Document Number: DSP1052

Date: 2008-12-08

Version: 1.0.0

Computer System Profile

Document Type: Specification

Document Status: Final Standard

Document Language: E

CONTENTS

Figures

Tables

Foreword

The *Computer System Profile* (DSP1052) was prepared by the Server Management 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 basic computer system 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.

Computer System Profile

1 Scope

The *Computer System Profile* is the autonomous profile that defines the minimum top-level object model needed to define a basic computing platform. This profile is intended to be the base profile for specialization for the modeling of specific types of computer systems such as virtual machines, servers, desktops, and mobile computers. The *Computer System Profile* identifies component profiles for integration of additional management functionality including system configuration, boot control, and other provisioning capabilities.

2 Normative References

The following referenced documents are indispensable for the application of this document.

2.1 Approved References

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 DSP1022, *CPU Profile 1.0.0*

DMTF DSP1037, *DHCP Client Profile 1.0.0*

DMTF DSP1038, *DNS Client Profile 1.0.0*

DMTF DSP1014, *Ethernet Port Profile 1.0.0*

DMTF DSP1036, *IP Interface Profile 1.0.0*

DMTF DSP1000, *Management Profile Specification Template*

DMTF DSP1001, *Management Profile Specification Usage Guide*

DMTF DSP1033, *Profile Registration Profile 1.0.0*

DMTF DSP1010, *Record Log Profile 1.0.0*

DMTF DSP1009, *Sensors Profile 1.0.0*

DMTF DSP1023, *Software Inventory Profile 1.0.0*

DMTF DSP1026, *System Memory Profile 1.0.0*

DMTF DSP1024, *Text Console Redirection Profile 1.0.0*

2.2 References under Development

DMTF DSP1005, *CLP Service Profile 1.0.0*

DMTF DSP1006, *SMASH Collections Profile 1.0.0*

DMTF DSP1016, *Telnet Service Profile 1.0.0*

DMTF DSP1017, *SSH Service Profile 1.0.0*

DMTF DSP1025, *Software Update Profile 1.0.0*

2.3 Other References

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

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

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

The following abbreviations are used in this document.

Experimental Maturity Level

Some of the content considered for inclusion in *Computer System Profile* has yet to receive sufficient review to satisfy the adoption requirements set forth by the Technical Committee within the DMTF. This content is presented here as an aid to implementers who are interested in likely future developments within this specification. The content marked experimental may change as implementation experience is gained. There is a high likelihood that it will be included in an upcoming revision of the specification. Until that time, it is purely informational, and is clearly marked within the text.
A sample of the typographical convention for experimental content is included here:

EXPERIMENTAL

Experimental content appears here

EXPERIMENTAL

4.1

IP

Internet Protocol

4.2

SSH

Secure Shell

5 Synopsis

Profile Name: *Computer System Profile*

Version: 1.0.0

Organization: DMTF

CIM schema version: 2.20.0

Central Class: CIM_ComputerSystem

Scoping Class: CIM_ComputerSystem

This abstract profile specification shall not be directly implemented; implementations shall be based on a profile specification that specializes the requirements of this profile.

The *Computer System Profile* is an autonomous profile that provides the capability to manage a general-purpose computer system. It is an appropriate target for management for clients that are interested in performing management tasks that are common across diverse computing platforms such as virtual machines, servers, and desktop platforms.

The Central Class of the *Computer System 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
Boot Control	DMTF	1.0	Optional	See section 7.7.2.
CLP Service	DMTF	1.0	Optional	See section 7.6.1. EXPERIMENTAL
CPU	DMTF	1.0	Optional	See section 7.2.1.
DHCP Client	DMTF	1.0	Optional	See section 7.4.3.
DNS Client	DMTF	1.0	Optional	See section 7.4.4.
Ethernet Port	DMTF	1.0	Optional	See section 7.4.1.
IP Interface	DMTF	1.0	Optional	See section 7.4.2.
Record Log	DMTF	1.0	Optional	See section 7.5.
Sensors	DMTF	1.0	Optional	See section 7.2.3.
SMASH Collections	DMTF	1.0	Optional	See section 7.8. EXPERIMENTAL
Software Inventory	DMTF	1.0	Optional	See section 7.3.1.
Software Update	DMTF	1.0	Optional	See section 7.3.2. EXPERIMENTAL
SSH Service	DMTF	1.0	Optional	See section 7.6.2. EXPERIMENTAL
System Memory	DMTF	1.0	Optional	See section 7.2.2.
Telnet Service	DMTF	1.0	Optional	See section 7.6.3. EXPERIMENTAL

