



1
2
3
4

Document Number: DSP1010

Date: 2016-08-15

Version: 2.1.0

5 **Record Log Profile**

- 6 **Supersedes: 2.0.0**
- 7 **Document Class: Normative**
- 8 **Document Status: Published**
- 9 **Document Language: en-US**

10 Copyright notice

11 Copyright © 2008, 2016 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26 withdrawn or modified after publication, and shall be indemnified and held harmless by any party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 This document's normative language is English. Translation into other languages is permitted.

33

CONTENTS

35	Foreword	5
36	Introduction.....	6
37	1 Scope	7
38	2 Normative references	7
39	3 Terms and definitions	7
40	4 Symbols and abbreviated terms.....	8
41	5 Synopsis	8
42	6 Description	9
43	7 Implementation.....	10
44	7.1 Representing logs	10
45	7.2 CIM_RecordLogCapabilities	12
46	7.3 Log state management (optional)	13
47	7.4 CIM_RecordLog.RequestedState	13
48	7.5 Representing log state	14
49	7.6 CIM_UseOfLog	15
50	7.7 CIM_HostedDependency.....	15
51	7.8 CIM_RecordLog.OverwritePolicy property	15
52	8 Methods.....	15
53	8.1 CIM_RecordLog.ClearLog().....	15
54	8.2 CIM_RecordLog.RequestStateChange().....	16
55	8.3 Profile conventions for operations	17
56	8.4 CIM_ElementCapabilities	17
57	8.5 CIM_RecordLogCapabilities	17
58	8.6 CIM_RecordLog.....	18
59	8.7 CIM_LogEntry	18
60	8.8 CIM_UseOfLog	19
61	8.9 CIM_LogManagesRecord.....	19
62	8.10 CIM_HostedDependency.....	19
63	9 Use cases.....	20
64	9.1 Object diagrams.....	20
65	9.2 Identify the log by the name.....	23
66	9.3 Browse the records of the log	24
67	9.4 Sort the log records based on the time stamp of the log entry	24
68	9.5 Delete a log entry.....	24
69	9.6 Clear the log.....	24
70	9.7 Determine which record types are supported	24
71	9.8 RecordLog instance for standard messages	24
72	9.9 RecordLog Instance for standard messages with MessageArguments supported	25
73	9.10 RecordLog Instance for Record Data and Standard Messages	28
74	9.11 List all logs hosted on this system	29
75	10 CIM Elements	32
76	10.1 CIM_ElementCapabilities	32
77	10.2 CIM_RecordLogCapabilities	33
78	10.3 CIM_LogManagesRecord.....	33
79	10.4 CIM_LogEntry	33
80	10.5 CIM_RecordLog.....	34
81	10.6 CIM_RegisteredProfile.....	34
82	10.7 CIM_UseOfLog	35
83	10.8 CIM_HostedDependency.....	35
84	ANNEX A (informative) Change log	36

85

86 **Figures**

87	Figure 1 – Record Log Profile: Class diagram	9
88	Figure 2 – RecordLog instance	20
89	Figure 3 – RecordLog instance before the log is cleared	21
90	Figure 4 – RecordLog instance after the log is cleared	23
91	Figure 5 – RecordLog instance for standard messages	25
92	Figure 6 – RecordLog with standard message and MessageArguments	27
93	Figure 7 – RecordLog instances for both record types	29
94	Figure 8 – Record log hosted on system1	30
95	Figure 9 – Record logs hosted on system1 and device1	31

96

97 **Tables**

98	Table 1 – Related Profiles	9
99	Table 2 – EnabledState Value Description	14
100	Table 3 – LogState value description and mapping to EnabledState value	14
101	Table 4 – CIM_RecordLog.ClearLog() method: Return code values	16
102	Table 5 – CIM_RecordLog.RequestStateChange() method: Return code values	16
103	Table 6 – CIM_RecordLog.RequestStateChange() method: Parameters	16
104	Table 7 – Operations: CIM_ElementCapabilities	17
105	Table 8 – Operations: CIM_RecordLog	18
106	Table 9 – Operations: CIM_LogEntry	18
107	Table 10 – Operations: CIM_UseOfLog	19
108	Table 11 – Operations: CIM_LogManagesRecord	19
109	Table 12 – Operations: CIM_HostedDependency	19
110	Table 13 – CIM Elements: Record Log Profile	32
111	Table 14 – Class: CIM_ElementCapabilities	32
112	Table 15 – Class: CIM_RecordLogCapabilities	33
113	Table 16 – Class: CIM_LogManagesRecord	33
114	Table 17 – Class: CIM_LogEntry	33
115	Table 18 – Class: CIM_RecordLog	34
116	Table 19 – Class: CIM_RegisteredProfile	34
117	Table 20 – Class: CIM_UseOfLog	35
118	Table 21 – Class: CIM_HostedDependency	35

