



1

2

3

4

**Document Number: DSP1010**

**Date: 2010-05-20**

**Version: 2.0.0**

## 5 **Record Log Profile**

6 **Document Type: Specification**

7 **Document Status: DMTF Standard**

8 **Document Language: en-US**

9

10 Copyright notice

11 Copyright © 2008, 2010 Distributed Management Task Force, Inc. (DMTF). All rights reserved.

12 DMTF is a not-for-profit association of industry members dedicated to promoting enterprise and systems  
13 management and interoperability. Members and non-members may reproduce DMTF specifications and  
14 documents, provided that correct attribution is given. As DMTF specifications may be revised from time to  
15 time, the particular version and release date should always be noted.

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26 withdrawn or modified after publication, and shall be indemnified and held harmless by any party  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

29 For information about patents held by third-parties which have notified the DMTF that, in their opinion,  
30 such patent may relate to or impact implementations of DMTF standards, visit  
31 <http://www.dmtf.org/about/policies/disclosures.php>.

32

# CONTENTS

34	Foreword .....	5
35	Introduction .....	6
36	1 Scope .....	7
37	2 Normative References.....	7
38	3 Terms and Definitions .....	7
39	4 Symbols and Abbreviated Terms .....	8
40	5 Synopsis.....	8
41	6 Description .....	9
42	7 Implementation.....	10
43	7.1 Representing Logs.....	10
44	7.2 CIM_RecordLogCapabilities .....	12
45	7.3 Log State Management (Optional).....	13
46	7.4 CIM_RecordLog.RequestedState.....	13
47	7.5 Representing Log State .....	13
48	7.6 CIM_UseOfLog .....	15
49	7.7 CIM_HostedDependency.....	15
50	7.8 CIM_RecordLog.OverwritePolicy Property .....	15
51	8 Methods.....	15
52	8.1 CIM_RecordLog.ClearLog( ).....	15
53	8.2 CIM_RecordLog.RequestStateChange( ).....	16
54	8.3 Profile Conventions for Operations.....	16
55	8.4 CIM_ElementCapabilities .....	17
56	8.5 CIM_RecordLogCapabilities .....	17
57	8.6 CIM_RecordLog.....	17
58	8.7 CIM_LogEntry .....	18
59	8.8 CIM_UseOfLog .....	18
60	8.9 CIM_LogManagesRecord.....	19
61	8.10 CIM_HostedDependency.....	19
62	9 Use Cases.....	19
63	9.1 Object Diagrams .....	19
64	9.2 Identify the Log by the Name.....	23
65	9.3 Browse the Records of the Log .....	23
66	9.4 Sort the Log Records Based on the Time Stamp of the Log Entry .....	23
67	9.5 Delete a Log Entry .....	23
68	9.6 Clear the Log .....	23
69	9.7 Determine Which Record Types Are Supported .....	23
70	9.8 RecordLog Instance for Standard Messages .....	23
71	9.9 RecordLog Instance for Standard Messages with MessageArguments Supported.....	25
72	9.10 RecordLog Instance for Record Data and Standard Messages.....	27
73	9.11 List All Logs Hosted on This System .....	28
74	10 CIM Elements.....	30
75	10.1 CIM_ElementCapabilities .....	30
76	10.2 CIM_RecordLogCapabilities .....	31
77	10.3 CIM_LogManagesRecord.....	31
78	10.4 CIM_LogEntry .....	31
79	10.5 CIM_RecordLog.....	32
80	10.6 CIM_RegisteredProfile.....	32
81	10.7 CIM_UseOfLog .....	33
82	10.8 CIM_HostedDependency.....	33
83	ANNEX A (informative) Change Log.....	34

84

85 **Figures**

86	Figure 1 – Record Log Profile: Class Diagram .....	9
87	Figure 2 – RecordLog Instance.....	20
88	Figure 3 – RecordLog Instance Before the Log Is Cleared .....	21
89	Figure 4 – RecordLog Instance after the Log Is Cleared.....	22
90	Figure 5 – RecordLog Instance for Standard Messages .....	24
91	Figure 6 – RecordLog with Standard Message and MessageArguments .....	26
92	Figure 7 – RecordLog Instances for Both Record Types.....	27
93	Figure 8 – Record Log Hosted on system1 .....	28
94	Figure 9 – Record Logs Hosted on system1 and device1 .....	29

95

96 **Tables**

97	Table 1 – Related Profiles.....	9
98	Table 2 – EnabledState Value Description .....	14
99	Table 3 – LogState Value Description and Mapping to EnabledState Value .....	14
100	Table 4 – CIM_RecordLog.ClearLog( ) Method: Return Code Values .....	15
101	Table 5 – CIM_RecordLog.RequestStateChange( ) Method: Return Code Values .....	16
102	Table 6 – CIM_RecordLog.RequestStateChange( ) Method: Parameters .....	16
103	Table 7 – Operations: CIM_ElementCapabilities .....	17
104	Table 8 – Operations: CIM_RecordLog .....	17
105	Table 9 – Operations: CIM_LogEntry .....	18
106	Table 10 – Operations: CIM_UseOfLog.....	18
107	Table 11 – Operations: CIM_LogManagesRecord .....	19
108	Table 12 – Operations: CIM_HostedDependency .....	19
109	Table 13 – CIM Elements: Record Log Profile .....	30
110	Table 14 – Class: CIM_ElementCapabilities.....	30
111	Table 15 – Class: CIM_RecordLogCapabilities .....	31
112	Table 16 – Class: CIM_LogManagesRecord.....	31
113	Table 17 – Class: CIM_LogEntry .....	31
114	Table 18 – Class: CIM_RecordLog.....	32
115	Table 19 – Class: CIM_RegisteredProfile.....	32
116	Table 20 – Class: CIM_UseOfLog .....	33
117	Table 21 – Class: CIM_HostedDependency.....	33

118

119

## Foreword

120 The *Record Log Profile* (DSP1010) was prepared by the Physical Platform Profiles Working Group.

121 DMTF is a not-for-profit association of industry members dedicated to promoting enterprise and systems  
122 management and interoperability. For information about the DMTF, see <http://www.dmtf.org>.

### 123 Acknowledgments

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

125 Editors:

- 126 • Jim Davis – WBEM Solutions
- 127 • Jon Hass – Dell
- 128 • Deb McDonald – IBM
- 129 • Khachatur Papanyan – Dell

130 Contributors:

- 131 • Jon Hass – Dell
- 132 • Jeff Hilland – HP
- 133 • John Leung – Intel
- 134 • Jeff Lynch – IBM
- 135 • Aaron Merkin – IBM
- 136 • Khachatur Papanyan – Dell
- 137 • Hemal Shah – Broadcom
- 138 • Christina Shaw – HP
- 139 • Perry Vincent – Intel

140

141

## Introduction

142 This document defines classes to describe the record logs of a managed system element. Also included  
143 are descriptions of the associations that can be used to associate the record log classes to DMTF profile  
144 version information. The information in this specification should be sufficient for a provider or consumer of  
145 this data to identify unambiguously the classes, properties, methods, and values that shall be instantiated  
146 and manipulated to represent and manage record logs of managed system elements and subsystems  
147 modeled using the DMTF CIM core and extended model definitions.

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

### 150 Document conventions

#### 151 Typographical conventions

152 The following typographical conventions are used in this document:

- 153 • Document titles are marked in *italics*.

154

# Record Log Profile

## 155 1 Scope

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

## 159 2 Normative References

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

164 DMTF DSP0004, *CIM Infrastructure Specification 2.5*,  
165 [http://www.dmtf.org/standards/published\\_documents/DSP0004\\_2.5.pdf](http://www.dmtf.org/standards/published_documents/DSP0004_2.5.pdf)

166 DMTF DSP0200, *CIM Operations over HTTP 1.3*,  
167 [http://www.dmtf.org/standards/published\\_documents/DSP0200\\_1.3.pdf](http://www.dmtf.org/standards/published_documents/DSP0200_1.3.pdf)

168 DMTF DSP0228, *Message Registry XML Schema 1.1*,  
169 [http://schemas.dmtf.org/wbem/messageregistry/1/dsp0228\\_1.1.xsd](http://schemas.dmtf.org/wbem/messageregistry/1/dsp0228_1.1.xsd)

170 DMTF DSP1001, *Management Profile Specification Usage Guide 1.0*,  
171 [http://www.dmtf.org/standards/published\\_documents/DSP1001\\_1.0.pdf](http://www.dmtf.org/standards/published_documents/DSP1001_1.0.pdf)

172 DMTF DSP1033, *Profile Registration Profile 1.0*,  
173 [http://www.dmtf.org/standards/published\\_documents/DSP1033\\_1.0.pdf](http://www.dmtf.org/standards/published_documents/DSP1033_1.0.pdf)

174 DMTF DSP8007, *Platform Message Registry 1.1*,  
175 [http://schemas.dmtf.org/wbem/messageregistry/1/dsp8007\\_1.1.xml](http://schemas.dmtf.org/wbem/messageregistry/1/dsp8007_1.1.xml)

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

## 178 3 Terms and Definitions

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

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

187 The terms "clause," "subclause," "paragraph," and "annex" in this document are to be interpreted as  
188 described in [ISO/IEC Directives, Part 2](#), Clause 5.

