ElementCapabilities
capabilities1 : EnabledLogicalElementCapabilities
ElementNameEditSupported : FALSE
Realizes
ElementConformsToProfile
synchdram1 : PhysicalMemory
FormFactor : 8 (DIMM)
MemoryType : 3 (Synchronous DRAM)
Capacity : 512,000,000
Tag : ABCDE12345
ElementName : DIMM 1
Manufacturer : HW Co.
Model : 1234Memory
SKU : ABCDEF12349
PoweredOn : TRUE
CanBeFRUed : TRUE
profile1 : Registered Profile
RegisteredName : System Memory Profile
RegisteredVersion : 1.0
profile2 : RegisteredProfile
RegisteredName : Physical Asset Profile
RegisteredVersion : 1.0
ElementConformsToProfile

Figure 2 – System Memory Profile: Object Diagram 1

Figure 3 also represents a possible instantiation of the *System Memory Profile*. In this case, the computer system represented by the system1 instance of CIM_ComputerSystem owns and uses memory represented by the memory1 instance of CIM_Memory. memory1, as in Figure 2, contains logical information about the system memory, but in this case, memory1 is a logical representation of two physical memories, synchdram1 and synchdram2. Thus, the size of system memory, represented by the properties BlockSize and NumberOfBlocks of memory1, is the sum of the physical sizes, represented by the Capacity property of synchdram1 and synchdram2. But because system1 has system memory redundancy, the available system memory represented by the properties of BlockSize and ConsumableBlocks is half of the total physical system memory. profile1 shows the version of the current *System Memory Profile* implementation.

For simplicity, the prefix CIM_ has been removed from the names of the classes.