119

120

Foreword

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

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

124 Acknowledgments

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

126 Editors:

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

131 Contributors:

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

142

143

Introduction

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

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

152 Document conventions

153 Typographical conventions

154 The following typographical conventions are used in this document:

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

156

Record Log Profile

1 Scope

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

2 Normative references

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

166 DMTF DSP0004, *CIM Infrastructure Specification 2.5*,
167 http://www.dmtf.org/standards/published_documents/DSP0004_2.5.pdf

168 DMTF DSP0200, *CIM Operations over HTTP 1.3*,
169 http://www.dmtf.org/standards/published_documents/DSP0200_1.3.pdf

170 DMTF DSP0228, *Message Registry XML Schema 1.1*,
171 http://schemas.dmtf.org/wbem/messageregistry/1/dsp0228_1.1.xsd

172 DMTF DSP1001, *Management Profile Specification Usage Guide 1.0*,
173 http://www.dmtf.org/standards/published_documents/DSP1001_1.0.pdf

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

176 DMTF DSP8007, *Platform Message Registry 1.1*,
177 http://schemas.dmtf.org/wbem/messageregistry/1/dsp8007_1.1.xml

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

3 Terms and definitions

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

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

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

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

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

196 3.1

197 Record Data Format

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

200 3.2

201 referencing profile

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

204 3.3

205 Standard Message Format

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

208 3.4

209 unspecified

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

211 4 Symbols and abbreviated terms

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

213 4.1

214 LIFO

215 Last In, First Out

216 5 Synopsis

217 **Profile Name:** Record Log

218 **Version:** 2.0.0

219 **Organization:** DMTF

220 **CIM Schema Version:** 2.25

221 **Central Class:** CIM_RecordLog

222 **Scoping Class:** CIM_RecordLog

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

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

228

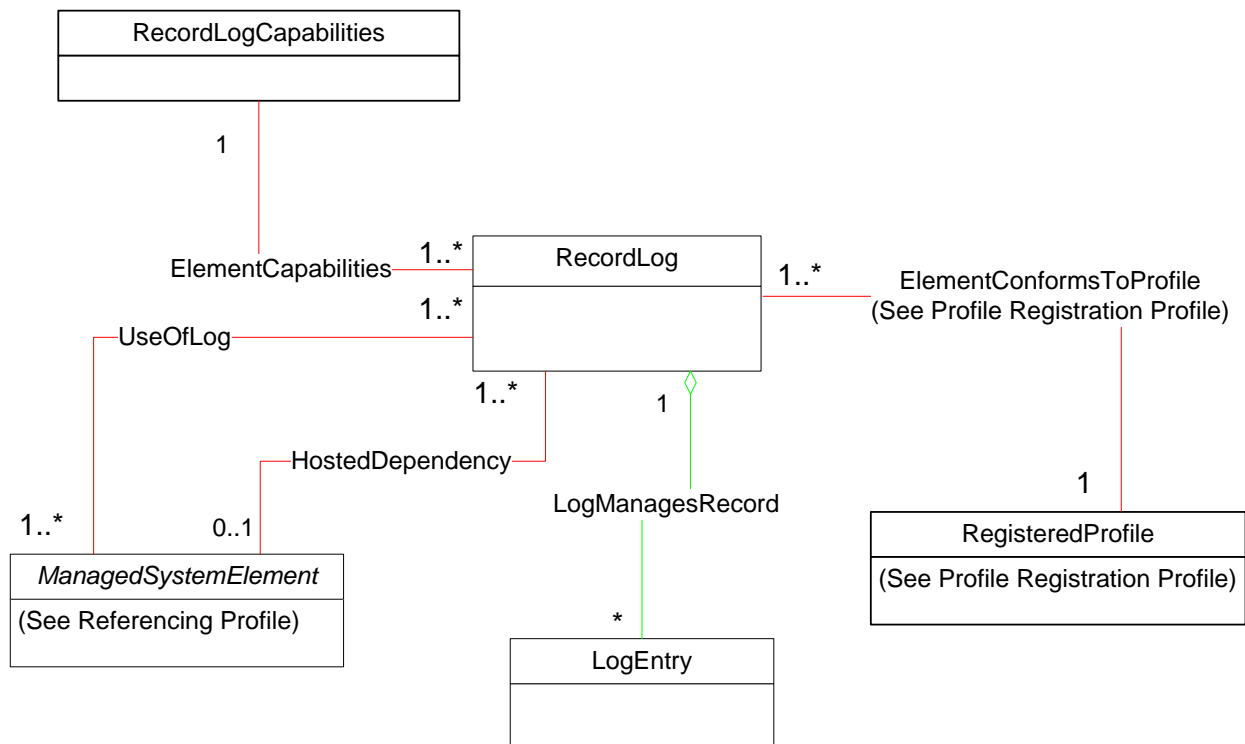
Table 1 – Related Profiles

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

229 **6 Description**

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

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



235

236 **Figure 1 – Record Log Profile: Class diagram**

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

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

244 7 Implementation

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

248 7.1 Representing logs

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

253 7.1.1 CIM_LogEntry.LogInstanceID

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

257 7.1.2 CIM_LogEntry.LogName

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

261 7.1.3 CIM_LogEntry data

262 The CIM_LogEntry data information shall be implemented using one or more of the following formats:

- 263 • Record Data Format
- 264 • Standard Message Format