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

192 The terms defined in [DSP0004](#), [DSP0200](#), and [DSP1001](#) apply to this document. The following additional  
193 terms are used in this document.

### 194 3.1

#### 195 Record Data Format

196 refers to a log entry where the LogEntry data is contained in RecordData property whose data structure is  
197 described by RecordFormat property.

### 198 3.2

#### 199 referencing profile

200 indicates a profile that owns the definition of this class and can include a reference to this profile in its  
201 "Referenced Profiles" table

### 202 3.3

#### 203 Standard Message Format

204 refers to a log entry where the LogEntry data is contained in Message and/or MessageArguments  
205 properties

### 206 3.4

#### 207 unspecified

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

## 209 4 Symbols and Abbreviated Terms

210 The following symbols and abbreviations are used in this document.

### 211 4.1

#### 212 LIFO

213 Last In, First Out

## 214 5 Synopsis

215 **Profile Name:** Record Log

216 **Version:** 2.0.0

217 **Organization:** DMTF

218 **CIM Schema Version:** 2.25

219 **Central Class:** CIM\_RecordLog

220 **Scoping Class:** CIM\_RecordLog

221 The *Record Log Profile* is an autonomous profile that provides the management capabilities to represent  
222 logs of a managed system element. Version 2.0.0 of the *Record Log Profile* adds the standard message  
223 format capability to the logs. The CIM\_EnabledLogicalElementCapabilities class that was previously  
224 Optional was subclassed as CIM\_RecordLogCapabilities and is now Mandatory.

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



226

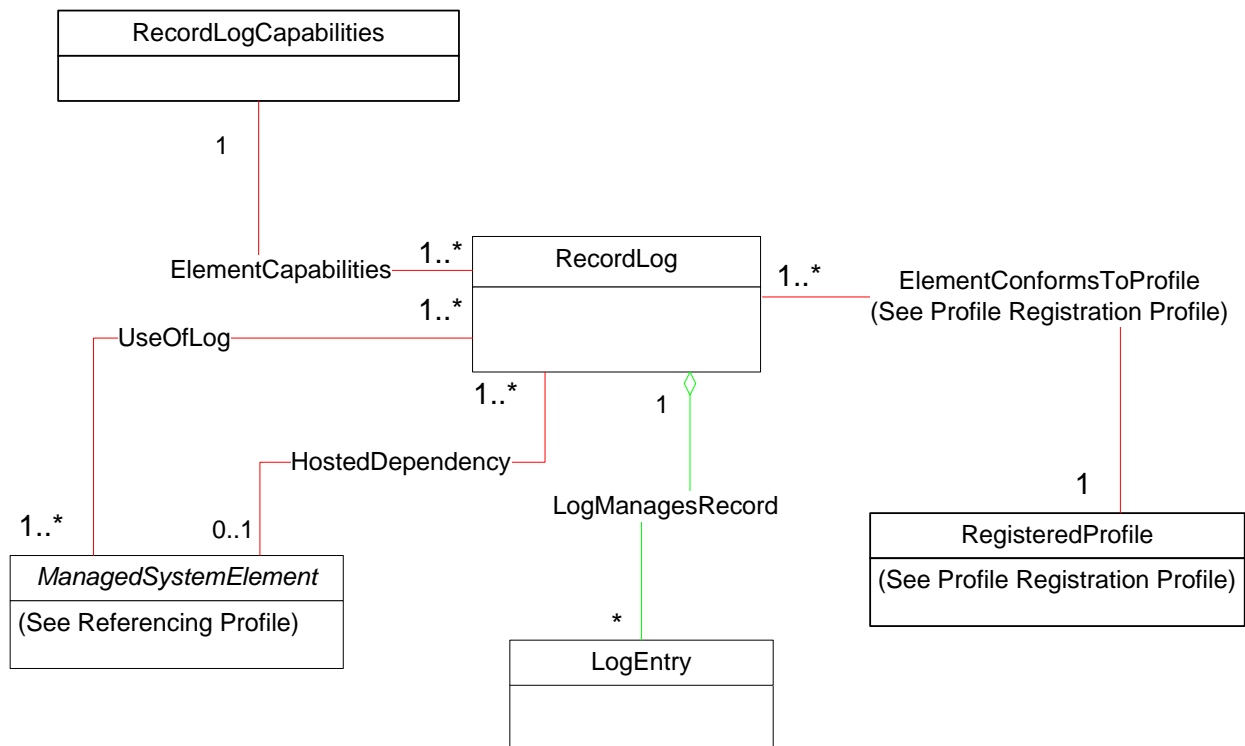
Table 1 – Related Profiles

Profile Name	Organization	Version	Requirement	Description
Profile Registration	DMTF	1.0.	Mandatory	None

227 **6 Description**

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

231 Figure 1 represents the class schema for the *Record Log Profile*. For simplicity, the prefix CIM\_ has been  
 232 removed from the names of the classes.



233

234 **Figure 1 – Record Log Profile: Class Diagram**

235 The CIM\_RecordLog class represents the container for the log entries. The individual log entries, which  
 236 are represented by the CIM\_LogEntry instances, are aggregated under the CIM\_RecordLog instance  
 237 through the CIM\_LogManagesRecord association. The managed system element that is associated with  
 238 the log, uses the log, or populates the log is referenced through the CIM\_UseOfLog association.

239 The CIM\_LogEntry class contains properties describing the information about individual records, such as  
 240 message text and timestamp. CIM\_RecordLog describes the general properties of the log, such as its  
 241 maximal length and state.

## 242 7 Implementation

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

### 246 7.1 Representing Logs

247 Each log in a managed system shall be represented by a single instance of CIM\_RecordLog. Each entry  
248 in the log shall be represented by a single instance of CIM\_LogEntry. The entries of the log, which are  
249 represented by the instances of CIM\_LogEntry, shall be associated through the instance of  
250 CIM\_LogManagesRecord to the instance of CIM\_RecordLog.

#### 251 7.1.1 CIM\_LogEntry.LogInstanceID

252 The CIM\_LogEntry.LogInstanceID shall have the same value as the InstanceID property of the instance  
253 of CIM\_RecordLog that is associated with the instance CIM\_LogEntry through an instance of  
254 CIM\_LogManagesRecord.

#### 255 7.1.2 CIM\_LogEntry.LogName

256 The CIM\_LogEntry.LogName shall have the same value as the ElementName property of the instance of  
257 CIM\_RecordLog that is associated with the instance CIM\_LogEntry through an instance of  
258 CIM\_LogManagesRecord.

#### 259 7.1.3 CIM\_LogEntry Data

260 The CIM\_LogEntry data information shall be implemented using one or more of the following formats:

- 261 • Record Data Format
- 262 • Standard Message Format

##### 263 7.1.3.1 When the CIM\_LogEntry Implements the Record Data Format

264 When the Record Data Format is supported for a log entry the  
265 CIM\_RecordLogCapabilities.SupportedRecordTypes property shall include the value 2 (Record Data) and  
266 it shall be implemented as described in the following sections.

###### 267 7.1.3.1.1 CIM\_LogEntry.RecordData

268 The CIM\_LogEntry.RecordData property shall be implemented. The RecordData property shall be non-  
269 NULL and shall contain information about the log entry.

###### 270 7.1.3.1.2 CIM\_LogEntry.RecordFormat

271 The CIM\_LogEntry.RecordFormat property shall be implemented if the RecordData property is  
272 implemented (non-NULL). The RecordFormat property shall specify the format of the data provided by the  
273 RecordData property.

##### 274 7.1.3.2 When the CIM\_LogEntry Does Not Implement the Record Data Format

275 If the Record Data Format is not supported for the log entry, the RecordData and RecordFormat  
276 properties shall be NULL.

### 277 7.1.3.3 When the CIM\_LogEntry Implements the Standard Message Format

278 When the Standard Message Format is supported for a log entry, the SupportedRecordTypes property  
279 shall include the value 3 (Standard Messages) and it shall be implemented as described in the following  
280 sections. The implementation will need to implement at least one of the following properties: Message  
281 and/or MessageArguments.

#### 282 7.1.3.3.1 CIM\_LogEntry.MessageID

283 The CIM\_LogEntry.MessageID property shall be specified. The value of the MessageID property conveys  
284 a message from a message registry and shall be set to the concatenation of the PREFIX and  
285 SEQUENCE\_NUMBER attribute values, as specified in the message registry (that is, no further padding  
286 or adjustment of these values takes place).

#### 287 7.1.3.3.2 CIM\_LogEntry.Message

288 The CIM\_LogEntry.Message property may be implemented.

289 If the MessageArguments property is NULL (not implemented), then the Message property shall be  
290 implemented and the value of the Message property shall be non-NULL.

291 If the MessageArguments property is non-NULL (implemented) and the Message property is not  
292 implemented, then the value of the Message property shall be NULL.

#### 293 7.1.3.3.3 CIM\_LogEntry.MessageArguments

294 The CIM\_LogEntry.MessageArguments property may be implemented.

295 If the Message property is NULL (not implemented), then the MessageArguments property shall be  
296 implemented and the value of the MessageArguments property shall be non-NULL.

297 If the Message property is non-NULL (implemented) and the MessageArguments property is not  
298 implemented, then the value of the MessageArguments shall be NULL.