6 Description

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

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

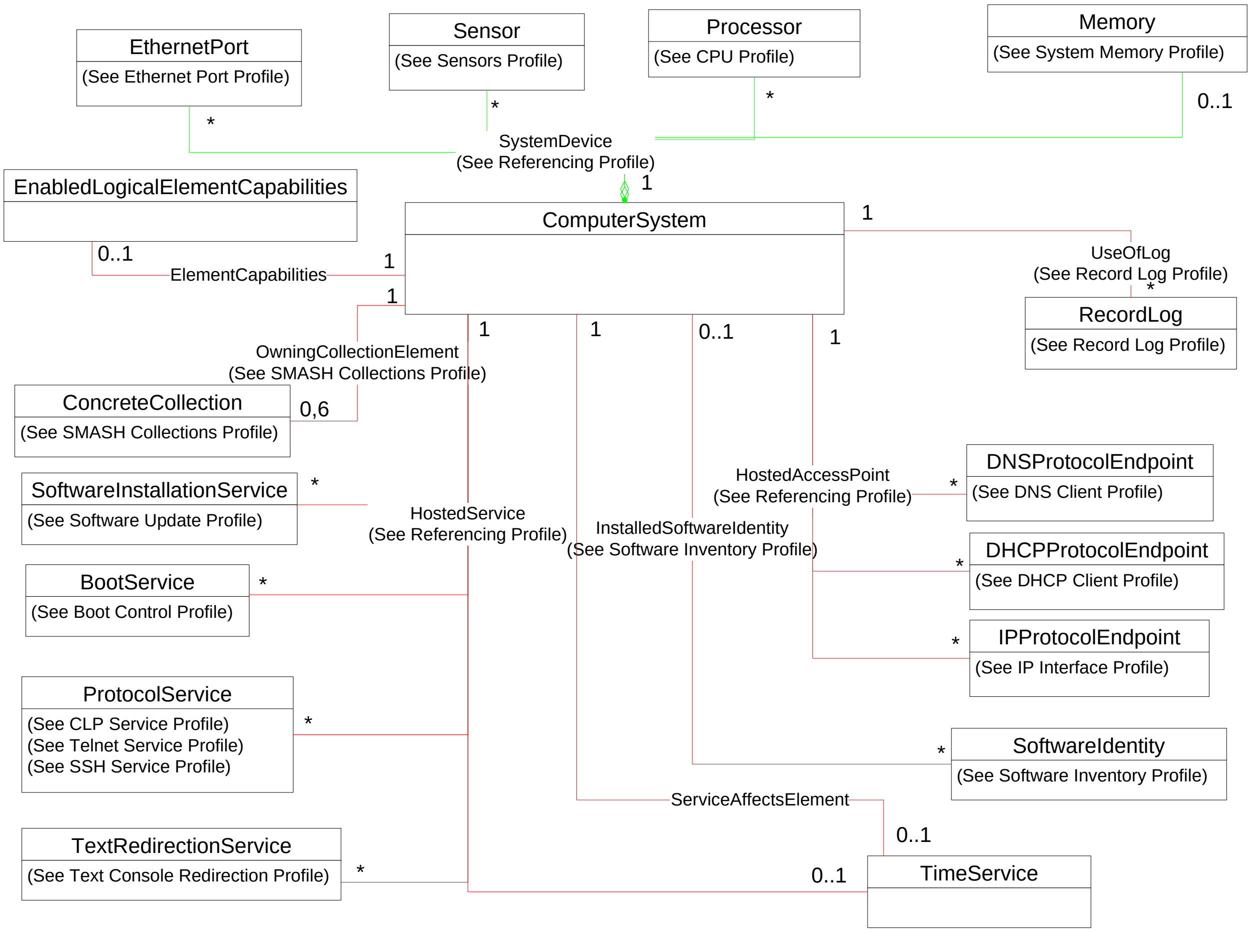

EthernetPort
(See Ethernet Port Profile)
Sensor
(See Sensors Profile)
Processor
(See CPU Profile)
Memory
(See System Memory Profile)
*
*
*
0..1
SystemDevice
(See Referencing Profile)
1
EnabledLogicalElementCapabilities
0..1
ElementCapabilities
1
ComputerSystem
1
UseOfLog
(See Record Log Profile)
*
RecordLog
(See Record Log Profile)
1
OwningCollectionElement
(See SMASH Collections Profile)
1
1
0..1
1
ConcreteCollection
(See SMASH Collections Profile)
0,6
SoftwareInstallationService
(See Software Update Profile)
*
HostedService
(See Referencing Profile)
InstalledSoftwareIdentity
(See Software Inventory Profile)
HostedAccessPoint
(See Referencing Profile)
*
DNSProtocolEndpoint
(See DNS Client Profile)
*
DHCPProtocolEndpoint
(See DHCP Client Profile)
BootService
(See Boot Control Profile)
*
*
IPProtocolEndpoint
(See IP Interface Profile)
ProtocolService
(See CLP Service Profile)
(See Telnet Service Profile)
(See SSH Service Profile)
*
*
SoftwareIdentity
(See Software Inventory Profile)
ServiceAffectsElement
0..1
TextRedirectionService
(See Text Console Redirection Profile)
*
0..1
TimeService

Figure 1 – Computer System Profile: Class Diagram

7 Implementation

The *Computer System Profile* consists of definitions for the classes CIM_ComputerSystem and CIM_TimeService, 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 *Computer System Profile* classes are discussed in this section.

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

7.1 Computer System

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

7.1.1 Identifying a Computer System

Name/Value pairs contained in the CIM_ComputerSystem.OtherIdentifyingInfo and CIM_ComputerSystem.IdentifyingDescriptions properties should contain values that clients can use to correlate instances of CIM_ComputerSystem that represent the same underlying real-world system that the specialization of the *Computer System Profile* has been instrumented to represent. The following paragraphs detail the requirements when the OtherIdentifyingInfo and IdentifyingDescriptions properties are implemented.

When the OtherIdentifyingInfo property is implemented, the IdentifyingDescriptions property shall be implemented. The IdentifyingDescriptions property shall be formatted using the following algorithm:

> < OrgID > : < LocalID >, where < OrgID > and < LocalID > are separated by a colon (:) and < OrgID > shall include a copyrighted, trademarked, or otherwise unique name that is owned by the business entity that is creating or defining the value or that is a registered ID assigned to the business entity by a recognized global authority. In addition, to ensure uniqueness, < OrgID > shall not contain a colon (:). When using this algorithm, the first colon to appear in the value shall appear between < OrgID > and < LocalID >. < LocalID > is chosen by the business entity and shall be used uniquely.

The values listed in the "IdentifyingDescriptions Value" column of Table 2 should be used as values for the IdentifyingDescriptions property. Every entry in Table 2 applicable for a given environment should be specified. An entry in Table 2 shall be used only if the value for the OtherIdentifyingInfo property is guaranteed to be globally unique across all underlying real-world systems.

Table 2 – Predefined Identifiers for a Computer System

IdentifyingDescriptions Value	OtherIdentifyingInfo Value
"CIM:GUID"	A globally unique identifier; see section 7.1.1.1
"CIM:MAC"	MAC address for one of the LAN interfaces of the system; see section 7.1.1.2
"CIM:Model:SerialNumber"	Model and serial number of the system; see section 7.1.1.3
"CIM:Tag"	Asset tag of the system; see section 7.1.1.4
“CIM:CorrelatableID”	An opaque identifier; see section 7.1.1.5

7.1.1.1 CIM:GUID

When the IdentifyingDescriptions property contains the value "CIM:GUID", the value of the corresponding array index of the OtherIdentifyingInfo property shall satisfy the following constraints:

- The value shall be a globally unique identifier for the system.
- The value shall match the pattern (“^[0..9A..F]{32}$”).

7.1.1.2 CIM:MAC

When the IdentifyingDescriptions property contains the value "CIM:MAC", the value of the corresponding array index of the OtherIdentifyingInfo property shall satisfy the following constraints:

- The value shall be the MAC address for one of the LAN interfaces of the system.

- The value shall be formatted as 12 contiguous uppercase hex digits (pattern "^[0123456789ABCDEF]{12}$").
- When the is implemented, the value shall match the value of the PermanentAddress property of an instance of CIM_EthernetPort.