265 7.1.3.1 When the CIM_LogEntry implements the Record Data Format

266 When the Record Data Format is supported for a log entry the
267 CIM_RecordLogCapabilities.SupportedRecordTypes property shall include the value 2 (Record Data) and
268 it shall be implemented as described in the following clauses.

269 7.1.3.1.1 CIM_LogEntry.RecordData

270 The CIM_LogEntry.RecordData property shall be implemented. The RecordData property shall be non-
271 NULL and shall contain information about the log entry.

272 7.1.3.1.2 CIM_LogEntry.RecordFormat

273 The CIM_LogEntry.RecordFormat property shall be implemented if the RecordData property is
274 implemented (non-NULL). The RecordFormat property shall specify the format of the data provided by the
275 RecordData property.

276 7.1.3.2 When the CIM_LogEntry does not implement the Record Data Format

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

279 **7.1.3.3 When the CIM_LogEntry implements the Standard Message Format**

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

284 **7.1.3.3.1 CIM_LogEntry.MessageID**

285 The CIM_LogEntry.MessageID property shall be specified. The value of the MessageID property conveys
286 a message from a message registry and shall be set to the concatenation of the PREFIX and
287 SEQUENCE_NUMBER attribute values, as specified in the message registry (that is, no further padding
288 or adjustment of these values takes place).

289 **7.1.3.3.2 CIM_LogEntry.Message**

290 The CIM_LogEntry.Message property may be implemented.

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

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

295 **7.1.3.3.3 CIM_LogEntry.MessageArguments**

296 The CIM_LogEntry.MessageArguments property may be implemented.

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

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

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

304 **7.1.3.3.4 CIM_LogEntry.PerceivedSeverity**

305 The CIM_LogEntry.PerceivedSeverity property shall be specified if the implementation is supporting
306 Standard Messages.

307 **7.1.3.3.5 CIM_LogEntry.OwningEntity**

308 The CIM_LogEntry.OwningEntity property shall be specified if the implementation is supporting Standard
309 Messages.

310 **7.1.3.4 When CIM_LogEntry does not implement the Standard Message Format**

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

- 313 • MessageID
- 314 • Message
- 315 • MessageArguments

- 316 • PerceivedSeverity
- 317 • OwningEntity

318 **7.2 CIM_RecordLogCapabilities**

319 There shall be an instance of CIM_RecordLogCapabilities that specifies the capabilities of the associated
320 record log. The instance of CIM_RecordLogCapabilities shall be associated with the CIM_RecordLog
321 instance through an instance of CIM_ElementCapabilities and be used for advertising the capabilities of
322 the CIM_RecordLog instance.

323 There shall be at most one instance of CIM_RecordLogCapabilities associated with a given instance of
324 CIM_RecordLog.

325 **7.2.1 CIM_RecordLogCapabilities.SupportedRecordTypes**

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

331 At least one format shall be implemented and specified in the SupportedRecordTypes property of the
332 CIM_RecordLogCapabilities instance. The SupportedRecordTypes property shall be non-NULL and shall
333 not be an empty array.

334 **7.2.1.1 SupportedRecordTypes with only Record Data Format**

335 If the SupportedRecordTypes property contains only 2 (Record Data), then the RecordData and
336 RecordFormat properties of all instances of CIM_LogEntry associated with the instance of
337 CIM_RecordLog shall be Non-NULL.