299 If the MessageArguments property is implemented but the message does not contain any dynamic  
300 elements, then the MessageArguments property shall be implemented as an empty array. Otherwise the  
301 MessageArguments property shall contain the values for all of the dynamic elements for the message.

#### 302 7.1.3.3.4 CIM\_LogEntry.PerceivedSeverity

303 The CIM\_LogEntry.PerceivedSeverity property shall be specified if the implementation is supporting  
304 Standard Messages.

#### 305 7.1.3.3.5 CIM\_LogEntry.OwningEntity

306 The CIM\_LogEntry.OwningEntity property shall be specified if the implementation is supporting Standard  
307 Messages.

### 308 7.1.3.4 When CIM\_LogEntry does not implement the Standard Message Format

309 When Standard Message Format is not supported for a log entry the associated properties shall be  
310 NULL. These properties are:

- 311 • MessageID
- 312 • Message
- 313 • MessageArguments
- 314 • PerceivedSeverity

- 315 • OwningEntity

## 316 7.2 CIM\_RecordLogCapabilities

317 There shall be an instance of CIM\_RecordLogCapabilities that specifies the capabilities of the associated  
318 record log. The instance of CIM\_RecordLogCapabilities shall be associated with the CIM\_RecordLog  
319 instance through an instance of CIM\_ElementCapabilities and be used for advertising the capabilities of  
320 the CIM\_RecordLog instance.

321 There shall be at most one instance of CIM\_RecordLogCapabilities associated with a given instance of  
322 CIM\_RecordLog.

### 323 7.2.1 CIM\_RecordLogCapabilities.SupportedRecordTypes

324 The CIM\_RecordLogCapabilities.SupportedRecordTypes property shall indicate which formats are  
325 supported by the implementation. A Record Log shall support at least one of the following formats:  
326 Record Data Format and/or Standard Message Format. . If both the Record Data Format and Standard  
327 Message Format are supported the client will need to check the individual property values to determine  
328 which format is available in each individual log entry.

329 At least one format shall be implemented and specified in the SupportedRecordTypes property of the  
330 CIM\_RecordLogCapabilities instance. The SupportedRecordTypes property shall be non-NULL and shall  
331 not be an empty array.

#### 332 7.2.1.1 SupportedRecordTypes with Only Record Data Format

333 If the SupportedRecordTypes property contains only 2 (Record Data), then the RecordData and  
334 RecordFormat properties of all instances of CIM\_LogEntry associated with the instance of  
335 CIM\_RecordLog shall be Non-NULL.

#### 336 7.2.1.2 SupportedRecordTypes with Only Standard Message Format

337 If the SupportedRecordTypes property contains only 3 (Standard Messages), then the PerceivedSeverity,  
338 OwningEntity, MessageID properties and at least one of the properties Message and MessageArguments  
339 of all instances of CIM\_LogEntry associated with the instance of CIM\_RecordLog shall be Non-NULL.

#### 340 7.2.1.3 SupportedRecordTypes with Record Data and Standard Message Formats

341 If the SupportedRecordTypes property contains only 2 (Record Data) and 3 (Standard Messages) the  
342 following conditions shall be implemented.

343 If the MessageID property of an instance of CIM\_LogEntry associated with the instance of  
344 CIM\_RecordLog is NULL, then the RecordData and RecordFormat properties of the instance of  
345 CIM\_LogEntry shall be non-NULL.

346 If the RecordData and RecordFormat properties of an instance of CIM\_LogEntry associated with the  
347 instance of CIM\_RecordLog are NULL, then the PerceivedSeverity, OwningEntity, MessageID properties  
348 and at least one of the properties Message and MessageArguments of the instance of CIM\_LogEntry  
349 shall be non-NULL.

### 350 7.2.2 CIM\_RecordLogCapabilities.RequestedStatesSupported

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

### 356 7.2.3 CIM\_RecordLogCapabilities.ElementNameEditSupported

357 The CIM\_RecordLogCapabilities.ElementNameEditSupported property shall have a value of TRUE when  
358 the implementation supports client modification of the CIM\_RecordLog.ElementName property.

### 359 7.2.4 CIM\_RecordLogCapabilities.MaxElementNameLen

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

## 362 7.3 Log State Management (Optional)

363 Log state management consists of the CIM\_RecordLog.RequestStateChange() method being supported  
364 (see 8.2) and the value of the CIM\_RecordLog.RequestedState not matching 12 (Not Applicable).

### 365 7.3.1 Log State Management Support

366 When a CIM\_RecordLogCapabilities.RequestedStatesSupported property is an empty array, log state  
367 management shall not be supported.

368 When a CIM\_RecordLogCapabilities instance is associated with the CIM\_RecordLog instance and the  
369 value of the CIM\_RecordLogCapabilities.RequestedStatesSupported property is not an empty array, log  
370 state management shall be supported.

## 371 7.4 CIM\_RecordLog.RequestedState

372 The CIM\_RecordLog.RequestedState property shall have a value of 12 (Not Applicable), 5 (No Change),  
373 or a value contained in the CIM\_RecordLogCapabilities.RequestedStatesSupported property array of the  
374 associated CIM\_RecordLogCapabilities instance (see 7.2.2).

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

### 381 7.4.1 RequestedState — 12 (Not Applicable) Value

382 When log state management is not supported, the value of the CIM\_RecordLog.RequestedState property  
383 shall be 12 (Not Applicable).

### 384 7.4.2 RequestedState — 5 (No Change) Value

385 When log state management is supported, the initial value of the CIM\_RecordLog.RequestedState  
386 property shall be 5 (No Change).

## 387 7.5 Representing Log State

388 The log's state shall be represented by two properties: CIM\_RecordLog.EnabledState (see 7.5.1) and  
389 CIM\_RecordLog.LogState (see 7.5.2).

### 390 7.5.1 CIM\_RecordLog.EnabledState

391 Table 2 describes the mapping between the values of the CIM\_RecordLog.EnabledState property and the  
392 corresponding description of the state of the log. The CIM\_RecordLog.EnabledState property shall match  
393 the values that are specified in Table 2. When the RequestStateChange() method executes but does not

394 complete successfully and the log is in an indeterminate state, the CIM\_RecordLog.EnabledState  
 395 property shall have a value of 5 (Not Applicable). The value of this property may also change as a result  
 396 of a non-CIM implementation's change to the log's enabled state.

397 **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 7.5.1.1.

398 **7.5.1.1 CIM\_RecordLog.EnabledState — 6 (Enabled but Offline) Value**

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

402 When the CIM\_RecordLog.OverwritePolicy property has a value of 2 (Wraps When Full), the  
 403 CIM\_RecordLog.EnabledState property shall not have a value of 6 (Enabled but Offline).

404 **7.5.2 CIM\_RecordLog.LogState**

405 The CIM\_RecordLog.LogState property is used to describe a more granular state of the log than that of  
 406 the CIM\_RecordLog.EnabledState property. Table 3 describes the mapping between the values of the  
 407 CIM\_RecordLog.LogState property and the corresponding description of the granular state of the log. The  
 408 CIM\_RecordLog.LogState property shall match the values that are specified in Table 3. Additionally,  
 409 Table 3 describes the mapping between the LogState property and the EnabledState property. When the  
 410 CIM\_RecordLog.LogState property has a value that matches the value in the "LogState Value" column in  
 411 Table 3, the CIM\_RecordLog.EnabledState property shall have a value that matches the value in the  
 412 "EnabledState Value" column for that row.

413 **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 7.5.2.1.
4	Not Applicable	Any value in Table 2	LogState property is not used to describe more granular state of the log, and EnabledState property shall be used only to represent the log state.

414 **7.5.2.1 CIM\_RecordLog.LogState — 3 (Erasing) Value**

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

419 **7.6 CIM\_UseOfLog**

420 At least one instance of the CIM\_UseOfLog association shall reference an instance of CIM\_RecordLog  
 421 and an instance of the subclass of CIM\_ManagedSystemElement.

422 **7.7 CIM\_HostedDependency**

423 An instance of the CIM\_HostedDependency association may reference an instance of CIM\_RecordLog  
 424 and an instance of the subclass of CIM\_ManagedSystemElement.

425 **7.8 CIM\_RecordLog.OverwritePolicy Property**

426 The CIM\_RecordLog.OverwritePolicy property indicates the behavior of the log when it has reached the  
 427 maximum capacity of its entries. The CIM\_RecordLog.OverwritePolicy property also affects the  
 428 CIM\_RecordLog.EnabledState property, as described in 7.5.1.1.

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

431 When the new log entries overwrite the old log entries, the CIM\_RecordLog.OverwritePolicy property has  
 432 a value of 2 (Wraps When Full). When the new log entries never overwrite the old log entries, the  
 433 CIM\_RecordLog.OverwritePolicy property has a value of 7 (Never Overwrites).

434 **8 Methods**

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

437 **8.1 CIM\_RecordLog.ClearLog( )**

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

441 CIM\_RecordLog.ClearLog() return code values shall be as specified in Table 4.

442 No parameters or standard messages are defined for the CIM\_RecordLog.ClearLog() method.

443 **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

## 444 8.2 CIM\_RecordLog.RequestStateChange()

445 Invocation of the CIM\_RecordLog.RequestStateChange() method shall attempt to change the element's  
446 state to the value that is specified in the RequestedState parameter.

