Document Number: DSP1010

Date: 2008-05-21

Version: 1.0.0

Record Log Profile

Document Type: Specification

Document Status: Final Standard

Document Language: E

CONTENTS

Figures

Tables

Foreword

The *Record Log Profile* (DSP1010) was prepared by the Server Management Working Group.

DMTF is a not-for-profit association of industry members dedicated to promoting enterprise and systems management and interoperability.

Introduction

This document defines classes to describe the record logs of a managed system. Also included are descriptions of the associations that can be used to associate the record log classes to DMTF profile version information. The information in this specification should 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 and manage record logs of managed systems and subsystems modeled using the DMTF CIM core and extended model definitions.

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

Record Log Profile

1 Scope

The *Record Log Profile* is an autonomous profile that provides the management capabilities to represent logs of a managed system. The log is modeled as referencing the managed elements that populate the log, and the profile registration for the schema implementation version information.

2 Normative References

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

2.1 Approved References

DMTF DSP0004, *CIM Infrastructure Specification 2.3.0*

DMTF DSP0200, *CIM Operations over HTTP 1.2.0*

DMTF DSP1000, *Management Profile Specification Template 1.0.0*

DMTF DSP1001, *Management Profile Specification Usage Guide 1.0.0*

DMTF DSP1033, *Profile Registration Profile 1.0.0*

2.2 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. The terms and definitions 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 in order to conform to the document when the specified conditions are met

3.4

mandatory

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

3.5

may

indicates a course of action permissible within the limits of the document

3.6

need not

indicates a course of action permissible within the limits of the document

3.7

optional

indicates a course of action permissible within the limits of the document

3.8

referencing profile

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 in order to conform to the document and from which no deviation is permitted

3.10

shall not

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

3.11

should

indicates that among several possibilities, one is recommended as particularly suitable, without mentioning or excluding others, or that a certain course of action is preferred but not necessarily required

3.12

should not

indicates that a certain possibility or course of action is deprecated but not prohibited

3.13

unspecified

indicates that this profile does not define any constraints for the referenced CIM element or operation

4 Symbols and Abbreviated Terms

The following symbols and abbreviations are used in this document.

4.1

LIFO

Last In, First Out

5 Synopsis

Profile Name: Record Log

Version: 1.0.0

Organization: DMTF

CIM schema version: 2.14

Central Class: CIM_RecordLog

Scoping Class: CIM_RecordLog

The *Record Log Profile* is an autonomous profile that provides the management capabilities to represent logs of a managed system.

Table 1 identifies profiles on which this profile has a dependency.

Table 1 – Referenced Profiles

Profile Name	Organization	Version	Relationship	Behavior
Profile Registration	DMTF	1.0.0	Mandatory	None

6 Description

The *Record Log Profile* describes the properties and methods of logs generated by the managed system or component. This profile describes the association between the managed system and the generated logs as well as how individual log entries are contained within a record log.