7.1.1.3 CIM:Model:SerialNumber

When the IdentifyingDescriptions property contains the value "CIM:Model:SerialNumber", the value of the corresponding array index of the OtherIdentifyingInfo property shall be of the form < OrgID > : < LocalID > : <Model Number> : <Serial Number>, where < OrgID > and < LocalID > are separated by a colon (:), and where < OrgID > shall include a copyrighted, trademarked, or otherwise unique name that is owned by the business entity that is creating or defining the value or that is a registered ID assigned to the business entity by a recognized global authority. In addition, to ensure uniqueness, < OrgID > shall not contain a colon (:). When using this algorithm, the first colon to appear in the value shall appear between < OrgID > and < LocalID >. <LocalID> is chosen by the business entity and shall be used uniquely. <Model Number> shall be the model number of the system and <Serial Number> shall be the serial number of the system.

7.1.1.4 CIM:Tag

An asset tag is a unique identifier assigned to a computer system. Generally this value is assigned by an administrator or a client application.

When the IdentifyingDescriptions property contains the value "CIM:Tag", the value of the corresponding array index of the OtherIdentifyingInfo property shall be a uniquely identifying tag of the system. An example is an asset tag.

7.1.1.5 CIM:CorrelatableID

When the IdentifyingDescriptions property contains the value “CIM:CorrelatableID”, the value of the corresponding array index of the OtherIdentifyingInfo property shall contain an opaque ID that can be used to correlate instances of CIM_ComputerSystem across namespace implementations that represent the same underlying real-world system. Underlying instrumentation shall guarantee that this value is the same for any two or more instances of CIM_ComputerSystem that represent the same underlying real-world system.

7.1.2 Modifying ElementName Is Supported

The CIM_ComputerSystem.ElementName property may support being modified by the ModifyInstance operation. See section 8.4.1. This behavior is conditional upon the existence of an instance of CIM_EnabledLogicalElementCapabilities being associated with the CIM_ComputerSystem instance where the CIM_EnabledLogicalElementCapabilities.ElementNameEditSupported property has the value TRUE.

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

7.1.2.1 CIM_EnabledLogicalElementCapabilities

An instance of CIM_EnabledLogicalElementCapabilities shall be associated with the CIM_ComputerSystem instance through an instance of CIM_ElementCapabilities.

7.1.2.1.1 CIM_EnabledLogicalElementCapabilities.ElementNameEditSupported

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

7.1.2.1.2 CIM_EnabledLogicalElement.MaxElementNameLen

The MaxElementNameLen property shall be implemented.

7.1.3 Modifying ElementName Is Not Supported

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

7.1.3.1 CIM_EnabledLogicalElementCapabilities

An instance of CIM_EnabledLogicalElementCapabilities may be associated with the CIM_ComputerSystem instance through an instance of CIM_ElementCapabilities.

7.1.3.1.1 CIM_EnabledLogicalElementCapabilities.ElementNameEditSupported

The ElementNameEditSupported property shall have a value of FALSE when the implementation does not support client modification of the CIM_ComputerSystem.ElementName property.

7.1.3.1.2 CIM_EnabledLogicalElement.MaxElementNameLen

The MaxElementNameLen property may be implemented. The MaxElementNameLen property is irrelevant in this context.

EXPERIMENTAL

7.1.4 Managing System Time

A system can maintain an internal clock, which provides the system with the current time (for example, to provide time stamps for log entries). The management of the current time of the system may be supported. This behavior is optional. See section 8.2 for requirements for the ManageTime( ) method.

7.1.4.1 Managing System Time Is Supported

If the management of the current time of the system is supported, it should be implemented in conformance with this profile. If the management of the current time of the system is supported in conformance with this profile, the requirements specified in this section shall be met.

An instance of CIM_TimeService shall be associated with the Central Instance through the CIM_HostedService association. The instance of CIM_TimeService shall also be associated with the Central Instance through the CIM_ServiceAffectsElement association. Management of system time is supported when the CIM_TimeService.ManageTime( ) method is supported for at least one value for the GetRequest parameter.

7.1.4.2 Managing System Time Is Not Supported

When the management of system time is not supported, no instance of CIM_TimeService shall be associated with the Central Instance through the CIM_ServiceAffectsElement association.

EXPERIMENTAL

7.2 Management of Computer System Components

The following subclauses detail the requirements for management of components of the system.

7.2.1 Instrumentation of Processors

If the processors of the system are instrumented, the instrumentation should be conformant with the *CPU Profile*. If the processors of the system are instrumented in conformance with the *CPU Profile*, the Central Instance of the *Computer System Profile* shall be associated with the Central Instance of the *CPU Profile* through the CIM_SystemDevice association.

7.2.2 Instrumentation of System Memory

If the memory of the system is modeled, the *System Memory Profile* should be implemented. If the system memory is modeled in conformance with the *System Memory Profile*, the Central Instance of the *Computer System Profile* shall be associated with the Central Instance of the *System Memory Profile* through the CIM_SystemDevice association.

7.2.3 Instrumentation of Sensors

A system can contain one or more sensors that monitor components within the system. If the sensors of the system are instrumented, the instrumentation should be conformant with the *Sensors Profile*. If the sensors of the system are instrumented in conformance with the *Sensors Profile*, the Central Instance of the *Computer System Profile* shall be associated with the Central Instance of the *Sensors Profile* through the CIM_SystemDevice association.

7.3 Software Asset Management

This section describes behavioral requirements for the management of software asset information for the system.

7.3.1 Software Inventory Support

The inventory of software installed on or for the system may be modeled. If the inventory of software installed on or for the system is modeled, the *Software Inventory Profile* should be implemented. If the inventory of software installed on or for the system is modeled in conformance with the *Software Inventory Profile*, at least one instance of CIM_SoftwareIdentity shall be associated with the Central Instance of the *Computer System Profile* through the CIM_InstalledSoftwareIdentity association, or exactly one instance of CIM_SystemSpecificCollection shall be implemented in accordance with the requirements specified in the section "Representing Available Software" of the *Software Inventory Profile* and associated with the Central Instance of the *Computer System Profile* through the CIM_HostedCollection association.

EXPERIMENTAL

7.3.2 Software Update Support

Management of software updates for the system or components contained in the system may be supported. If the management of software updates for a component installed in the system is supported, the *Software Update Profile* should be implemented. If the management of software updates for a component installed in the system is supported in conformance with the *Software Update Profile,* the instance of a subclass of CIM_ManagedElement that represents the component shall be associated with the Central Instance of the *Software Update Profile* through the CIM_ServiceAffectsElement association.

If the management of software updates for the system is supported in conformance with the *Software Update Profile*, the Central Instance of the *Computer System Profile* shall be associated with the Central Instance of the *Software Update Profile* through the CIM_ServiceAffectsElement association.