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

450 When log state management is supported, the RequestStateChange() method shall be implemented and  
451 shall not return a value of 1 (Not Supported) (see 7.3.1).

452 When the RequestedState parameter is set to 2 (Enabled) but the CIM\_RecordLog.EnabledState  
453 property has a value of 6 (Enabled but Offline), the RequestStateChange() method invocation shall return  
454 2 (Error Occurred).

455 Invoking the CIM\_RecordLog.RequestStateChange() method multiple times could result in earlier  
456 requests being overwritten or lost.

457 No standard messages are defined for this method.

458 **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

459 **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

## 460 8.3 Profile Conventions for Operations

461 This profile specification defines operations in terms of [DSP0200](#).

462 For each profile class (including associations), the implementation requirements for operations, including  
463 those in the following default list, are specified in class-specific subclauses of this clause.



464 The default list of operations is as follows:

- 465 • Associators( )
- 466 • AssociatorNames( )
- 467 • EnumerateInstances( )
- 468 • EnumerateInstanceNames( )
- 469 • GetInstance( )
- 470 • References( )
- 471 • ReferenceNames( )

472 **8.4 CIM\_ElementCapabilities**

473 Table 7 lists implementation requirements for operations. If implemented, these operations shall be  
 474 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 7, all operations in  
 475 the default list in 8.3 shall be implemented as defined in [DSP0200](#).

476 NOTE: Related profiles may define additional requirements on operations for the profile class.

477 **Table 7 – Operations: CIM\_ElementCapabilities**

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

478 **8.5 CIM\_RecordLogCapabilities**

479 All operations in the default list in 8.3 shall be implemented as defined in [DSP0200](#).

480 NOTE: Related profiles may define additional requirements on operations for the profile class.

481 **8.6 CIM\_RecordLog**

482 Table 8 lists implementation requirements for operations. If implemented, these operations shall be  
 483 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 8, all operations in  
 484 the default list in 8.3 shall be implemented as defined in [DSP0200](#).

485 NOTE: Related profiles may define additional requirements on operations for the profile class.

486 **Table 8 – Operations: CIM\_RecordLog**

Operation	Requirement	Messages
ModifyInstance	Optional: See 8.6.1.	None

487 **8.6.1 CIM\_RecordLog — ModifyInstance**

488 This section details the requirements for the ModifyInstance operation applied to an instance of  
 489 CIM\_RecordLog. The ModifyInstance operation may be supported.

490 The ModifyInstance operation shall be supported and CIM\_RecordLog.ElementName shall be modifiable  
 491 when the ElementNameEditSupported property of the CIM\_EnabledLogicalElementCapabilities instance  
 492 that is associated with the CIM\_RecordLog instance has a value of TRUE. See 8.6.2.

### 493 8.6.2 CIM\_RecordLog.ElementName

494 When the ElementNameEditSupported property of the CIM\_EnabledLogicalElementCapabilities instance  
 495 that is associated with the CIM\_RecordLog instance has a value of TRUE, the implementation shall allow  
 496 the ModifyInstance operation to change the value of the ElementName property of the CIM\_RecordLog  
 497 instance. The ModifyInstance operation shall enforce the length restriction specified in the  
 498 MaxElementNameLen property of the CIM\_EnabledLogicalElementCapabilities instance.

499 When the associated CIM\_EnabledLogicalElementCapabilities instance does not exist or the  
 500 ElementNameEditSupported property of the associated CIM\_EnabledLogicalElementCapabilities  
 501 instance has a value of FALSE, the implementation shall not allow the ModifyInstance operation to  
 502 change the value of the ElementName property of the CIM\_RecordLog instance.

## 503 8.7 CIM\_LogEntry

504 Table 9 lists implementation requirements for operations. If implemented, these operations shall be  
 505 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 9, all operations in  
 506 the default list in 8.3 shall be implemented as defined in [DSP0200](#).

507 NOTE: Related profiles may define additional requirements on operations for the profile class.

508 **Table 9 – Operations: CIM\_LogEntry**

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

### 509 8.7.1 CIM\_LogEntry DeleteInstance

510 CIM\_LogEntry DeleteInstance operation shall be optional. The implementation shall also remove any  
 511 association instances that reference the instance of CIM\_LogEntry, including the instance of  
 512 CIM\_LogManagesRecord.

## 513 8.8 CIM\_UseOfLog

514 Table 10 lists implementation requirements for operations. If implemented, these operations shall be  
 515 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 10, all operations  
 516 in the default list in 8.3 shall be implemented as defined in [DSP0200](#).

517 NOTE: Related profiles may define additional requirements on operations for the profile class.

518 **Table 10 – Operations: CIM\_UseOfLog**

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

519 **8.9 CIM\_LogManagesRecord**

520 Table 11 lists implementation requirements for operations. If implemented, these operations shall be  
 521 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 11, all operations  
 522 in the default list in 8.3 shall be implemented as defined in [DSP0200](#).

523 NOTE: Related profiles may define additional requirements on operations for the profile class.

524 **Table 11 – Operations: CIM\_LogManagesRecord**

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

525 **8.10 CIM\_HostedDependency**

526 Table 12 lists implementation requirements for operations. If implemented, these operations shall be  
 527 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 12, all operations  
 528 in the default list in 8.3 shall be implemented as defined in [DSP0200](#).

529 NOTE: Related profiles may define additional requirements on operations for the profile class.

530 **Table 12 – Operations: CIM\_HostedDependency**

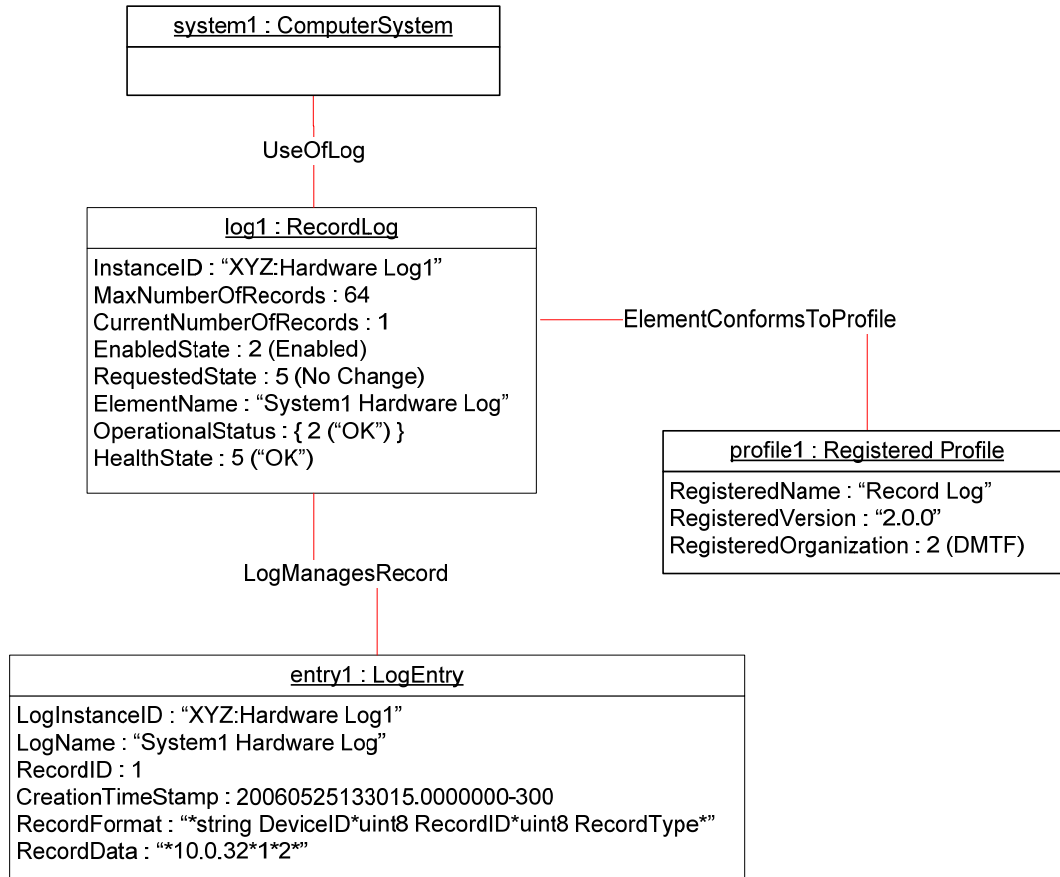
Operation	Requirement	Messages
Associators	Unspecified	None
AssociatorNames	Unspecified	None
References	Unspecified	None
ReferenceNames	Unspecified	None