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

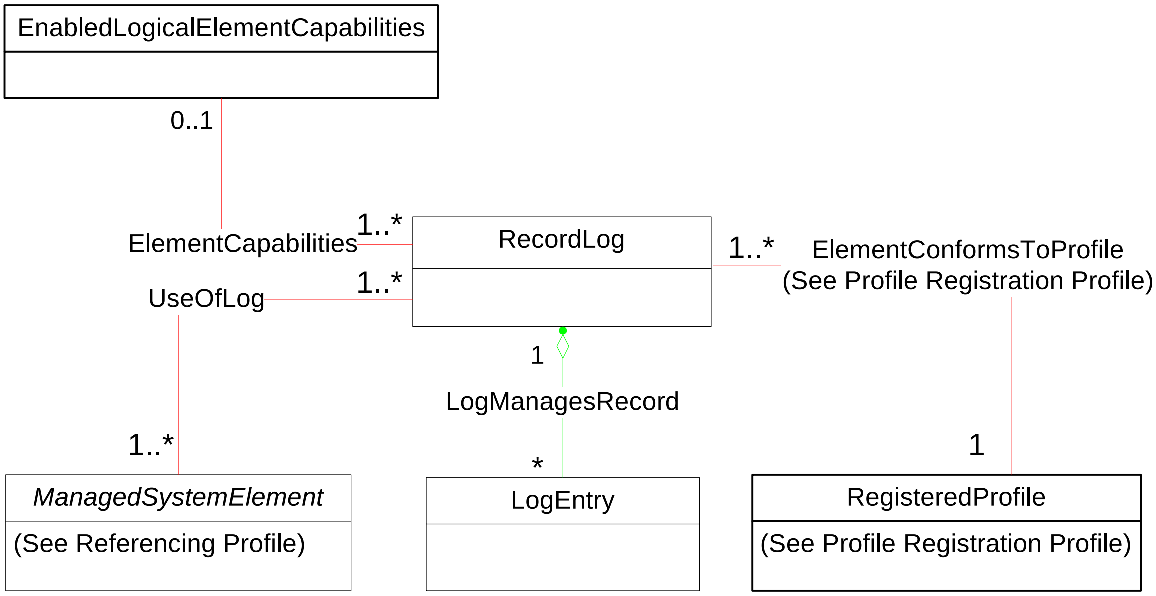


Figure 1 – Record Log Profile: Class Diagram

The CIM_RecordLog class represents the container for the log entries. The individual log entries, which are represented by the CIM_LogEntry instances, are aggregated under the CIM_RecordLog instance through the CIM_LogManagesRecord association. The managed system element that is associated with the log, uses the log, or populates the log is referenced through the CIM_UseOfLog association.

The CIM_LogEntry class contains properties describing the information about individual records, such as message text and timestamp. CIM_RecordLog describes the general properties of the log, such as its maximal length and state.

7 Implementation

This section details the requirements and guidelines related to the arrangement of instances and their properties for implementations of this profile. For a list of all required methods, see section 8 (“Methods”). For properties, see section 10 (“CIM Elements”).

7.1 Representing Logs

Each log in a managed system shall be represented by a single instance of CIM_RecordLog. Each entry in the log shall be represented by a single instance of CIM_LogEntry. The entries of the log, which are represented by the instances of CIM_LogEntry, shall be associated through the instance of CIM_LogManagesRecord to the instance of CIM_RecordLog.

7.1.1 CIM_LogEntry.LogInstanceID

The CIM_LogEntry.LogInstanceID shall have the same value as the InstanceID property of the instance of CIM_RecordLog that is associated with the instance CIM_LogEntry through an instance of CIM_LogManagesRecord.

7.1.2 CIM_LogEntry.LogName

The CIM_LogEntry.LogName shall have the same value as the ElementName property of the instance of CIM_RecordLog that is associated with the instance CIM_LogEntry through an instance of CIM_LogManagesRecord.

7.1.3 CIM_LogEntry.RecordData

The CIM_LogEntry.RecordData property should be implemented. Note that this property is not required in order to allow for alternate usage of standard messages in the future.

7.1.4 CIM_LogEntryRecordFormat

The CIM_LogEntry.RecordFormat property should be implemented. Note that this property is not required in order to allow for alternate usage of standard messages in the future.

7.2 CIM_EnabledLogicalElementCapabilities

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

There shall be at most one instance of CIM_EnabledLogicalElementCapabilities associated with a given instance of CIM_RecordLog.

7.2.1 CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported

The CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported property is an array that contains the supported requested states for the instance of CIM_RecordLog. This property shall be the super set of the values to be used as the RequestedState parameter in the RequestStateChange( ) method (see section 8.2). The value of the CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported property shall be an empty array or any combination of the following values: 2 (Enabled), 3 (Disabled), or 11 (Reset).

7.2.2 CIM_EnabledLogicalElementCapabilities.ElementNameEditSupported

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

7.2.3 CIM_EnabledLogicalElementCapabilities.MaxElementNameLen

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

7.2.4 Log State Management (Optional)

Log state management consists of the CIM_RecordLog.RequestStateChange( ) method being supported (see section 8.2) and the value of the CIM_RecordLog.RequestedState not matching 12 (Not Applicable).

7.2.5 Log State Management Support

When no CIM_EnabledLogicalElementCapabilities instance is associated with the CIM_RecordLog instance, log state management shall not be supported.

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

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

7.3 CIM_RecordLog.RequestedState

The CIM_RecordLog.RequestedState property shall have a value of 12 (Not Applicable), 5 (No Change), or a value contained in the CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported property array of the associated CIM_EnabledLogicalElementCapabilities instance (see section 7.2.1).

When log state management is supported and the RequestStateChange( ) method is successfully executed, the RequestedState property shall be set to the value of the RequestedState parameter of the RequestStateChange( ) method. After the RequestStateChange( ) method has successfully executed, the RequestedState and EnabledState parameters shall have equal values, with the exception of the transitional requested state 11 (Reset). The value of the RequestedState property may also change as a result of a non-CIM implementation’s request for a change to the log’s enabled state.

7.3.1 RequestedState—12 (Not Applicable) Value

When log state management is not supported, the value of the CIM_RecordLog.RequestedState property shall be 12 (Not Applicable).

7.3.2 RequestedState—5 (No Change) Value

When log state management is supported, the initial value of the CIM_RecordLog.RequestedState property shall be 5 (No Change).