338 **7.2.1.2 SupportedRecordTypes with only Standard Message Format**

339 If the SupportedRecordTypes property contains only 3 (Standard Messages), then the PerceivedSeverity,
340 OwningEntity, MessageID properties and at least one of the properties Message and MessageArguments
341 of all instances of CIM_LogEntry associated with the instance of CIM_RecordLog shall be Non-NULL.

342 **7.2.1.3 SupportedRecordTypes with Record Data and Standard Message Formats**

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

345 If the MessageID property of an instance of CIM_LogEntry associated with the instance of
346 CIM_RecordLog is NULL, then the RecordData and RecordFormat properties of the instance of
347 CIM_LogEntry shall be non-NULL.

348 If the RecordData and RecordFormat properties of an instance of CIM_LogEntry associated with the
349 instance of CIM_RecordLog are NULL, then the PerceivedSeverity, OwningEntity, MessageID properties
350 and at least one of the properties Message and MessageArguments of the instance of CIM_LogEntry
351 shall be non-NULL.

352 **7.2.2 CIM_RecordLogCapabilities.RequestedStatesSupported**

353 The CIM_RecordLogCapabilities.RequestedStatesSupported property is an array that contains the
354 supported requested states for the instance of CIM_RecordLog. This property shall be the super set of
355 the values to be used as the RequestedState parameter in the RequestStateChange() method (see 8.2).

356 The value of the CIM_RecordLogCapabilities.RequestedStatesSupported property shall be an empty
357 array or any combination of the following values: 2 (Enabled), 3 (Disabled), or 11 (Reset).

358 **7.2.3 CIM_RecordLogCapabilities.ElementNameEditSupported**

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

361 **7.2.4 CIM_RecordLogCapabilities.MaxElementNameLen**

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

364 **7.2.5 CIM_RecordLogCapabilities.MethodsSupported**

365 CIM_RecordLogCapabilities.MethodsSupported may be implemented.

366 If this property contains a value of 2 (ClearLog), then ClearLog() method shall be implemented.

367 **7.3 Log state management (optional)**

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

370 **7.3.1 Log state management support**

371 When a CIM_RecordLogCapabilities.RequestedStatesSupported property is an empty array, log state
372 management shall not be supported.

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

376 **7.4 CIM_RecordLog.RequestedState**

377 The CIM_RecordLog.RequestedState property shall have a value of 12 (Not Applicable), 5 (No Change),
378 or a value contained in the CIM_RecordLogCapabilities.RequestedStatesSupported property array of the
379 associated CIM_RecordLogCapabilities instance (see 7.2.2).

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

386 **7.4.1 RequestedState — 12 (Not Applicable) value**

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

389 **7.4.2 RequestedState — 5 (No Change) value**

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

392 7.5 Representing log state

393 The log's state shall be represented by two properties: CIM_RecordLog.EnabledState (see 7.5.1) and
394 CIM_RecordLog.LogState (see 7.5.2).

395 7.5.1 CIM_RecordLog.EnabledState

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

402 **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.

403 7.5.1.1 CIM_RecordLog.EnabledState — 6 (Enabled but Offline) value

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

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

409 7.5.2 CIM_RecordLog.LogState

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

418 **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.

LogState Value	Description	EnabledState Value	Extended Description
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.

419 **7.5.2.1 CIM_RecordLog.LogState — 3 (Erasing) value**

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

424 **7.6 CIM_UseOfLog**

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

427 **7.7 CIM_HostedDependency**

428 An instance of the CIM_HostedDependency association may reference an instance of CIM_RecordLog
 429 and an instance of the subclass of CIM_ManagedSystemElement.

430 **7.8 CIM_RecordLog.OverwritePolicy property**

431 The CIM_RecordLog.OverwritePolicy property indicates the behavior of the log when it has reached the
 432 maximum capacity of its entries. The CIM_RecordLog.OverwritePolicy property also affects the
 433 CIM_RecordLog.EnabledState property, as described in 7.5.1.1.

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

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

439 **8 Methods**

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

442 **8.1 CIM_RecordLog.ClearLog()**

443 The CIM_RecordLog.ClearLog() method is used to request the deletion of all entries in the record log for
 444 an instance of CIM_RecordLog. . If the CIM_RecordLogCapabilities.MethodsSupported property contains
 445 a value of 2 (ClearLog) then the implementation shall support ClearLog() method.

446 A return code value of zero shall indicate that the clearing of the log entries was successfully initiated.

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

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

449

450

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

451 8.2 CIM_RecordLog.RequestStateChange()

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

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

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

459 When the RequestedState parameter is set to 2 (Enabled) but the CIM_RecordLog.EnabledState
460 property has a value of 6 (Enabled but Offline), the RequestStateChange() method invocation shall return
461 2 (Error Occurred).