If the system provides the ability to perform software updates for itself or other systems in conformance with the *Software Update Profile*, the Central Instance of the *Computer System Profile* shall be associated with the Central Instance of the *Software Update Profile* through the CIM_HostedService association.

EXPERIMENTAL

7.4 Network Interface Management

This section describes the requirements for the management of network interfaces of the system.

7.4.1 Ethernet Interface Management

If the Ethernet interfaces of the system are instrumented, the instrumentation should be conformant with the *Ethernet Port Profile*. If the Ethernet Interfaces of the system are instrumented in conformance with the *Ethernet Port Profile*, at least one instance of CIM_EthernetPort shall be associated with the Central Instance of the *Computer System Profile* through the CIM_SystemDevice association.

7.4.2 IP Interface Management

If the management of one or more IP interfaces of the system is supported, the *IP Interface Profile* should be implemented. If the management of one or more IP interfaces of the system is supported in conformance with the *IP Interface Profile*, the Central Instance of the *Computer System Profile* shall be associated with the Central Instance of the *IP Interface Profile* through the CIM_HostedAccessPoint association.

If the system provides the optional behavior of managing alternate configurations for the IP interface in conformance with the *IP Interface Profile*, the instance of CIM_IPConfigurationService specified by the *IP Interface Profile* shall be associated with the Central Instance of the *Computer System Profile* through the CIM_HostedService association.

7.4.3 DHCP Client Management

If the DHCP client of the system is modeled, the *DHCP Client Profile* should be implemented. If the DHCP client of the system is modeled in conformance with the *DHCP Client Profile*, at least one instance of CIM_DHCPProtocolEndpoint shall be associated with the Central Instance of the *Computer System Profile* through the CIM_HostedAccessPoint association.

7.4.4 DNS Client Management

If the DNS client of the system is modeled, the *DNS Client Profile* should be implemented. If the DNS client of the system is modeled in conformance with the *DNS Client Profile*, at least one instance of CIM_DNSProtocolEndpoint shall be associated with the Central Instance of the *Computer System Profile* through the CIM_HostedAccessPoint association.

7.5 Record Logs

Error and event information about a system can be recorded in one or more record logs. If a record log that contains information about the system is instrumented, the *Record Log Profile* should be implemented. If a record log that contains information about a system is instrumented in conformance with the *Record Log Profile*, the Central Instance of the *Computer System Profile* shall be associated with the Central Instance of the *Record Log Profile* through the CIM_UseOfLog association.

EXPERIMENTAL

7.6 Management of Protocol Services

This section describes behavioral requirements for the management of protocol services hosted on the system.

7.6.1 Hosting a CLP Service

The system may host one or more CLP services. If the system hosts at least one CLP service, the *CLP Service Profile* should be implemented. If a CLP service that is hosted by the system is modeled in conformance with the *CLP Service Profile*, the Central Instance of the *Computer System Profile* shall be associated with the Central Instance of the *CLP Service Profile* through the CIM_HostedService association.

7.6.2 Hosting an SSH Service

The system may host one or more SSH services. If the system hosts at least one SSH service, the *SSH Service Profile* should be implemented. If a SSH service that is hosted by the system is modeled in conformance with the *SSH Service Profile*, the Central Instance of the *Computer System Profile* shall be associated with the Central Instance of the *SSH Service Profile* through the CIM_HostedService association.

7.6.3 Hosting a Telnet Service

The system may host one or more telnet services. If the system hosts at least one telnet service, the *Telnet Service Profile* should be implemented. If a telnet service that is hosted by the system is modeled in conformance with the *Telnet Service Profile*, the Central Instance of the *Computer System Profile* shall be associated with the Central Instance of the *Telnet Service Profile* through the CIM_HostedService association.

EXPERIMENTAL

7.7 System Lifecycle Management

The following subclauses detail requirements related to lifecycle management of the system.

7.7.1 System State Management

This section details the requirements for representing and managing the state of a computer system.

7.7.1.1 Representing Current System State

The current state and last requested state for a computer system may be modeled using the EnabledState and RequestedState properties of CIM_ComputerSystem. This behavior is optional.

When modeling system state is supported, the CIM_ComputerSystem.EnabledState property shall have a value corresponding to the current state of the system and shall not have the value 12 (Not Applicable). The CIM_ComputerSystem.RequestedState property shall not have the value 5 (Not Applicable). The system state can change; therefore, the values of the RequestedState and EnabledState properties may still change even when the optional behavior in section 7.7.1.2 is not implemented.

When modeling system state is not supported, the CIM_ComputerSystem.EnabledState property shall have the value 12 (Not Applicable) and the CIM_ComputerSystem.RequestedState property shall have the value 5 (Not Applicable).

7.7.1.2 Client State Management Is Supported

The EnabledState and RequestedState properties and the RequestStateChange( ) method of CIM_ComputerSystem are used to perform basic lifecycle and state management of abstract systems.

Common lifecycle states and state changes (for example, enable, disable, and reset) can be managed using these CIM elements. Specializations of this profile define semantics for each state and state change specific to the management domain targeted by the specializing profile.

When management of the state of a system is supported, exactly one instance of CIM_EnabledLogicalElementCapabilities shall be associated with the CIM_ComputerSystem instance through an instance of CIM_ElementCapabilities.

Note that even when client state management is supported, the values of the RequestedState and EnabledState properties may still change implicitly to reflect state changes and requests that were not initiated by a client of the instrumentation.

Support for managing the state of the system is optional behavior. This section describes the CIM elements and behaviors that shall be implemented when this behavior is supported.

7.7.1.2.1 CIM_EnabledLogicalElementCapabilities

When state management is supported, exactly one instance of CIM_EnabledLogicalElementCapabilities shall be associated with the CIM_ComputerSystem instance through an instance of CIM_ElementCapabilities.

7.7.1.2.1.1 CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported

The RequestedStatesSupported property may contain zero or more values.

7.7.1.2.2 CIM_ComputerSystem.RequestedState

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

The CIM_ComputerSystem.RequestedState property shall have one of the values specified in the CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported property or a value of 5 (No Change). A value of 5 (No Change) shall indicate that the instrumentation is not aware of a request to change the state of the managed system.

7.7.1.2.3 CIM_ComputerSystem.EnabledState

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

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

7.7.1.3 Client State Management Is Not Supported

This section describes the CIM elements and behaviors that shall be implemented when client state management is not supported.

7.7.1.3.1 CIM_EnabledLogicalElementCapabilities

When client state management is not supported, exactly one instance of CIM_EnabledLogicalElementCapabilities may be associated with the CIM_ComputerSystem instance through an instance of CIM_ElementCapabilities.