7.4 Representing Log State

The log's state shall be represented by two properties: CIM_RecordLog.EnabledState (see section 7.4.1) and CIM_RecordLog.LogState (see section 7.4.2).

7.4.1 CIM_RecordLog.EnabledState

Table 2 describes the mapping between the values of the CIM_RecordLog.EnabledState property and the corresponding description of the state of the log. The CIM_RecordLog.EnabledState property shall match the values that are specified in Table 2. When the RequestStateChange( ) method executes but does not complete successfully and the log is in an indeterminate state, the CIM_RecordLog.EnabledState property shall have a value of 5 (Not Applicable). The value of this property may also change as a result of a non-CIM implementation's change to the log's enabled state.

Table 2 – EnabledState Value Description

Value	Description	Extended Description
2	Enabled	Log shall be enabled; new log entries may be added.
3	Disabled	Log shall be disabled; new log entries shall not be added.
5	Not Applicable	Log state is indeterminate, or the log state management is not supported.
6	Enabled but Offline	Log shall be enabled, but new log entries shall not be added. See section 7.4.1.1.

7.4.1.1 CIM_RecordLog.EnabledState—6 (Enabled but Offline) Value

When the log is enabled but has reached its maximum capacity of entries and the CIM_RecordLog.OverwritePolicy property has a value of 7 (Never Overwrites), the CIM_RecordLog.EnabledState property shall have a value of 6 (Enabled but Offline).

When the CIM_RecordLog.OverwritePolicy property has a value of 2 (Wraps When Full), the CIM_RecordLog.EnabledState property shall not have a value of 6 (Enabled but Offline).

7.4.2 CIM_RecordLog.LogState

The CIM_RecordLog.LogState property is used to describe a more granular state of the log than that of the CIM_RecordLog.EnabledState property. Table 3 describes the mapping between the values of the CIM_RecordLog.LogState property and the corresponding description of the granular state of the log. The CIM_RecordLog.LogState property shall match the values that are specified in Table 3. Additionally, Table 3 describes the mapping between the LogState property and the EnabledState property. When the CIM_RecordLog.LogState property has a value that matches the value in the "LogState Value" column in Table 3, the CIM_RecordLog.EnabledState property shall have a value that matches the value in the "EnabledState Value" column for that row.

Table 3 – LogState Value Description and Mapping to EnabledState Value

LogState Value	Description	EnabledState Value	Extended Description
0	Unknown	5 (Not Applicable)	See the “Extended Description” column of Table 2 for the corresponding EnabledState value.
2	Normal	2 (Enabled)	See the “Extended Description” column of Table 2 for the corresponding EnabledState value.
3	Erasing	Any value in Table 2	Log shall be in the process of erasing its entries. See section 7.4.2.1.
4	Not Applicable	Any value in Table 2	LogState property is not supported, and EnabledState property shall be used only to represent the log state.

7.4.2.1 CIM_RecordLog.LogState—3 (Erasing) Value

The instrumentation may be able to represent the transitional states of the log, such as the state when the log entries are being cleared. When the log is being cleared through the invocation of the ClearLog( ) method or by a non-CIM implementation, the CIM_RecordLog.LogState property shall have a value of 3 (Erasing).

7.5 CIM_UseOfLog

At least one instance of the CIM_UseOfLog association shall reference an instance of CIM_RecordLog and an instance of the subclass of CIM_ManagedSystemElement.

7.6 CIM_RecordLog.OverwritePolicy Property

The CIM_RecordLog.OverwritePolicy property indicates the behavior of the log when it has reached the maximum capacity of its entries. The CIM_RecordLog.OverwritePolicy property also affects the CIM_RecordLog.EnabledState property, as described in section 7.4.1.1.

The log could be designed such that when the log reaches its maximum capacity, new entries would overwrite the oldest entries. An example of this type of log would be circular buffer logs.

When the new log entries overwrite the old log entries, the CIM_RecordLog.OverwritePolicy property has a value of 2 (Wraps When Full). When the new log entries never overwrite the old log entries, the CIM_RecordLog.OverwritePolicy property has a value of 7 (Never Overwrites).

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_RecordLog.ClearLog( )

The CIM_RecordLog.ClearLog( ) method is used to request the deletion of all entries in the record log for an instance of CIM_RecordLog. A return code value of zero shall indicate that the clearing of the log entries was successfully initiated.

CIM_RecordLog.ClearLog( ) return code values shall be as specified in Table 4.

No parameters or standard messages are defined for the CIM_RecordLog.ClearLog( ) method.