531 **9 Use Cases**

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

533 **9.1 Object Diagrams**

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

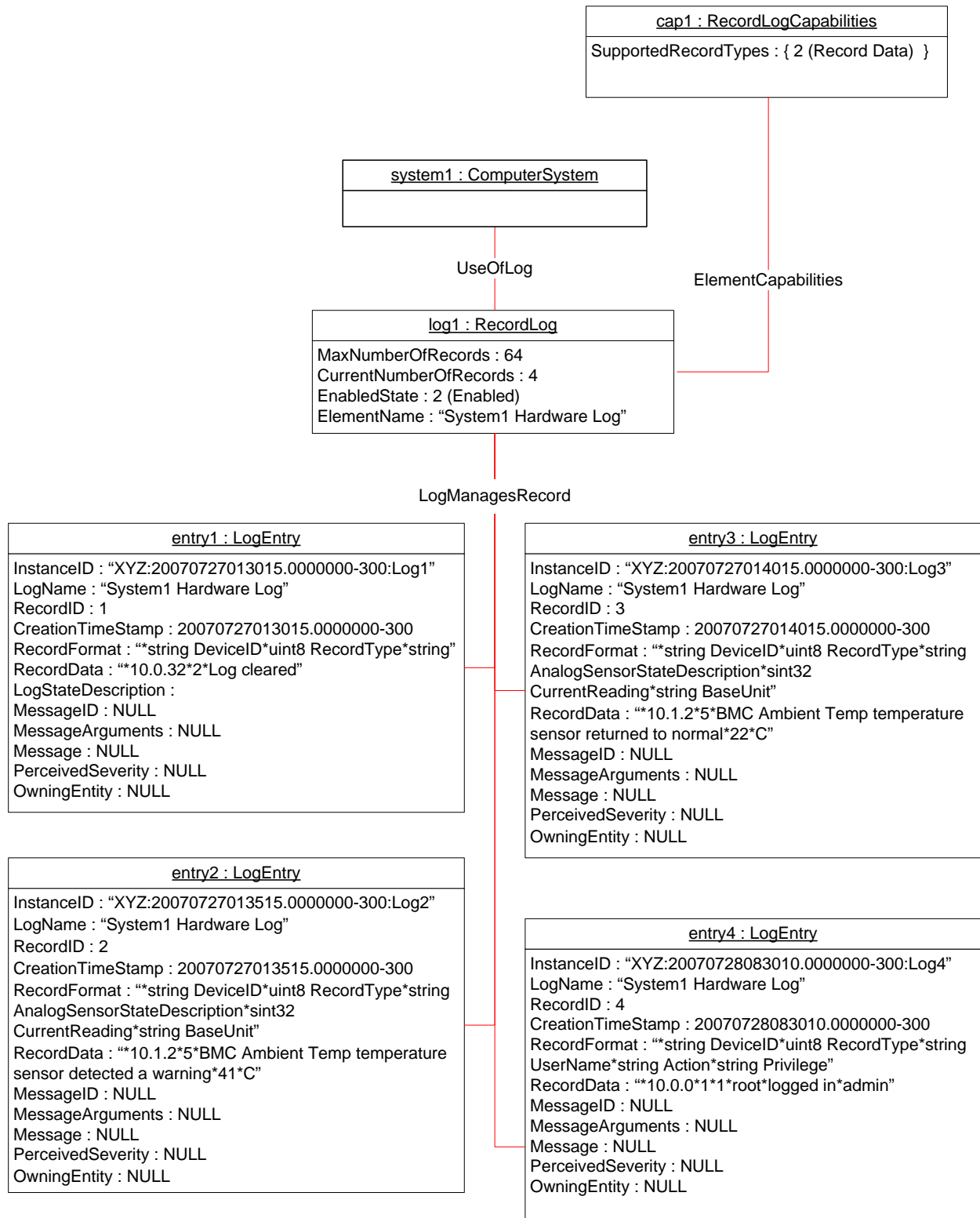


538

539

**Figure 2 – RecordLog Instance**

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

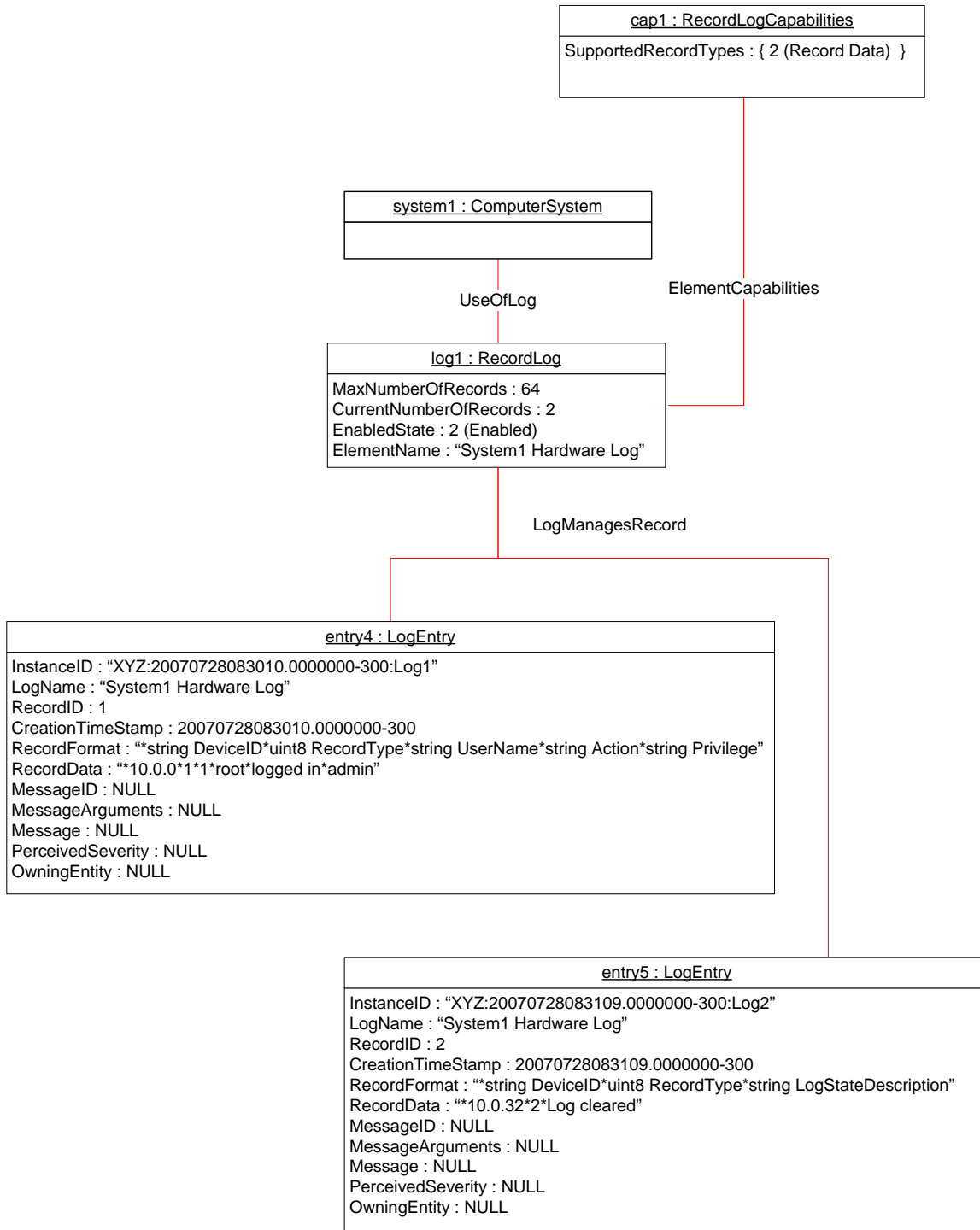


547

548

Figure 3 – RecordLog Instance Before the Log Is Cleared

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



553

554

Figure 4 – RecordLog Instance after the Log Is Cleared

## 555 9.2 Identify the Log by the Name

556 To select a log by its name, a client can select the CIM\_RecordLog instance in which the ElementName  
557 property corresponds to the desired name.

## 558 9.3 Browse the Records of the Log

559 To browse log records, a client can iterate through all the instances of CIM\_LogEntry that are associated  
560 through the CIM\_LogManagesRecord association to the given instance of CIM\_RecordLog and sort them  
561 based on the RecordID.

## 562 9.4 Sort the Log Records Based on the Time Stamp of the Log Entry

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

- 564 1) Iterate through all the instances of CIM\_LogEntry that are associated through the  
565 CIM\_LogManagesRecord association to the given instance of CIM\_RecordLog that represents  
566 the log record.
- 567 2) Sort the instances of CIM\_LogEntry based on the CreationTimeStamp property value in LIFO  
568 order.

## 569 9.5 Delete a Log Entry

570 A client can delete a log entry as follows:

- 571 1) Select the instance of CIM\_LogEntry that represents the desired log entry to be deleted.
- 572 2) Execute DeleteInstance operation on the selected instance of CIM\_LogEntry.

573 Upon successful execution, the instance of CIM\_LogEntry and the instance of CIM\_LogManagesRecord  
574 that associates the log entry to the instance of CIM\_RecordLog are deleted.