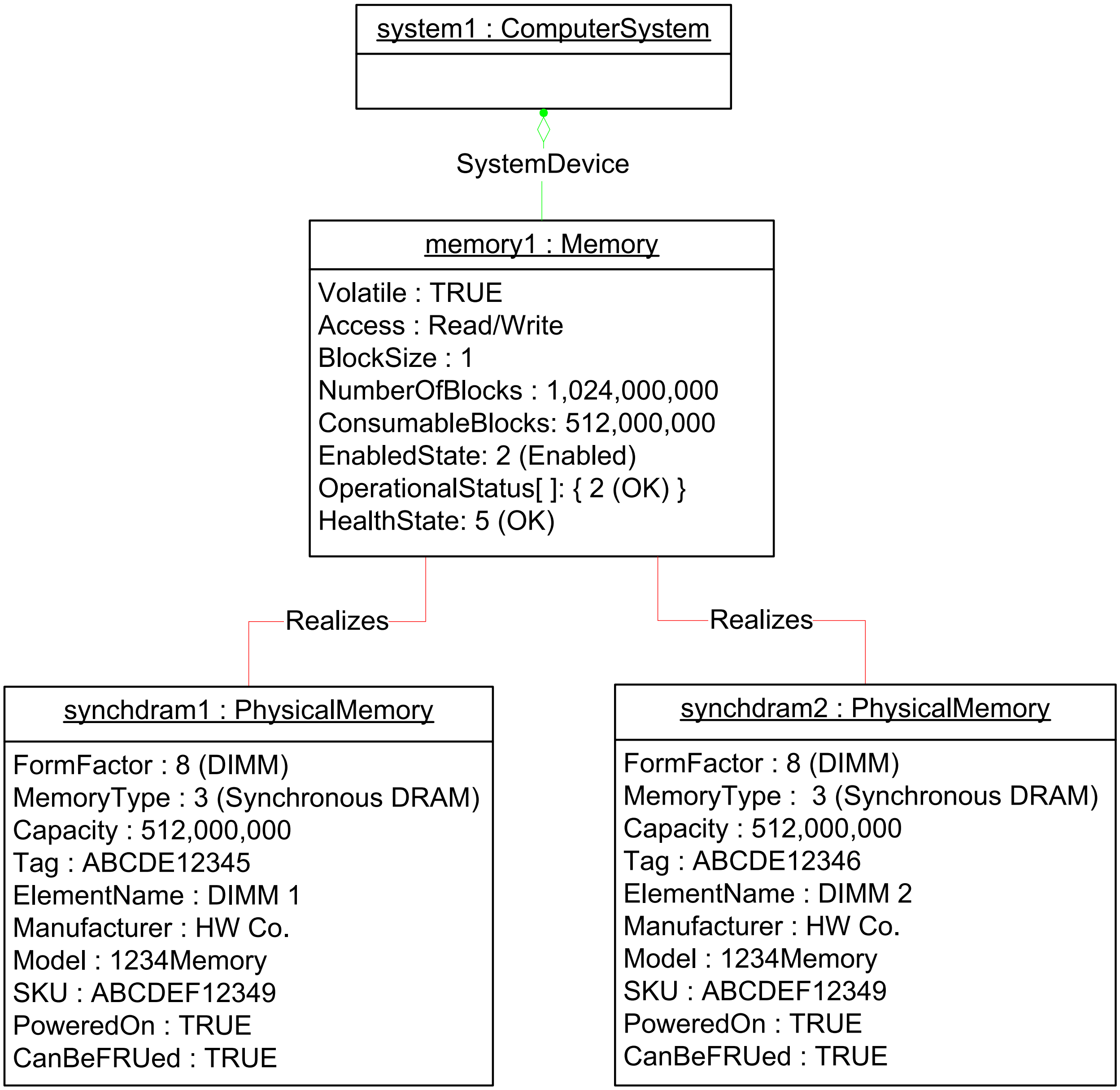

Figure 3 – System Memory Profile: Object Diagram 2

9.2 Find the System's Memory Information

A client can find the system's memory information as follows:

1) Select the instance of CIM_Memory that is associated with the given instance of CIM_ComputerSystem through the CIM_SystemDevice association.
2) Select the values of the properties of the CIM_Memory instance.

9.3 Find the System's Physical Memory Information

A client can find the system's physical memory information as follows:

1) Select all the instances of CIM_Realizes that reference the instance of CIM_Memory.

2) Select the CIM_PhysicalMemory instances that are referenced by the instances of CIM_Realizes.

3) Select the values of the properties of the selected CIM_PhysicalMemory instances.

9.4 Find the Total Physical System Memory Size

A client can determine the total system memory size as follows:

1) Select the instance of CIM_Memory that is associated with the given instance of CIM_ComputerSystem through the CIM_SystemDevice association.

2) For the instance of CIM_Memory, select the BlockSize and NumberOfBlocks properties and multiply their values together to show the total system memory in bytes.

9.5 Find the Total Available System Memory Size

A client can determine the total available system memory size as follows:

1) Select the instance of CIM_Memory that is associated with the given instance of CIM_ComputerSystem through the CIM_SystemDevice association.

2) For the instances of CIM_Memory, select the BlockSize and ConsumableBlocks properties and multiply their values together to show the total available system memory in bytes.

9.6 Find the Physical System Memory Size per Memory Device

A client can determine the total physical system memory size as follows:

1) Select all of the instances of CIM_Realizes that reference the instance of the CIM_Memory.

2) Select the CIM_PhysicalMemory instances that are referenced by the CIM_Realizes instances.

3) The Capacity property of the selected instances of CIM_PhysicalMemory to show thel size of the physical system memory in bytes per memory device.

9.7 Determine Whether ElementName for the Instance of CIM_Memory Is Modifiable

A client can determine whether it can modify the ElementName property of the CIM_Memory instance as follows:

1) Select the instance of CIM_EnabledLogicalElementCapabilities that is associated with the instance of CIM_Memory through the CIM_ElementCapabilities association.

2) Determine if the ElementNameEditSupported property has value of TRUE.

If the value is TRUE, the client can invoke the ModifyInstance operation to modify the value of the CIM_Memory.ElementName property. If the value is FALSE or the instance of CIM_EnabledLogicalElementCapabilities does not exist, the client cannot modify the value of the CIM_Memory.ElementName property.

10 CIM Elements

Table 5 shows the instances of CIM Elements for this profile. Instances of the CIM Elements shall be implemented as described in Table 5. Sections 7 (“Implementation”) and 8 (“Methods”) may impose additional requirements on these elements.