7.7.1.3.1.1 CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported

The CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported property shall not contain any values.

7.7.2 Boot Control

This section describes the behavioral requirements for modeling and managing the boot process and configuration of the managed system.

7.7.2.1 Boot Configuration Management Is Not Supported

When management of boot configurations and the boot process is not supported for the system, the managed system may initiate its boot process when it is enabled.

7.7.2.2 Boot Configuration Management Is Supported

Management of boot configurations and the boot process may be supported for the system. This section describes the requirements when the management of boot configurations and the boot process is supported.

If the instrumentation of the boot configurations and the boot process is supported, the instrumentation should be conformant with the *Boot Control Profile*. If the instrumentation of the boot configurations and the boot process is in conformance with the *Boot Control Profile*, the Central Instance of the *Computer System Profile* shall be associated with the Central Instance of the *Boot Control Profile* through the CIM_ServiceAffectsElement association.

7.7.2.3 Hosting a Boot Service

The system may provide the ability to manage the boot configurations and control the boot process of itself or other systems. If the system provides this ability, the *Boot Control Profile* should be implemented. If the modeling of ability to manage the boot configurations and control the boot process of itself or other systems is in conformance with the *Boot Control Profile*, the Central Instance of the *Computer System Profile* shall be associated with the Central Instance of the through the CIM_HostedService association.

EXPERIMENTAL

7.8 SMASH Collections Profile

The *SMASH Collections Profile* may be implemented. If the *SMASH Collections Profile* is implemented, each instance of CIM_ConcreteCollection defined by the *SMASH Collections Profile* shall be associated with the Central Instance the *Computer System Profile* through the CIM_OwningCollectionElement association.

EXPERIMENTAL

8 Methods

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

8.1 CIM_ComputerSystem.RequestStateChange( )

Invoking the CIM_ComputerSystem.RequestStateChange( ) method changes the element’s state to the value specified in the RequestedState parameter. The values 2 (Enabled) and 3 (Disabled) of the RequestedState parameter correspond to enabling or disabling the system. A value of 2 (Enabled) shall correspond to a request to enable the system. A value of 3 (Disabled) shall correspond to a request to disable the system. A value of 11 (Reset) shall be equivalent to invoking the method with a value of 3 (Disabled), waiting for the operation to complete, and then invoking the method with a value of 2 (Enabled).

See section 7.7.1.2.2 for information about the effect of this method on the RequestedState property.

The method shall be considered successful if the state of the system upon completion of the method corresponds to the desired state indicated by the RequestedState parameter. An actual change in state does not need to occur for the method to be considered successful; the resultant state only needs to be equal to the requested state. When the method completes successfully, the return value shall be zero.

See section 7.7.1.2.3 for information about the effect of this method on the EnabledState property.

Detailed requirements of the RequestStateChange( ) method are specified in Table 3 and Table 4.

No standard messages are defined.

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

Table 3 – CIM_ComputerSystem.RequestStateChange( ) Method: Return Code Values

Value	Description
0	Request was successfully executed.
1	Method is unsupported in the implementation.
2	Error occurred
0x1000	Job started: REF returned to started CIM_ConcreteJob

Table 4 – CIM_ComputerSystem.RequestStateChange( ) Method: Parameters

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

8.1.1 CIM_ComputerSystem.RequestStateChange( ) Conditional Support

When the CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported property contains at least one value, the CIM_ComputerSystem.RequestStateChange( ) method shall be implemented and supported. The CIM_ComputerSystem.RequestStateChange( ) method shall not return a value of 1 (Not Supported).

EXPERIMENTAL

8.2 CIM_TimeService.ManageTime( )

The CIM_TimeService.ManageTime( ) method is used to query or modify the system time. When the GetRequest parameter has a value of TRUE, the TimeData parameter shall be ignored. When the ManagedElement parameter is not a reference to the Central Instance, the method shall return a value of

2. When the method is not supported for the specified value of GetRequest, the method shall return a value of 2.

When the GetRequest parameter is TRUE and the method completes successfully, the value of the TimeData parameter shall be an absolute date-time and shall not be an interval. When the value of the GetRequest parameter is FALSE, and the TimeData parameter is expressed as an interval, the method shall return a value of 2.

CIM_TimeService.ManageTime( ) method's detailed requirements shall be as specified in Table 5 and Table 6.

No standard messages are defined for this method.

Table 5 – CIM_TimeService.ManageTime( ) Method: Return Code Values

Value	Description
0	Request was successfully executed.
1	Method is not supported in the implementation.
2	Error occurred

Table 6 – CIM_TimeService.ManageTime( ) Method: Parameters

Qualifiers	Name	Type	Description/Values
IN	GetRequest	Boolean	Indicates whether the request is to get (TRUE) or set (FALSE) the time for the specified element
IN / OUT	TimeData	datetime	On input, this is the desired value for the system time. On output, this is the system time.
IN	ManagedElement	CIM_Managed Element	Reference to the Central Instance

EXPERIMENTAL

8.3 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 8.3 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
- Associators
- AssociatorNames
- References
- ReferenceNames
- EnumerateInstances
- EnumerateInstanceNames

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*.

8.4 CIM_ComputerSystem

Table 7 lists operations that either have special requirements beyond those from DSP0200 version 1.2 or shall not be supported.

Table 7 – Operations: CIM_ComputerSystem

Operation	Requirement	Messages
ModifyInstance	Optional. See section 8.4.1.	None

8.4.1 CIM_ComputerSystem—ModifyInstance

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

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

8.4.1.1 CIM_ComputerSystem.ElementName

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

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

8.5 CIM_ElementCapabilities

Table 8 lists operations that either have special requirements beyond those from DSP0200 version 1.2 or shall not be supported.

Table 8 – Operations: CIM_ElementCapabilities

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

8.6 CIM_EnabledLogicalElementCapabilities

All operations in the default list in section 8.3 are supported as described by DSP0200 version 1.2.

8.7 CIM_HostedService

Table 9 lists operations that either have special requirements beyond those from DSP0200 version 1.2 or shall not be supported.

Table 9 – Operations: CIM_HostedService

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

8.8 CIM_ServiceAffectsElement

Table 10 lists operations that either have special requirements beyond those from DSP0200 version 1.2 or shall not be supported.

Table 10 – Operations: CIM_ServiceAffectsElement

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

EXPERIMENTAL

8.9 CIM_TimeService

All operations in the default list in section 8.3 are supported as described by DSP0200 version 1.2.