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

464 No standard messages are defined for this method.

465 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

466

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.

Qualifiers	Name	Type	Description/Values
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

467 **8.3 Profile conventions for operations**

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

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

471 The default list of operations is as follows:

- 472 • Associators()
- 473 • AssociatorNames()
- 474 • EnumerateInstances()
- 475 • EnumerateInstanceNames()
- 476 • GetInstance()
- 477 • References()
- 478 • ReferenceNames()

479 **8.4 CIM_ElementCapabilities**

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

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

484 **Table 7 – Operations: CIM_ElementCapabilities**

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

485 **8.5 CIM_RecordLogCapabilities**

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

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

488 8.6 CIM_RecordLog

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

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

493 **Table 8 – Operations: CIM_RecordLog**

Operation	Requirement	Messages
ModifyInstance	Optional: See 8.6.1.	None

494 8.6.1 CIM_RecordLog — ModifyInstance

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

497 The ModifyInstance operation shall be supported and CIM_RecordLog.ElementName shall be modifiable
 498 when the ElementNameEditSupported property of the CIM_EnabledLogicalElementCapabilities instance
 499 that is associated with the CIM_RecordLog instance has a value of TRUE. See 8.6.2.

500 8.6.2 CIM_RecordLog.ElementName

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

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

510 8.7 CIM_LogEntry

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

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

515 **Table 9 – Operations: CIM_LogEntry**

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

516 8.7.1 CIM_LogEntry DeleteInstance

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

520 **8.8 CIM_UseOfLog**

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

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

525 **Table 10 – Operations: CIM_UseOfLog**

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

526 **8.9 CIM_LogManagesRecord**

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

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

531 **Table 11 – Operations: CIM_LogManagesRecord**

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

532 **8.10 CIM_HostedDependency**

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

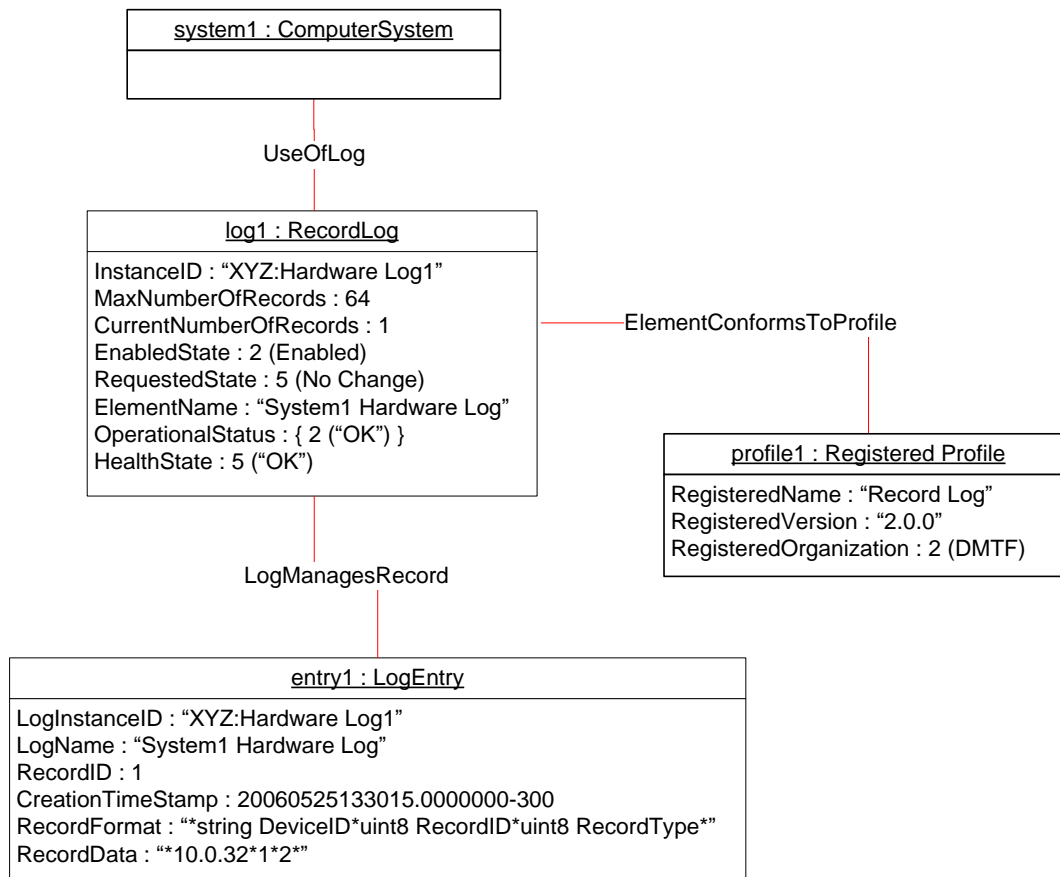
536 NOTE: Related profiles may define additional requirements on operations for the profile class.