## 575 9.6 Clear the Log

576 To clear the log, a client can execute the ClearLog() method for the given instance of CIM\_RecordLog.

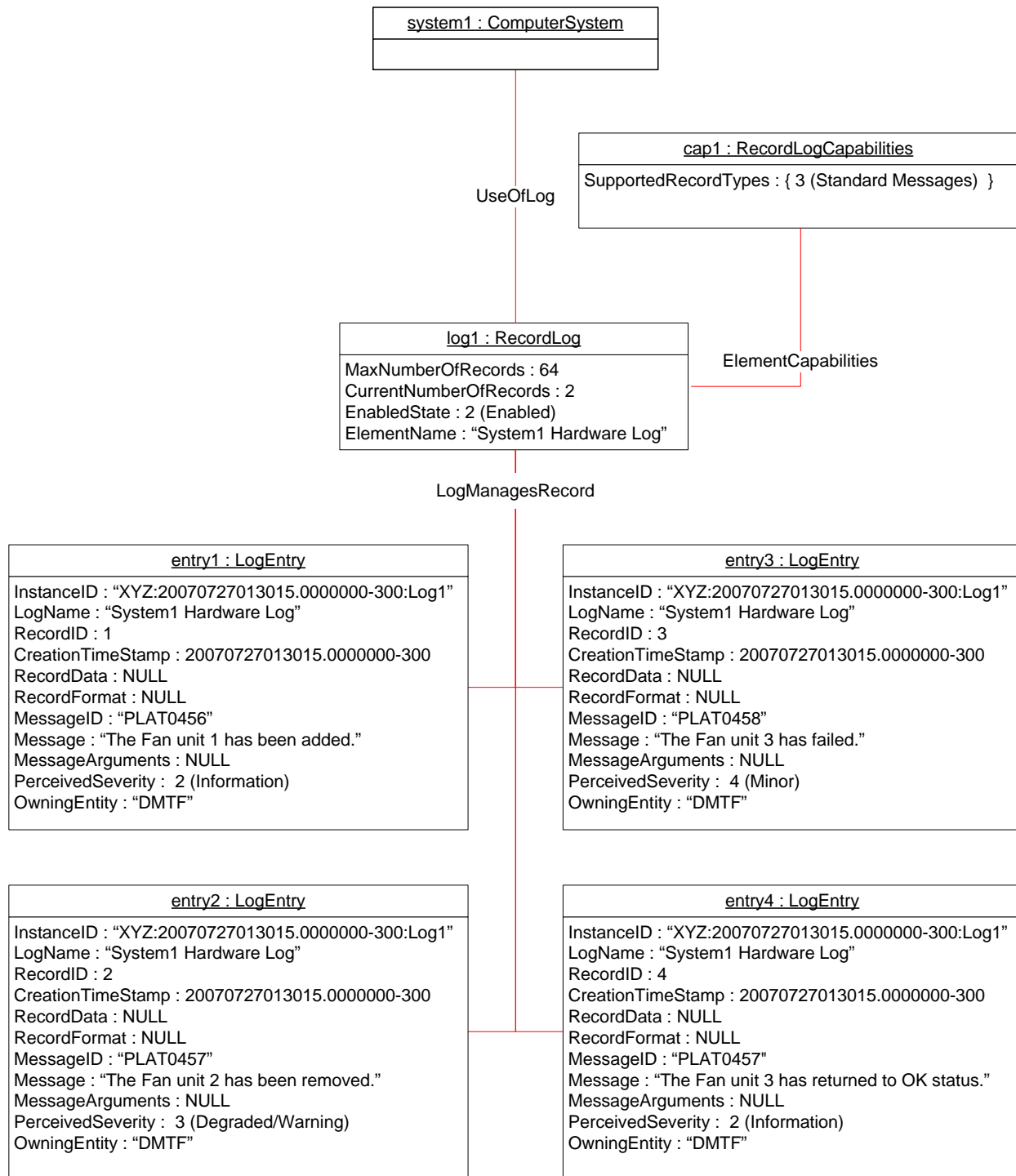
## 577 9.7 Determine Which Record Types Are Supported

578 A client can determine which record types are supported as follows:

- 579 1) From the instance of CIM\_RecordLog use the CIM\_ElementCapabilities association to obtain  
580 the associated instance of CIM\_RecordLogCapabilities.
- 581 2) The SupportedRecordTypes property of the CIM\_RecordLogCapabilities instance indicates  
582 which record types are supported by the implementation.

## 583 9.8 RecordLog Instance for Standard Messages

584 Figure 5 shows four messages using the standard message record log type. For these record log entries  
585 that support standard messages only the Message property has been supplied but the message  
586 arguments are not; therefore, the MessageArguments property is NULL.



587

588

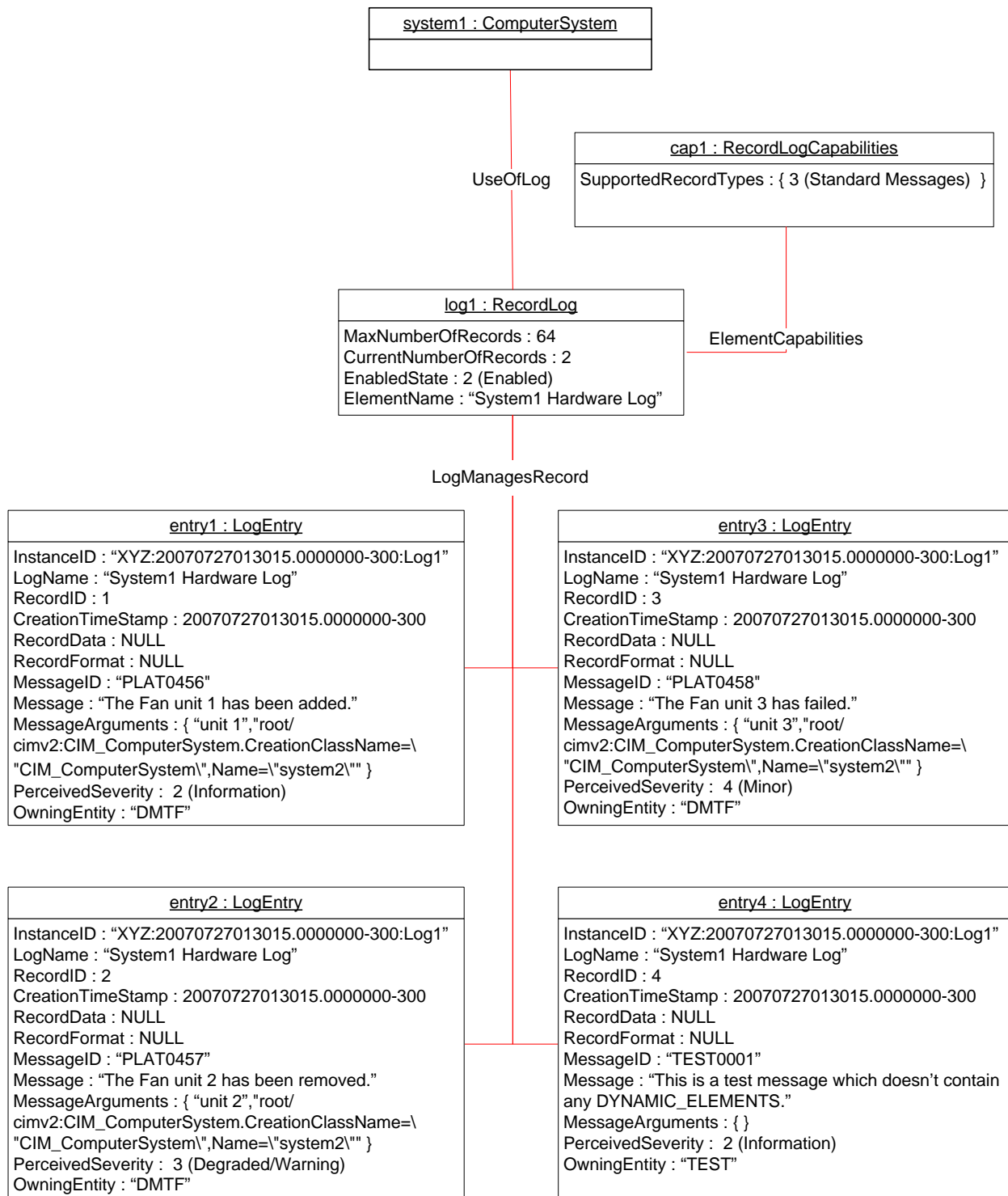
**Figure 5 – RecordLog Instance for Standard Messages**



589 **9.9 RecordLog Instance for Standard Messages with MessageArguments**  
590 **Supported**

591 Figure 6 shows four messages using the standard message record log type. For these record log entries  
592 that support standard messages both the Message and MessageArguments properties are supported.  
593 entry4 shows a potential message that does not contain any DYNAMIC\_ELEMENTS. For this case, the  
594 MessageArguments property is an empty array.