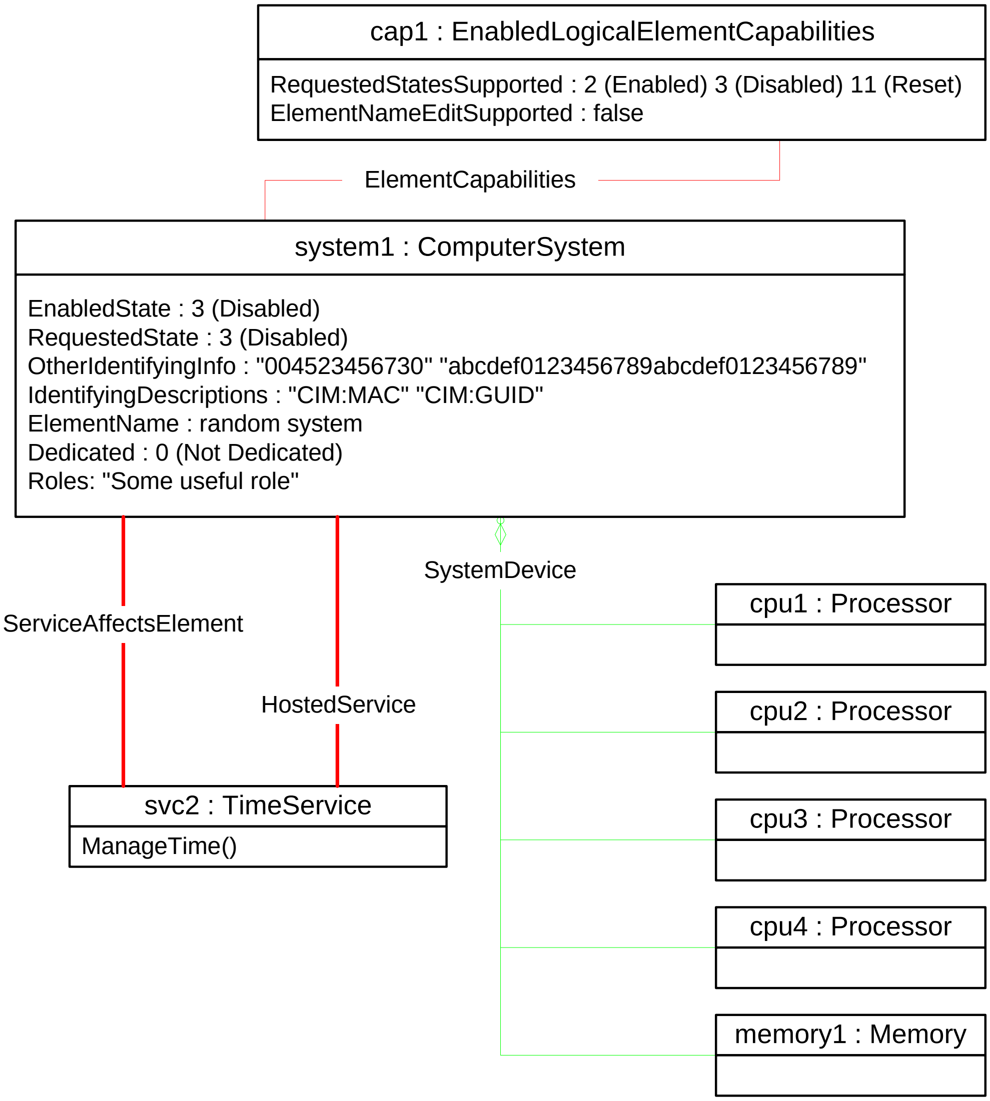
EXPERIMENTAL

9 Use Cases

The following use cases and object diagrams illustrate use of the *Computer System Profile*. They are for informational purposes only and do not introduce behavioral requirements for implementations of the profile.

9.1 Object Diagrams

The object diagram in Figure 2 shows an abstract system in which the optional state management and time management behaviors are supported as well as the *CPU Profile* and *System Memory Profile*.


cap1 : EnabledLogicalElementCapabilities
RequestedStatesSupported : 2 (Enabled) 3 (Disabled) 11 (Reset)
ElementNameEditSupported : false
ElementCapabilities
system1 : ComputerSystem
EnabledState : 3 (Disabled)
RequestedState : 3 (Disabled)
OtherIdentifyingInfo : "004523456730" "abcdef0123456789abcdef0123456789"
IdentifyingDescriptions : "CIM:MAC" "CIM:GUID"
ElementName : random system
Dedicated : 0 (Not Dedicated)
Roles: "Some useful role"
SystemDevice
ServiceAffectsElement
HostedService
svc2 : TimeService
ManageTime()
cpu1 : Processor
cpu2 : Processor
cpu3 : Processor
cpu4 : Processor
memory1 : Memory

680

681 **Figure 2 – Logical Topology**

Figure 3 is an object diagram illustrating the network interfaces of the system: *Ethernet Port Profile*, *IP Interface Profile*, *DHCP Client Profile*, and *DNS Client Profile*. The system has a single network interface.