537 **Table 12 – Operations: CIM_HostedDependency**

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

538 **9 Use cases**539 This clause contains object diagrams and use cases for the *Record Log Profile*.540 **9.1 Object diagrams**

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

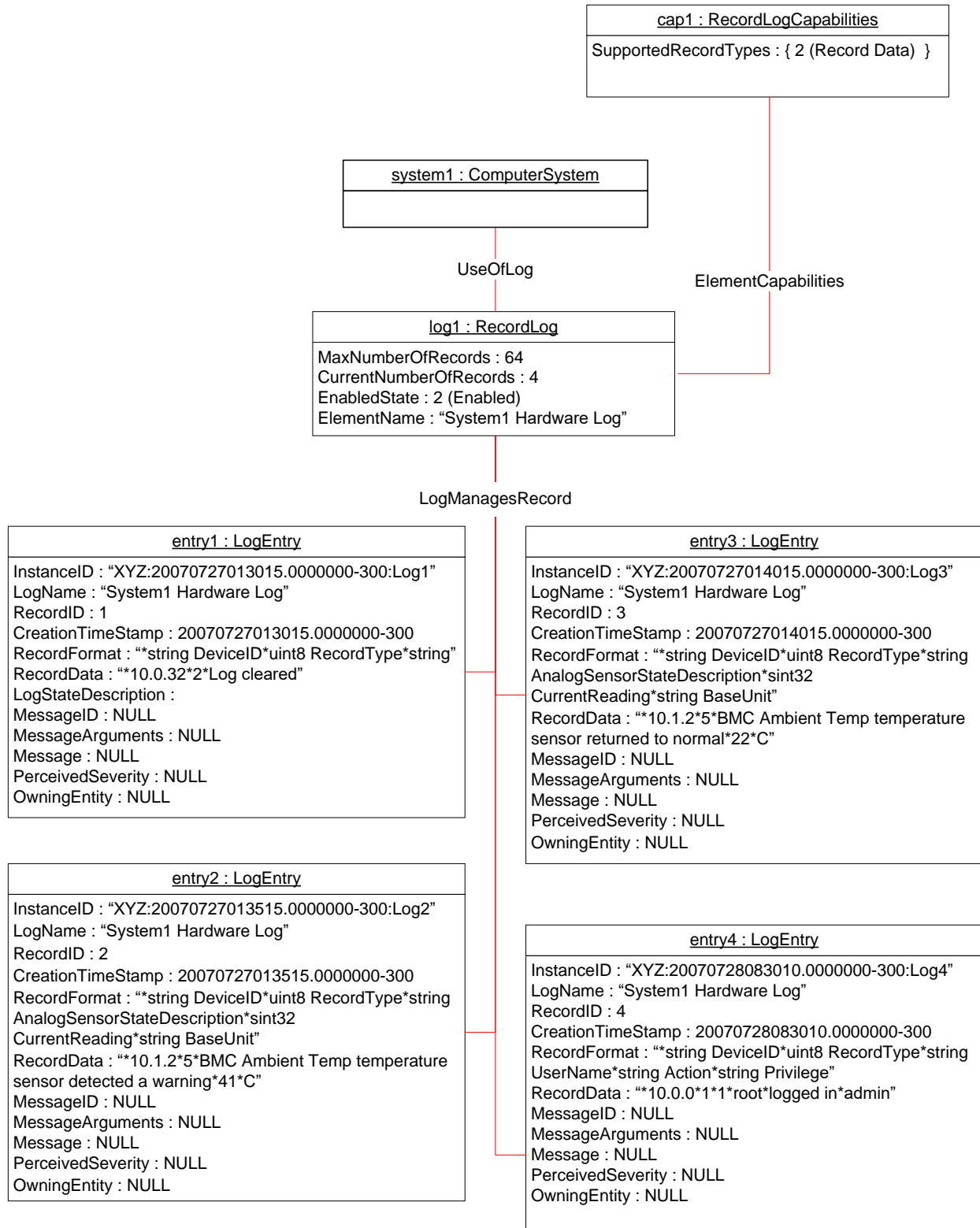


545

546

Figure 2 – RecordLog instance

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

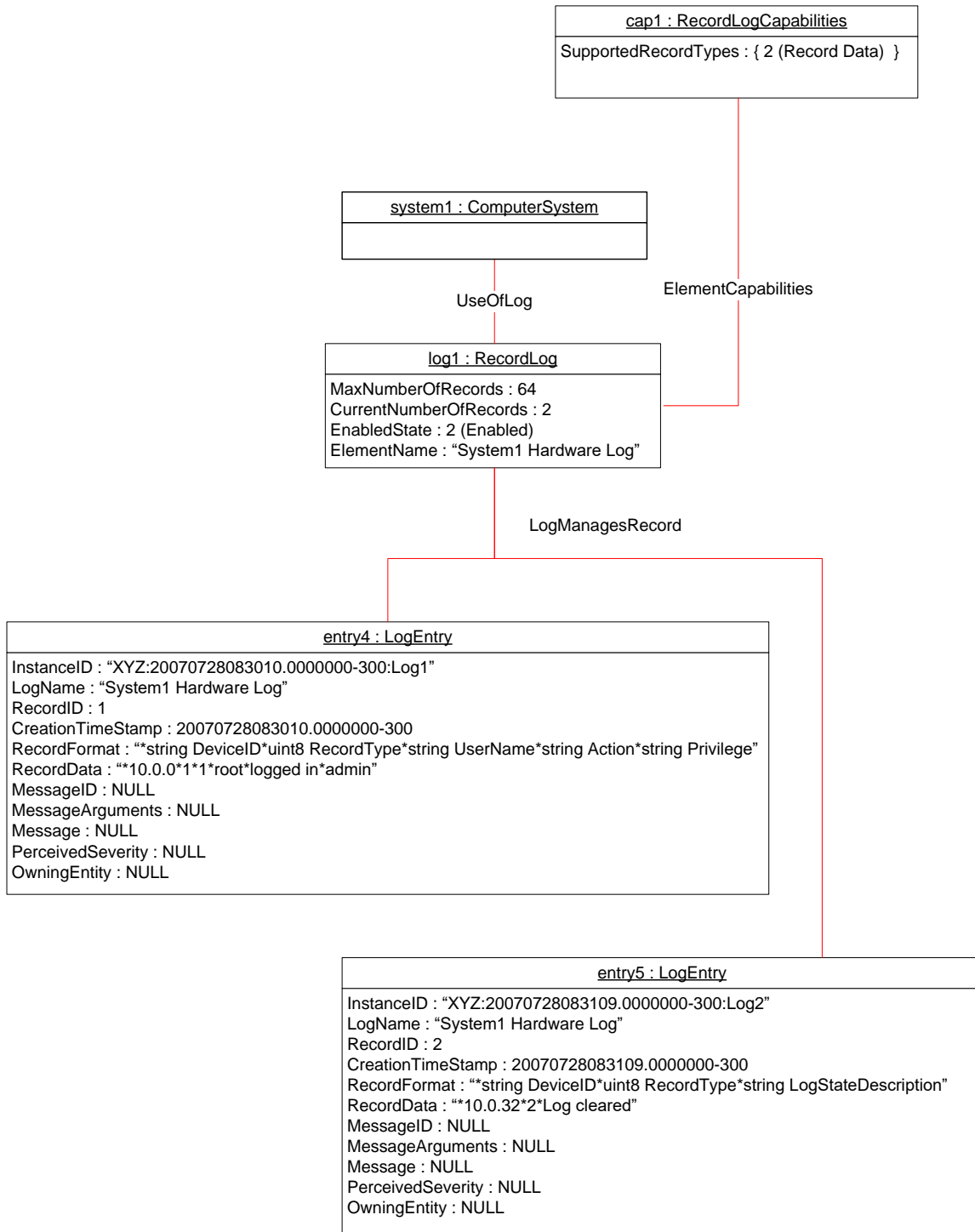


554

555

Figure 3 – RecordLog instance before the log is cleared

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



560

561

Figure 4 – RecordLog instance after the log is cleared

562 **9.2 Identify the log by the name**

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

565 **9.3 Browse the records of the log**

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

569 **9.4 Sort the log records based on the time stamp of the log entry**

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

- 571 1) Iterate through all the instances of CIM_LogEntry that are associated through the
572 CIM_LogManagesRecord association to the given instance of CIM_RecordLog that represents
573 the log record.
- 574 2) Sort the instances of CIM_LogEntry based on the CreationTimeStamp property value in LIFO
575 order.