595



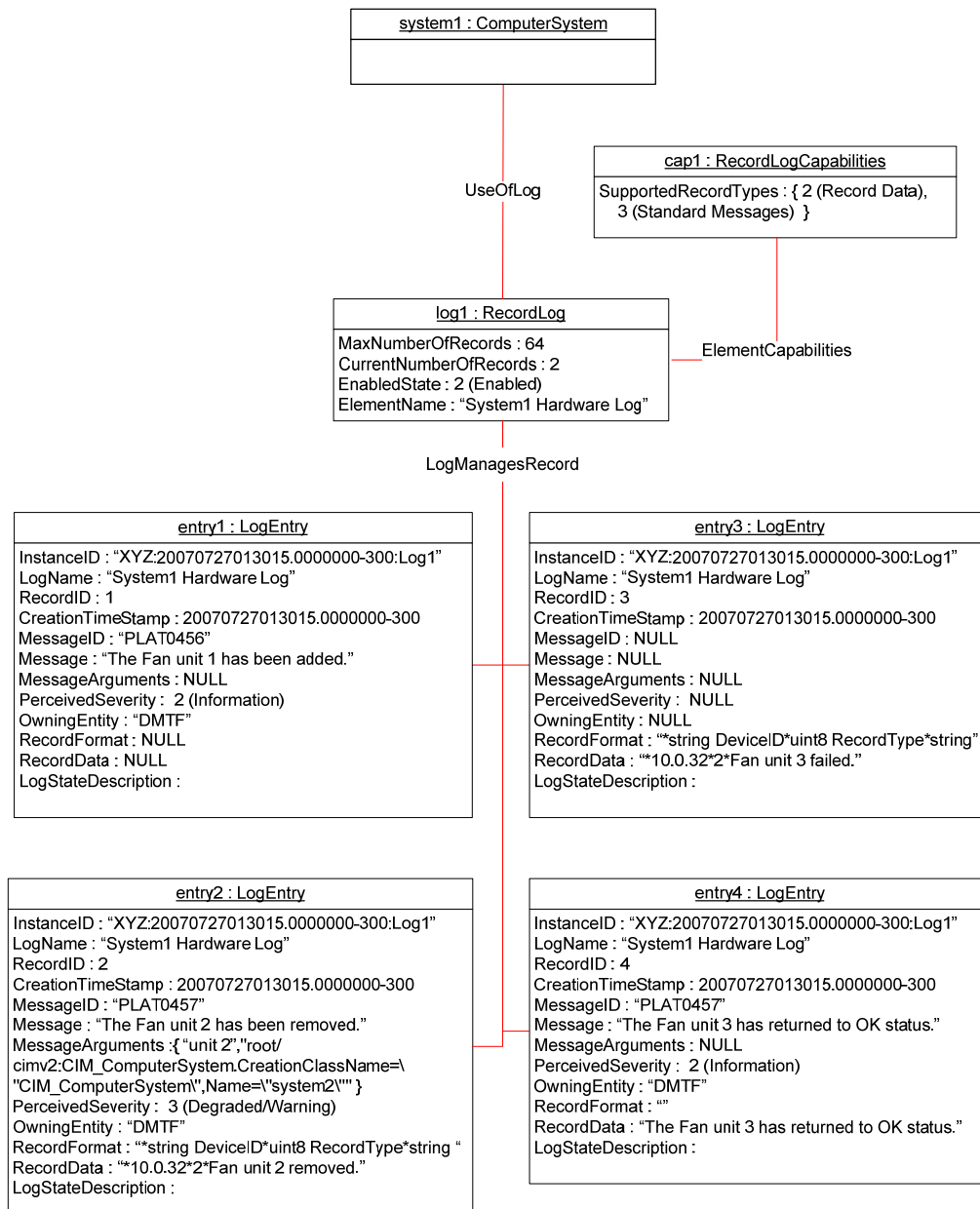
596

597

Figure 6 – RecordLog with Standard Message and MessageArguments

598 **9.10 RecordLog Instance for Record Data and Standard Messages**

599 Figure 7 shows a Record Log that supports both the Record Data and Standard Message formats. In  
 600 addition, for the standard messages both the Message and MessageArguments properties are supplied.  
 601 The entry1 log entry contains only the Standard Message Format with the Message property, but not the  
 602 MessageArguments property. The entry2 log entry contains both the Record Data and Standard Message  
 603 Formats. The entry3 log entry contains only the Record Data Format. And the entry4 log entry contains  
 604 both the Record Data and Standard Message Formats. The entry4 log entry shows properties for the  
 605 Record Data Format where the RecordFormat property is an empty string, which indicates that the  
 606 RecordData is a free-format string.



607

608

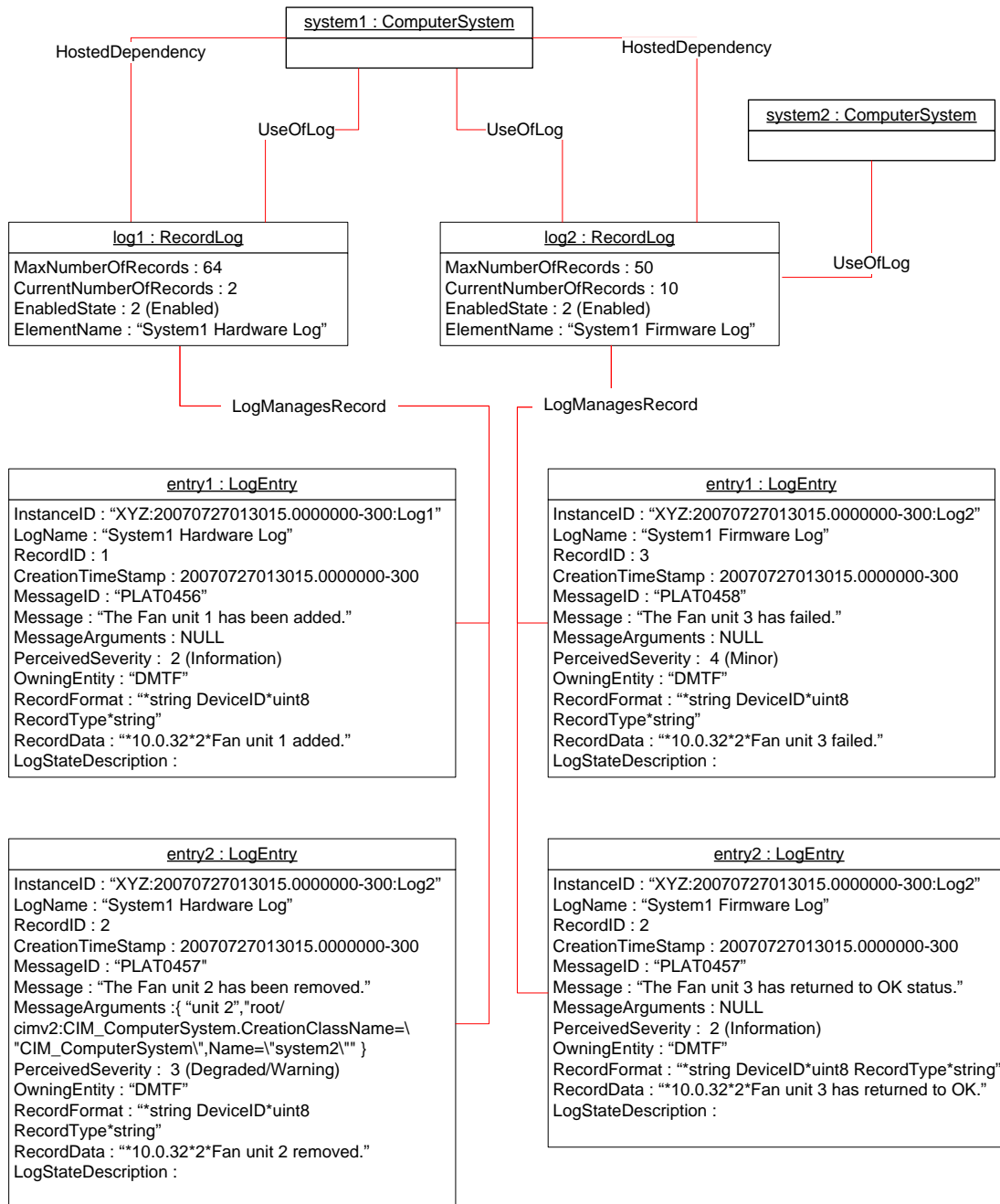
**Figure 7 – RecordLog Instances for Both Record Types**

609 **9.11 List All Logs Hosted on This System**

610 Figure 8 shows two record logs. Both of these record logs are used by and hosted on system1. system2  
 611 uses log2 but does not host any record logs.

612 A client can list all logs hosted on system1 as follows:

- 613 1) From the CIM\_ComputerSystem instance for system1 enumerate all CIM\_HostedDependency  
 614 association instances where CIM\_RecordLog is the result class.