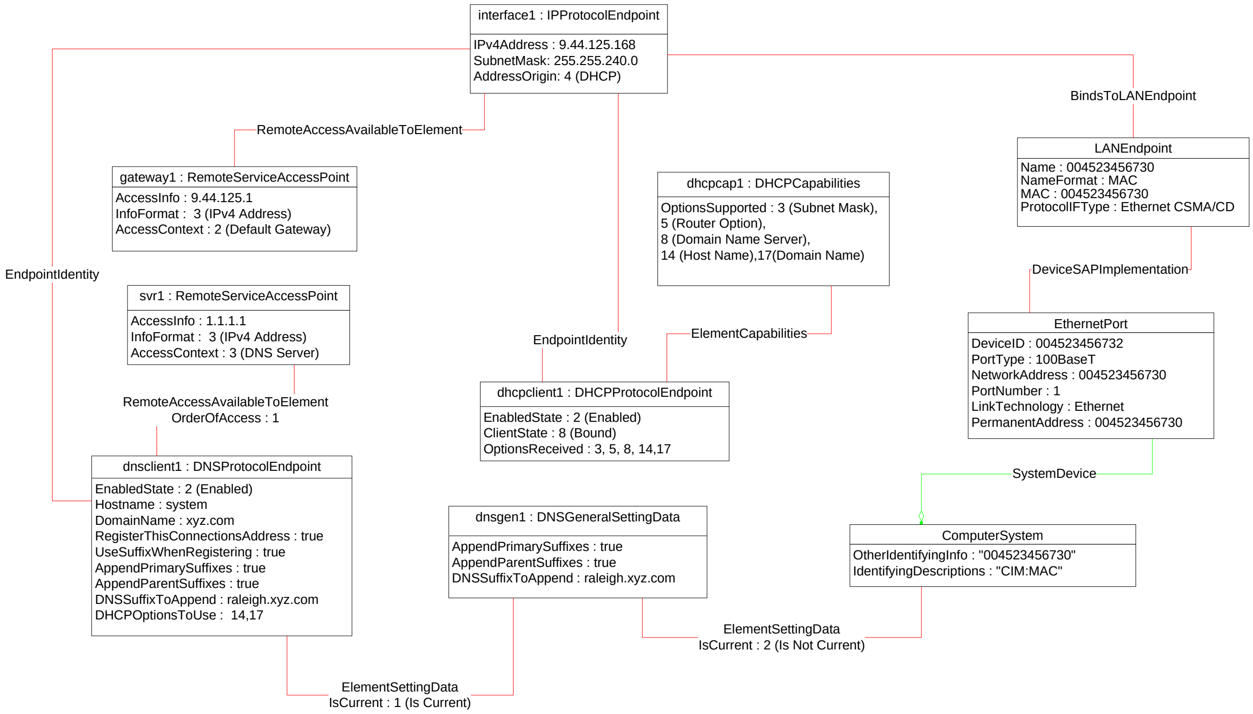

interface1 : IPProtocolEndpoint
IPv4Address : 9.44.125.168
SubnetMask: 255.255.240.0
AddressOrigin: 4 (DHCP)
BindsToLANEndpoint
LANEndpoint
Name : 004523456730
NameFormat : MAC
MAC : 004523456730
ProtocolIFType : Ethernet CSMA/CD
RemoteAccessAvailableToElement
gateway1 : RemoteServiceAccessPoint
AccessInfo : 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)
EndpointIdentity
DeviceSAPImplementation
svr1 : RemoteServiceAccessPoint
AccessInfo : 1.1.1.1
InfoFormat : 3 (IPv4 Address)
AccessContext : 3 (DNS Server)
EndpointIdentity
ElementCapabilities
EthernetPort
DeviceID : 004523456732
PortType : 100BaseT
NetworkAddress : 004523456730
PortNumber : 1
LinkTechnology : Ethernet
PermanentAddress : 004523456730
RemoteAccessAvailableToElement
OrderOfAccess : 1
dhcpclient1 : DHCPProtocolEndpoint
EnabledState : 2 (Enabled)
ClientState : 8 (Bound)
OptionsReceived : 3, 5, 8, 14,17
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
SystemDevice
dnsgen1 : DNSGeneralSettingData
AppendPrimarySuffixes : true
AppendParentSuffixes : true
DNSSuffixToAppend : raleigh.xyz.com
ComputerSystem
OtherIdentifyingInfo : "004523456730"
IdentifyingDescriptions : "CIM:MAC"
ElementSettingData
IsCurrent : 2 (Is Not Current)
ElementSettingData
IsCurrent : 1 (Is Current)

Figure 3 – Network Interfaces

9.2 Find a Dedicated System

Computer systems can have dedicated purposes or assigned roles. A client can find a system with a dedicated purpose or role by querying the value of the CIM_ComputerSystem.Dedicated or CIM_ComputerSystem.Roles property.

9.3 Correlate Instrumented Systems

For a given real system modeled with a specialization of the *Computer System Profile*, multiple implementations of the profile can exist to model the same real system within a single namespace, across namespaces, and across implementations. Across all of the namespaces to which the client has access to, starting with a single instance of CIM_ComputerSystem A that represents the real-world system, a client can find all of the other implementations of a specialization of the *Computer System Profile* that represent the same real system, as follows:

1) Form a set of identification pairs consisting of each pair of names and values from the IdentifyingDescriptions and OtherIdentifyingInfo properties of instance A*.*

2) For each CIM OM, query the Interop namespace to determine if the specialization of the *Computer System Profile* is advertised as instrumented.

3) If the specialization of the *Computer System Profile* has been instrumented, for the instance of CIM_RegisteredProfile that advertised it, find all instances of CIM_ComputerSystem associated through the CIM_ElementConformsToProfile association.

4) For each instance of CIM_ComputerSystem found in step 3), query the IdentifyingDescriptions and OtherIdentifyingInfo properties to determine if a name/value pair matches a name/value pair in the set of identification pairs found in step 1) for instance A.

5) If there is a match, then the instance of CIM_ComputerSystem from step 4) is instrumented for the same real-world system as instance A. For each name/value pair for the instance, if it is not already in the set of identification pairs known by the client for the system, add it to the set.

6) If a new identification pair was added in step 5), go back to step 4) and retest each instance of CIM_ComputerSystem.

9.4 Enable a System

A client can enable a system as follows:

1) Look for an instance of CIM_EnabledLogicalElementCapabilities 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.

9.5 Disable a System

A client can disable a system as follows:

1) Look for an instance of CIM_EnabledLogicalElementCapabilities 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.

9.6 Reset a System

A client can reset a system as follows:

1) Look for an instance of CIM_EnabledLogicalElementCapabilities 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.

9.7 Manage the System Boot Configuration

A client can verify that an instance of CIM_RegisteredProfile for the *Boot Control Profile* exists using either the central class or scoping class methodology as described in *Profile Registration Profile*. If it exists, a client can determine whether management of the system boot configuration is supported by searching for an instance of CIM_BootService that is conformant with the *Boot Control Profile* and associated with the Central Instance of the *Computer System Profile* through the CIM_ServiceAffectsElement association. The specific use cases for managing the system boot configuration are documented in the *Boot Control Profile*.

9.8 Determine the Number of Processors in the System

A client can verify that an instance of CIM_RegisteredProfile for the *CPU Profile* exists using either the central class or scoping class methodology as described in *Profile Registration Profile*. If it exists, then the CPU profile is implemented. 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 conformant with the *CPU Profile* and associated with the Central Instance of the *Computer System Profile* through the CIM_SystemDevice association.

EXPERIMENTAL

9.9 Determine If Time Management Is Supported

To determine if time management is supported, the client can look for an instance of CIM_TimeService associated with the target instance through the CIM_ServiceAffectsElement association.

9.10 Get Time for System

A client can determine the system time by first using the steps in section 9.9 to determine if time management is supported and find the associated instance of CIM_TimeService. The client can then invoke the CIM_TimeService.ManageTime( ) method, specifying a value of TRUE for the value of the GetRequest parameter and a reference to the target instance for the value of the ManagedElement parameter.