Table 4 – CIM_RecordLog.ClearLog( ) Method: Return Code Values

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

8.2 CIM_RecordLog.RequestStateChange( )

Invocation of the CIM_RecordLog.RequestStateChange( ) method shall attempt to change the element's state to the value that is specified in the RequestedState parameter.

Return code values for the RequestStateChange( ) method shall be as specified in Table 5 where the method-execution behavior matches the return-code description. Parameters for the RequestStateChange( ) method are specified in Table 6.

When log state management is supported, the RequestStateChange( ) method shall be implemented and shall not return a value of 1 (Not Supported) (see section 7.2.5).

When the RequestedState parameter is set to 2 (Enabled) but the CIM_RecordLog.EnabledState property has a value of 6 (Enabled but Offline), the RequestStateChange() method invocation shall return 2 (Error Occurred).

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

No standard messages are defined for this method.

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

Value	Description
0	Request was successfully executed.
1	Method is not supported in the implementation.
2	Error occurred
4096	Job started

Table 6 – CIM_RecordLog.RequestStateChange( ) Method: Parameters

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

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_ElementCapabilities

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_ElementCapabilities

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

8.5 CIM_EnabledLogicalElementCapabilities

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

8.6 CIM_RecordLog

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_RecordLog

Operation	Requirement	Messages
ModifyInstance	Optional: See section 8.6.1.	None

8.6.1 CIM_RecordLog—ModifyInstance

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

The ModifyInstance operation shall be supported and CIM_RecordLog.ElementName shall be modifiable when the ElementNameEditSupported property of the CIM_EnabledLogicalElementCapabilities instance that is associated with the CIM_RecordLog instance has a value of TRUE. See section 8.6.1.1.

8.6.1.1 CIM_RecordLog.ElementName

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

When the associated CIM_EnabledLogicalElementCapabilities instance does not exist or the ElementNameEditSupported property of the associated CIM_EnabledLogicalElementCapabilities instance has a value of FALSE, the implementation shall not allow the ModifyInstance operation to change the value of the ElementName property of the CIM_RecordLog instance.

8.7 CIM_LogEntry

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_LogEntry

Operation	Requirement	Messages
DeleteInstance	Optional: See section 8.7.1 for additional requirements.	None

8.7.1 CIM_LogEntry DeleteInstance

CIM_LogEntry DeleteInstance operation shall be optional. The implementation shall also remove any association instances that reference the instance of CIM_LogEntry, including the instance of CIM_LogManagesRecord.

8.8 CIM_UseOfLog

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_UseOfLog

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

8.9 CIM_LogManagesRecord

Table 11 lists operations that either have special requirements beyond those from DSP0200 version 1.2 or shall not be supported.

Table 11 – Operations: CIM_LogManagesRecord

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

9 Use Cases

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

9.1 Object Diagrams

Figure 2 represents possible instances of *Record Log Profile* classes. In this case, system1 uses log1 for its hardware log. log1 has only one record, but it has a maximum capacity of 64 records. The value of the EnabledState property for log1 is 1 (Enabled), which means the log is active. Profile registration information is represented with the profile1 instance.