Table 5 – CIM Elements: System Memory Profile

Element Name	Requirement	Description
Classes		
CIM_ElementCapabilities	Conditional	See section 10.1.
CIM_EnabledLogicalElementCapabilities	Optional	See sections 7.3 and 10.2.
CIM_Memory	Mandatory	See sections 7.1 and 10.3.
CIM_RegisteredProfile	Mandatory	See section 10.4.
CIM_SystemDevice	Mandatory	See section 0.
Indications		
None defined in this profile		

10.1 CIM_ElementCapabilities

CIM_ElementCapabilities associates the CIM_Memory instance with the CIM_EnabledLogicalElementCapabilities instance that describes the capabilities of CIM_Memory. CIM_ElementCapabilities is mandatory when the CIM_EnabledLogicalElementCapabilities instance is instantiated. Table 6 contains the requirements for elements of this class.

Table 6 – Class: CIM_ElementCapabilities

Elements	Requirement	Notes
ManagedElement	Mandatory	**Key**: This property shall reference the instance of CIM_Memory. Cardinality 1..*, indicating one or many references
Capabilities	Mandatory	**Key**: This property shall reference the instance of CIM_EnabledLogicalElementCapabilities. Cardinality 0..1, indicating zero or one reference

10.2 CIM_EnabledLogicalElementCapabilities

CIM_EnabledLogicalElementCapabilities represents the capabilities of the system memory. Table 7 contains the requirements for elements of this class.

Table 7 – Class: CIM_EnabledLogicalElementCapabilities

Elements	Requirement	Notes
InstanceID	Mandatory	**Key**
ElementNameEditSupported	Mandatory	See section 7.3.1.
MaxElementNameLen	Conditional	See section 7.3.2.

10.3 CIM_Memory

CIM_Memory represents the logical properties of memory in a managed system. For more implementation details, see section 7.1. Table 8 contains the requirements for elements of this class.

Table 8 – Class: CIM_Memory

Elements	Requirement	Notes
SystemCreationClassName	Mandatory	**Key**
SystemName	Mandatory	**Key**
CreationClassName	Mandatory	**Key**
DeviceID	Mandatory	**Key**
Volatile	Mandatory	None
Access	Mandatory	None
BlockSize	Mandatory	None
NumberOfBlocks	Mandatory	See section **Error! Reference source not found.**.
ConsumableBlocks	Mandatory	See section **Error! Reference source not found.**.
EnabledState	Mandatory	This property shall match 2 (Enabled).
RequestedState	Mandatory	This property shall match 12 (Not Applicable).
OperationalStatus	Mandatory	None
HealthState	Mandatory	None
ElementName	Mandatory	This property shall be formatted as a free-form string of variable length (pattern “.*”).

10.4 CIM_RegisteredProfile

CIM_RegisteredProfile is defined by the *Profile Registration Profile*. The requirements denoted in Table 9 are in addition to those mandated by the *Profile Registration Profile*.

Table 9 – Class: CIM_RegisteredProfile

Elements	Requirement	Notes
RegisteredName	Mandatory	This property shall have a value of “System Memory”.
RegisteredVersion	Mandatory	This property shall have a value of “1.0.0”.
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 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.

10.5 CIM_SystemDevice

CIM_SystemDevice associates the CIM_Memory instance with the CIM_ComputerSystem instance of which CIM_Memory is a member. Table 10 contains the requirements for elements of this class.

Table 10 – Class: CIM_SystemDevice

Elements	Requirement	Notes
GroupComponent	Mandatory	**Key:** This property shall reference the CIM_ComputerSystem instance of which the CIM_Memory instance is a member. Cardinality 1, indicating one reference
PartComponent	Mandatory	**Key:** This property shall reference the CIM_Memory instance. Cardinality 1, indicating one reference

ANNEX A
(informative)

Change Log

Version	Date	Description
1.0.0f	08/08/2006	Preliminary Standard
1.0.0	10/12/2007	Final Standard

ANNEX B
(informative)

Acknowledgments

The authors wish to acknowledge the following people.

Editor:

- Khachatur Papanyan – Dell Inc.

Contributors:

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