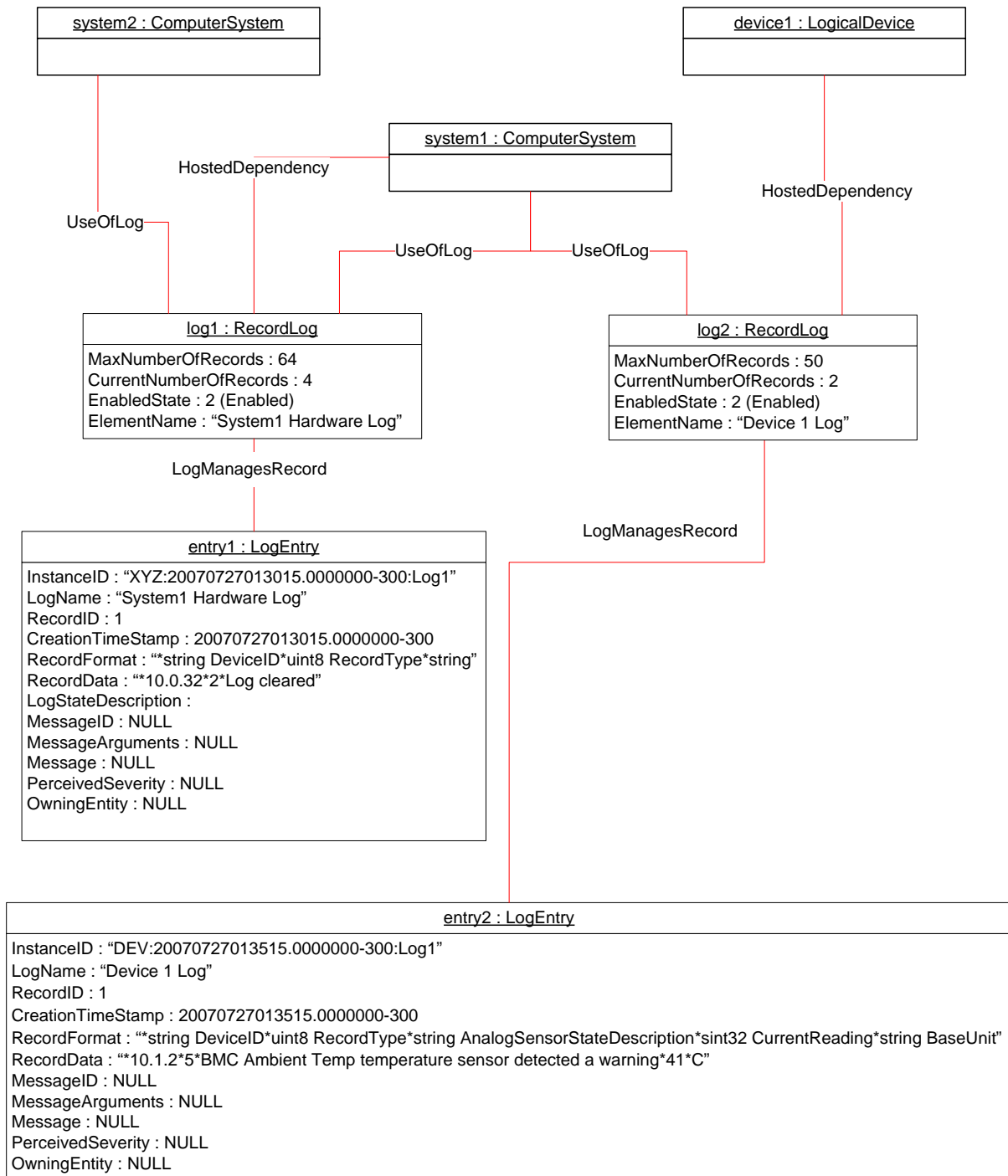


615

616

**Figure 8 – Record Log Hosted on system1**

617 Figure 9 shows two record logs (log1 and log2). log1 is hosted on system1 and used by system1 and  
 618 system2. log2 is hosted on device1 and is used by system1.



619

620

**Figure 9 – Record Logs Hosted on system1 and device1**

621 **10 CIM Elements**

622 Table 13 shows the instances of CIM Elements for this profile. Instances of the CIM Elements shall be  
 623 implemented as described in Table 13. Sections 7 (“Implementation”) and 8 (“Methods”) may impose  
 624 additional requirements on these elements.

625 **Table 13 – CIM Elements: Record Log Profile**

Element Name	Requirement	Description
<b>Classes</b>		
CIM_ElementCapabilities	Optional	See 10.1.
CIM_RecordLogCapabilities	Mandatory	See 7.2 and 10.2.
CIM_LogManagesRecord	Optional	See 10.3.
CIM_LogEntry	Optional	See 10.4.
CIM_RecordLog	Mandatory	See 10.5.
CIM_RegisteredProfile	Mandatory	See 10.6.
CIM_UseOfLog	Mandatory	See 10.7.
CIM_HostedDependency	Optional	See 10.8.
<b>Indications</b>		
None defined in this profile		

626 **10.1 CIM\_ElementCapabilities**

627 CIM\_ElementCapabilities associates an instance of CIM\_RecordLog with an instance of  
 628 CIM\_EnabledLogicalElementCapabilities that describes the capabilities of CIM\_RecordLog. Table 14  
 629 provides information about the properties of CIM\_ElementCapabilities.

630 **Table 14 – Class: CIM\_ElementCapabilities**

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

631 **10.2 CIM\_RecordLogCapabilities**

632 CIM\_RecordLogCapabilities represents the capabilities of the log. Table 15 provides information about  
633 the properties of CIM\_RecordLogCapabilities.

634 **Table 15 – Class: CIM\_RecordLogCapabilities**

Elements	Requirement	Notes
InstanceID	Mandatory	<b>Key</b>
RequestedStatesSupported	Mandatory	See 7.2.2.
ElementNameEditSupported	Mandatory	See 7.2.3.
MaxElementNameLen	Conditional	See 7.2.4.
SupportedRecordTypes	Mandatory	See 7.2.1.

635 **10.3 CIM\_LogManagesRecord**

636 CIM\_LogManagesRecord associates the CIM\_RecordLog instance, which represents the log, with an  
637 instance of CIM\_LogEntry, which represents an entry within the log. Table 16 provides information about  
638 the properties of CIM\_LogManagesRecord.

639 **Table 16 – Class: CIM\_LogManagesRecord**

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

640 **10.4 CIM\_LogEntry**

641 CIM\_LogEntry represents the log entry within the log in the managed system. Table 17 provides  
642 information about the properties of CIM\_LogEntry.

643 **Table 17 – Class: CIM\_LogEntry**

Elements	Requirement	Notes
InstanceID	Mandatory	<b>Key</b>
LogInstanceID	Optional	See 7.1.1.
LogName	Optional	See 7.1.2.
RecordID	Mandatory	None
CreationTimeStamp	Mandatory	None
RecordData	Conditional	See 7.1.3.1.1 and 7.2.1.
RecordFormat	Conditional	See 7.1.3.1.2 and 7.2.1.
ElementName	Mandatory	The property shall match pattern “.*”.

Elements	Requirement	Notes
PerceivedSeverity	Conditional	See 7.1.3.3.4 and 7.2.1.
OwningEntity	Conditional	See 7.1.3.3.5 and 7.2.1.
MessageID	Conditional	See 7.1.3.3.1 and 7.2.1.
Message	Conditional	See 7.1.3.3.2 and 7.2.1.
MessageArguments	Conditional	See 7.1.3.3.2 and 7.2.1.

## 644 10.5 CIM\_RecordLog

645 CIM\_RecordLog represents the log in the managed system. Table 18 provides information about the  
646 properties of CIM\_RecordLog.

647 **Table 18 – Class: CIM\_RecordLog**

Elements	Requirement	Notes
InstanceID	Mandatory	<b>Key</b>
MaxNumberOfRecords	Mandatory	A value of 0 shall mean “Unknown” or “Not Applicable”.
LogState	Mandatory	See 7.5.2.
OverwritePolicy	Mandatory	See 7.8.
RequestedState	Mandatory	See 7.4.
EnabledState	Mandatory	See 7.5.1.
OperationalStatus	Mandatory	None
HealthState	Mandatory	None
ElementName	Mandatory	The property shall match pattern “.*”.
CurrentNumberOfRecords	Optional	None

## 648 10.6 CIM\_RegisteredProfile

649 CIM\_RegisteredProfile identifies the *Record Log Profile* in order for a client to determine the conformance  
650 with the profile. The CIM\_RegisteredProfile class is defined by the [Profile Registration Profile](#). With the  
651 exception of the mandatory values specified for the properties in Table 19, the behavior of the  
652 RegisteredProfile instance is per the [Profile Registration Profile](#).

653 **Table 19 – 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 “2.0.0”.
RegisteredOrganization	Mandatory	This property shall have a value of 2 (DMTF).

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



657 **10.7 CIM\_UseOfLog**

658 CIM\_UseOfLog associates CIM\_RecordLog, which represents the log, with a subclass of  
 659 CIM\_ManagedSystemElement, which represents the element that uses or populates the log. Table 20  
 660 provides information about the properties of CIM\_UseOfLog.

661 **Table 20 – Class: CIM\_UseOfLog**

Elements	Requirement	Notes
Antecedent	Mandatory	<b>Key:</b> This property shall reference the CIM_RecordLog instance that represents the log. Cardinality 1..*, indicating one or many references
Dependent	Mandatory	<b>Key:</b> 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

662 **10.8 CIM\_HostedDependency**

663 CIM\_HostedDependency associates CIM\_RecordLog, which represents the log, with a subclass of  
 664 CIM\_ManagedElement, which represents the element that hosts the log. Table 21 provides information  
 665 about the properties of CIM\_HostedDependency.

666 **Table 21 – Class: CIM\_HostedDependency**

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

667  
668  
669  
670

## 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
1.0.1	2008-09-23	Errata 1.0.1
2.0.0	2010-05-20	DMTF Standard, with the following changes: <ul style="list-style-type: none"><li>• Add Standard Message support. Either Record Data or Standard Message Format must be implemented.</li><li>• The new class CIM_RecordLogCapabilities, which subclasses ( and replaces) CIM_EnabledLogicalElementCapabilities, must now be implemented.</li></ul>

671