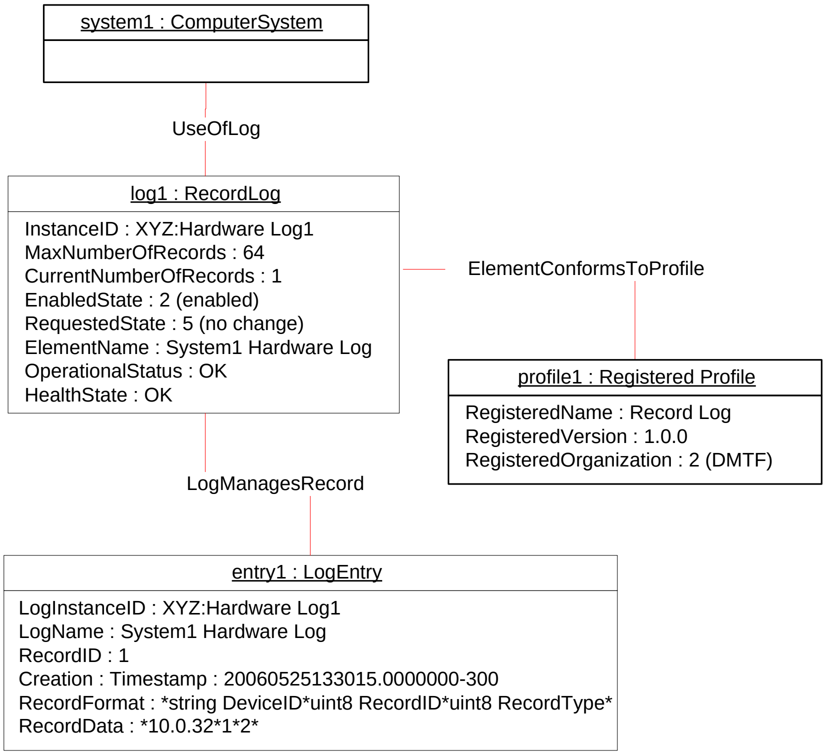
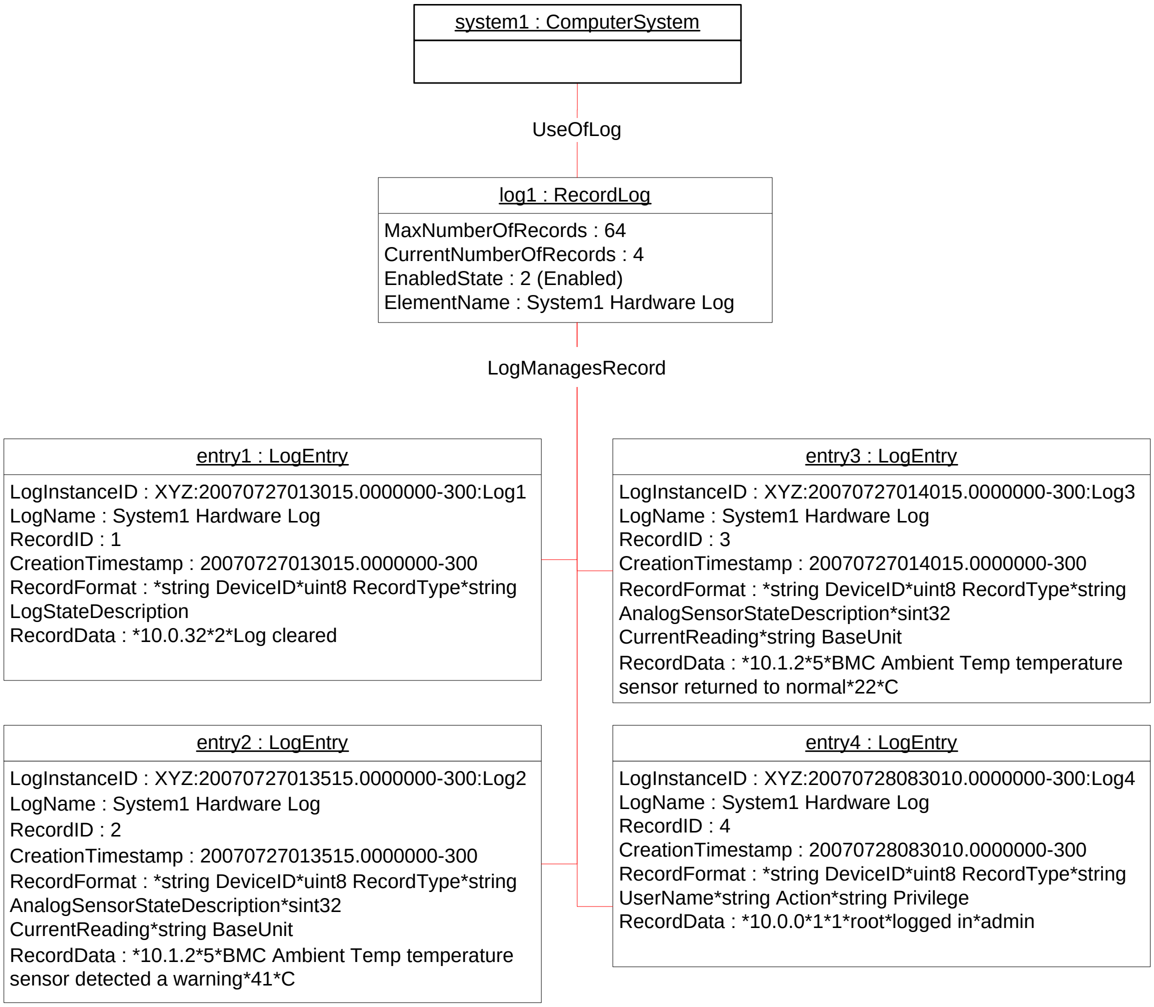

system1 : ComputerSystem
UseOfLog
log1 : RecordLog
InstanceID : XYZ:Hardware Log1
MaxNumberOfRecords : 64
CurrentNumberOfRecords : 1
EnabledState : 2 (enabled)
RequestedState : 5 (no change)
ElementName : System1 Hardware Log
OperationalStatus : OK
HealthState : OK
ElementConformsToProfile
profile1 : Registered Profile
RegisteredName : Record Log
RegisteredVersion : 1.0.0
RegisteredOrganization : 2 (DMTF)
LogManagesRecord
entry1 : LogEntry
LogInstanceID : XYZ:Hardware Log1
LogName : System1 Hardware Log
RecordID : 1
Creation : Timestamp : 20060525133015.0000000-300
RecordFormat : *string DeviceID*uint8 RecordID*uint8 RecordType*
RecordData : *10.0.32*1*2*