9.11 Set Time for System

A client can determine the system time by first using the steps in section 9.9 to determine if time management is supported and find the associated instance of CIM_TimeService. The client can then invoke the CIM_TimeService.ManageTime( ) method, specifying a value of FALSE for the value of the GetRequest parameter, the desired time for the value of the TimeData parameter, and a reference to the target instance for the value of the ManagedElement parameter.

EXPERIMENTAL

9.12 Determining If ElementName Can Be Modified

For a given instance of CIM_ComputerSystem, a client can determine whether the ElementName property can be modified as follows:

1) Find the CIM_EnabledLogicalElementCapabilities instance that is associated with the target instance.

2) If an instance of CIM_EnabledLogicalElementCapabilities is not found, client cannot modify the ElementName property.

3) Query the value of the ElementNameEditSupported property of the CIM_EnabledLogicalElementCapabilities instance. If the value is TRUE, the client can modify the ElementName property of the target instance.

9.13 Determining If State Management Is Supported

For a given instance of CIM_ComputerSystem, a client can determine whether state management is supported as follows:

1) Find the CIM_EnabledLogicalElementCapabilities instance that is associated with the target instance.

2) If an instance of CIM_EnabledLogicalElementCapabilities is not found, state management is not supported.

3) Query the value of the RequestedStatesSupported property. If at least one value is specified, state management is supported.

10 CIM Elements

Table 11 shows the instances of CIM Elements for this profile. Instances of the CIM elements shall be implemented as described in Table 11. Sections 7 (“Implementation”) and 8 (“Methods”) may impose additional requirements on these elements.

Table 11 – CIM Elements: Computer System Profile

Element Name	Requirement	Description
Classes		
CIM_ComputerSystem	Mandatory	See section 10.1.
CIM_ElementCapabilities	Optional	See section 10.2.
CIM_EnabledLogicalElementCapabilities	Optional	See section 10.3.
CIM_HostedService	Optional	See section 10.4.
CIM_ServiceAffectsElement	Optional	See section 10.5.
CIM_TimeService	Optional	See section 10.6. EXPERIMENTAL
Indications		
None defined in this profile		

10.1 CIM_ComputerSystem

An instance of CIM_ComputerSystem is used to represent the system. Table 12 contains the requirements for elements of this class.

Table 12 – Class: CIM_ComputerSystem

Elements	Requirement	Description
Name	Mandatory	**Key**
CreationClassName	Mandatory	**Key**
OtherIdentifyingInfo	Optional	See section 7.1.1.
IdentifyingDescriptions	Optional	See section 7.1.1.
EnabledState	Mandatory	See section 7.7.1.
RequestedState	Mandatory	See sections 7.7.1.2.2.
OperationalStatus	Mandatory	None
HealthState	Mandatory	None
ElementName	Mandatory	See sections 7.1.2 and 7.1.3.
RequestStateChange( )	Conditional	See section 8.1.

10.2 CIM_ElementCapabilities

CIM_ElementCapabilities associates an instance of CIM_EnabledLogicalElementCapabilities with an instance of CIM_ComputerSystem. Table 13 contains the requirements for elements of this class.

Table 13 – Class: CIM_ElementCapabilities

Elements	Requirement	Notes
ManagedElement	Mandatory	This property shall be a reference to an instance of CIM_ComputerSystem. Cardinality 1..*
Capabilities	Mandatory	This property shall be a reference to the instance of CIM_EnabledLogicalElementCapabilities. Cardinality 0..1

10.3 CIM_EnabledLogicalElementCapabilities

CIM_EnabledLogicalElementCapabilities indicates support for managing the state of the system. Table 14 contains the requirements for elements of this class.

Table 14 – Class: CIM_EnabledLogicalElementCapabilities

Elements	Requirement	Notes
InstanceID	Mandatory	**Key**
RequestedStatesSupported	Mandatory	See sections 7.7.1.2.1.1 and 7.7.1.3.1.1.
ElementNameEditSupported	Mandatory	See sections 7.1.2.1.1 and 7.1.3.1.1.
MaxElementNameLen	Conditional	See sections 7.1.2.1.2 and 7.1.3.1.2.

10.4 CIM_HostedService

CIM_HostedService relates the CIM_TimeService to its scoping CIM_ComputerSystem instance. Table 15 contains the requirements for elements of this class.

Table 15 – Class: CIM_HostedService

Elements	Requirement	Notes
Antecedent	Mandatory	This property shall reference the Central Instance. Cardinality 1
Dependent	Mandatory	This property shall reference CIM_TimeService. Cardinality 0..1

10.5 CIM_ServiceAffectsElement

CIM_ServiceAffectsElement associates the CIM_TimeService instance with the Central Instance. Table 16 contains the requirements for elements of this class.

Table 16 – Class: CIM_ServiceAffectsElement

Elements	Requirement	Notes
AffectedElement	Mandatory	This property shall be a reference to the Central Instance. Cardinality 1
AffectingElement	Mandatory	This property shall be a reference to an instance of CIM_TimeService. Cardinality 0..1
ElementEffects	Mandatory	Matches 5 (Manages)

EXPERIMENTAL

10.6 CIM_TimeService

CIM_TimeService manages the current time on the system. Table 17 contains the requirements for elements of this class.

Table 17 – Class: CIM_TimeService

Elements	Requirement	Notes
SystemCreationClassName	Mandatory	**Key**
SystemName	Mandatory	**Key**
CreationClassName	Mandatory	**Key**
Name	Mandatory	**Key**
ElementName	Mandatory	Pattern (“.*”). See sections 7 and 8.
ManageTime( )	Mandatory	See section 8.2.

EXPERIMENTAL

ANNEX A
(Informative)

Change Log

Version	Date	Description
1.0.0b	08/28/2006	Editorial changes for release as Preliminary Standard
1.0.0	11/10/2008	Released as a draft for Final Standard.
1.0.0	12/2/2008	Updated final version based on TC comments.
1.0.0	12/8/2008	Released as a draft for final after addressing working group ballot comments.

ANNEX B
(informative)

Acknowledgments

The authors wish to acknowledge the following people.

Editor:

- Hemal Shah – Broadcom
- Aaron Merkin – IBM

Contributors:

- Jon Hass – Dell
- Khachatur Papanyan – Dell
- Jeff Hilland – HP
- Christina Shaw – HP
- Aaron Merkin – IBM
- Perry Vincent – Intel
- John Leung – Intel
- Hemal Shah – Broadcom
- David Hines – Intel
- Jim Davis – WBEM Solutions