Figure 2 – RecordLog Instance

Figure 3 represents a possible instantiation of the *Record Log Profile*. log1, which is the hardware log for system1, has four log entries. entry1 is a log entry for clearing the log, entry2 and entry3 are sensor logged information, and entry4 contains information about the logged-in users. If the ClearLog( ) method is supported on log1, the client might execute the ClearLog( ) method on log1 to erase the entries. Depending on the log1 settings, some of the entries may not be erasable through executing the ClearLog( ) method. Figure 4 shows the change of instances of CIM_LogEntry after the successful execution of the ClearLog( ) method on log1.



log1 : RecordLog
MaxNumberOfRecords : 64
CurrentNumberOfRecords : 4
EnabledState : 2 (Enabled)
ElementName : System1 Hardware Log

entry1 : LogEntry
LogInstanceID : XYZ:20070727013015.0000000-300:Log1
LogName : System1 Hardware Log
RecordID : 1
CreationTimestamp : 20070727013015.0000000-300
RecordFormat : *string DeviceID*uint8 RecordType*string LogStateDescription
RecordData : *10.0.32*2*Log cleared

entry3 : LogEntry
LogInstanceID : XYZ:20070727014015.0000000-300:Log3
LogName : System1 Hardware Log
RecordID : 3
CreationTimestamp : 20070727014015.0000000-300
RecordFormat : *string DeviceID*uint8 RecordType*string AnalogSensorStateDescription*sint32 CurrentReading*string BaseUnit
RecordData : *10.1.2*5*BMC Ambient Temp temperature sensor returned to normal*22*C

entry2 : LogEntry
LogInstanceID : XYZ:20070727013515.0000000-300:Log2
LogName : System1 Hardware Log
RecordID : 2
CreationTimestamp : 20070727013515.0000000-300
RecordFormat : *string DeviceID*uint8 RecordType*string AnalogSensorStateDescription*sint32 CurrentReading*string BaseUnit
RecordData : *10.1.2*5*BMC Ambient Temp temperature sensor detected a warning*41*C

entry4 : LogEntry
LogInstanceID : XYZ:20070728083010.0000000-300:Log4
LogName : System1 Hardware Log
RecordID : 4
CreationTimestamp : 20070728083010.0000000-300
RecordFormat : *string DeviceID*uint8 RecordType*string UserName*string Action*string Privilege
RecordData : *10.0.0*1*1*root*logged in*admin

Figure 3 – RecordLog Instance Before the Log Is Cleared

Figure 4 shows the representation of log1 after the ClearLog() method successfully executed. entry1, entry2, and entry3 from Figure 3 have been erased. Because of log1's policies, entry4 has not been erased and still exists in log1. entry5 is a new log entry that has been added to the log after the successful clearing of log1. Note that the RecordID properties have been reset for the entries in log1.

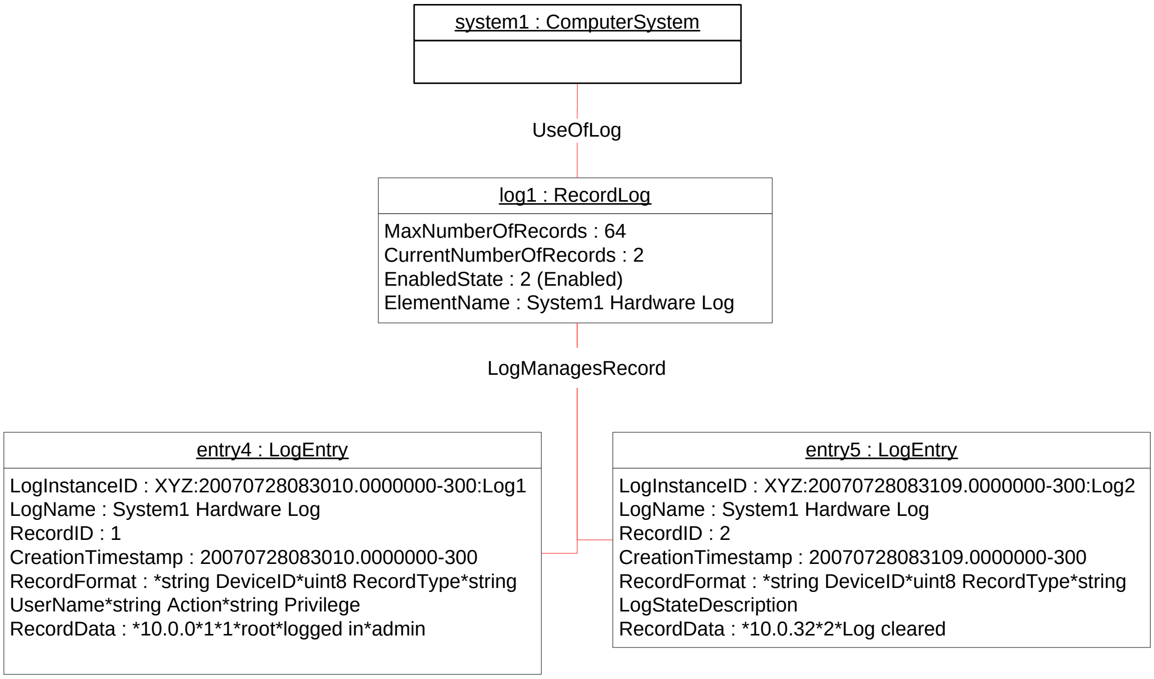

system1 : ComputerSystem
UseOfLog
log1 : RecordLog
MaxNumberOfRecords : 64
CurrentNumberOfRecords : 2
EnabledState : 2 (Enabled)
ElementName : System1 Hardware Log
LogManagesRecord
entry4 : LogEntry
LogInstanceID : XYZ:20070728083010.0000000-300:Log1
LogName : System1 Hardware Log
RecordID : 1
CreationTimestamp : 20070728083010.0000000-300
RecordFormat : *string DeviceID*uint8 RecordType*string UserName*string Action*string Privilege
RecordData : *10.0.0*1*1*root*logged in*admin
entry5 : LogEntry
LogInstanceID : XYZ:20070728083109.0000000-300:Log2
LogName : System1 Hardware Log
RecordID : 2
CreationTimestamp : 20070728083109.0000000-300
RecordFormat : *string DeviceID*uint8 RecordType*string LogStateDescription
RecordData : *10.0.32*2*Log cleared

Figure 4 – RecordLog Instance after the Log Is Cleared

9.2 Identify the Log by the Name

To select a log by its name, a client can select the CIM_RecordLog instance in which the ElementName property corresponds to the desired name.

9.3 Browse the Records of the Log

To browse log records, a client can iterate through all the instances of CIM_LogEntry that are associated through the CIM_LogManagesRecord association to the given instance of CIM_RecordLog and sort them based on the RecordID.

9.4 Sort the Log Records Based on the Time Stamp of the Log Entry

A client can sort log records by time stamp as follows:

1. Iterate through all the instances of CIM_LogEntry that are associated through the CIM_LogManagesRecord association to the given instance of CIM_RecordLog that represents the log record.

2. Sort the instances of CIM_LogEntry based on the CreationTimeStamp property value in LIFO order.

9.5 Delete a Log Entry

A client can delete a log entry as follows:

1. Select the instance of CIM_LogEntry that represents the desired log entry to be deleted.
2. Execute DeleteInstance operation on the selected instance of CIM_LogEntry.

Upon successful execution, the instance of CIM_LogEntry and the instance of CIM_LogManagesRecord that associates the log entry to the instance of CIM_RecordLog are deleted.

9.6 Clear the Log

To clear the log, a client can execute the ClearLog() method for the given instance of CIM_RecordLog.

10 CIM Elements

Table 12 shows the instances of CIM Elements for this profile. Instances of the CIM Elements shall be implemented as described in Table 12. Sections 7 (“Implementation”) and 8 (“Methods”) may impose additional requirements on these elements.

Table 12 – CIM Elements: Record Log Profile

Element Name	Requirement	Description
Classes		
CIM_ElementCapabilities	Conditional	See section 10.1.
CIM_EnabledLogicalElementCapabilities	Optional	See section 10.2.
CIM_LogManagesRecord	Conditional	See section 10.3.
CIM_LogEntry	Optional	See section 10.4.
CIM_RecordLog	Mandatory	See section 10.5.
CIM_RegisteredProfile	Mandatory	See section 10.6.
CIM_UseOfLog	Mandatory	See section 10.7.
Indications		
None defined in this profile		

10.1 CIM_ElementCapabilities

CIM_ElementCapabilities associates an instance of CIM_RecordLog with an instance of CIM_EnabledLogicalElementCapabilities that describes the capabilities of CIM_RecordLog. CIM_ElementCapabilities is mandatory when the implementation instantiates an instance of CIM_EnabledLogicalElementCapabilities that represents the capabilities of the log.

Table 13 – Class: CIM_ElementCapabilities

Elements	Requirement	Notes
ManagedElement	Mandatory	**Key:** This property shall reference the instance of CIM_RecordLog that represents the log. Cardinality 1..*, indicating one or many references
Capabilities	Mandatory	**Key:** This property shall reference the instance of CIM_EnabledLogicalElement that represents the capabilities of the log. Cardinality 0..1, indicating zero or one reference

10.2 CIM_EnabledLogicalElementCapabilities

CIM_EnabledLogicalElementCapabilities represents the capabilities of the log.

Table 14 – Class: CIM_EnabledLogicalElementCapabilities

Elements	Requirement	Notes
InstanceID	Mandatory	**Key**
RequestedStatesSupported	Mandatory	See section 7.2.1.
ElementNameEditSupported	Mandatory	See Section 7.2.2.
MaxElementNameLen	Conditional	See Section 7.2.3.

10.3 CIM_LogManagesRecord

CIM_LogManagesRecord associates the CIM_RecordLog instance, which represents the log, with an instance of CIM_LogEntry, which represents an entry within the log. CIM_LogManagesRecord is mandatory when at least one instance of CIM_LogEntry exists.

Table 15 – Class: CIM_LogManagesRecord

Elements	Requirement	Notes
Log	Mandatory	**Key:** This property shall reference the CIM_RecordLog instance that represents the log. Cardinality 1, indicating one reference
Record	Mandatory	**Key:** This property shall reference the instance of CIM_LogEntry that represents the entry within the log. Cardinality *, indicating many references

10.4 CIM_LogEntry

CIM_LogEntry represents the log entry within the log in the managed system.

Table 16 – Class: CIM_LogEntry

Elements	Requirement	Notes
InstanceID	Mandatory	**Key**
LogInstanceID	Optional	See section 7.1.1.
LogName	Optional	See section 7.1.2.
RecordID	Mandatory	None
CreationTimestamp	Mandatory	None
RecordData	Optional	See section 7.1.3.
RecordFormat	Optional	See section 7.1.4.
ElementName	Mandatory	The property shall match pattern “.*”.

10.5 CIM_RecordLog

CIM_RecordLog represents the log in the managed system.

Table 17 – Class: CIM_RecordLog

Elements	Requirement	Notes
InstanceID	Mandatory	**Key**
MaxNumberOfRecords	Mandatory	A value of 0 shall mean “Unknown” or “Not Applicable”.
LogState	Mandatory	See section 7.4.2.
OverwritePolicy	Mandatory	See section 7.6.
RequestedState	Mandatory	See section 7.3.
EnabledState	Mandatory	See section 7.4.1.
OperationalStatus	Mandatory	None
HealthState	Mandatory	None
ElementName	Mandatory	The property shall match pattern “.*”.

10.6 CIM_RegisteredProfile

CIM_RegisteredProfile identifies the *Record Log Profile* in order for a client to determine the conformance with the 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 18, the behavior of the RegisteredProfile instance is per the *Profile Registration Profile*.

Table 18 – Class: CIM_RegisteredProfile

Elements	Requirement	Description
RegisteredName	Mandatory	This property shall have a value of “Record Log”.
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 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.

10.7 CIM_UseOfLog

CIM_UseOfLog associates CIM_RecordLog, which represents the log, with a subclass of CIM_ManagedSystemElement, which represents the element that uses or populates the log.

Table 19 – Class: CIM_UseOfLog

Elements	Requirement	Notes
Antecedent	Mandatory	**Key:** This property shall reference the CIM_RecordLog instance that represents the log. Cardinality 1..*, indicating one or many references
Dependent	Mandatory	**Key:** This property shall reference the instance of a subclass of CIM_ManagedSystemElement (such as CIM_ComputerSystem) that owns the log. Cardinality 1..*, indicating one or many references

ANNEX A
(informative)

Change Log

Version	Date	Description
1.0.0b	2006/08/16	Preliminary Standard version.
1.0.0c	2007/02/14	Preliminary Standard refresh. Updated the value/valuemaps of CIM_RecordLog.OverwrityPolicy and updated the CIM schema version from 2.11 to 2.14 to reflect the corresponding schema containing the change mentioned.
1.0.0	2007/10/04	Final Standard version

ANNEX B
(informative)

Acknowledgements

The authors wish to acknowledge the following people.

Editor:

- Jon Hass – Dell
- Khachatur Papanyan – Dell

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