576 **9.5 Delete a log entry**

577 A client can delete a log entry as follows:

- 578 1) Select the instance of CIM_LogEntry that represents the desired log entry to be deleted.
- 579 3) Execute DeleteInstance operation on the selected instance of CIM_LogEntry.

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

582 **9.6 Clear the log**

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

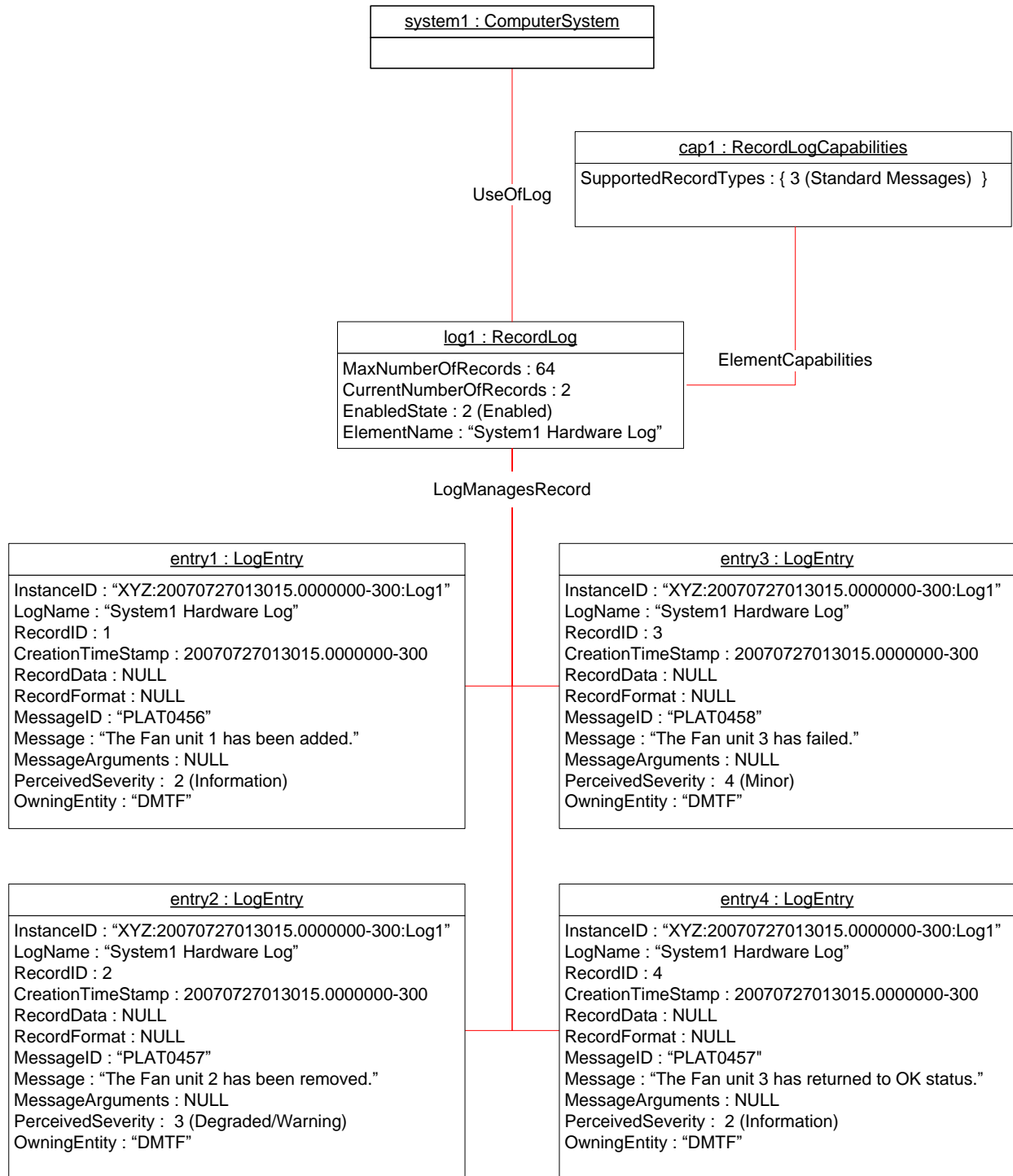
584 **9.7 Determine which record types are supported**

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

- 586 1) From the instance of CIM_RecordLog use the CIM_ElementCapabilities association to obtain
587 the associated instance of CIM_RecordLogCapabilities.
- 588 4) The SupportedRecordTypes property of the CIM_RecordLogCapabilities instance indicates
589 which record types are supported by the implementation.

590 **9.8 RecordLog instance for standard messages**

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



594

595

Figure 5 – RecordLog instance for standard messages

596

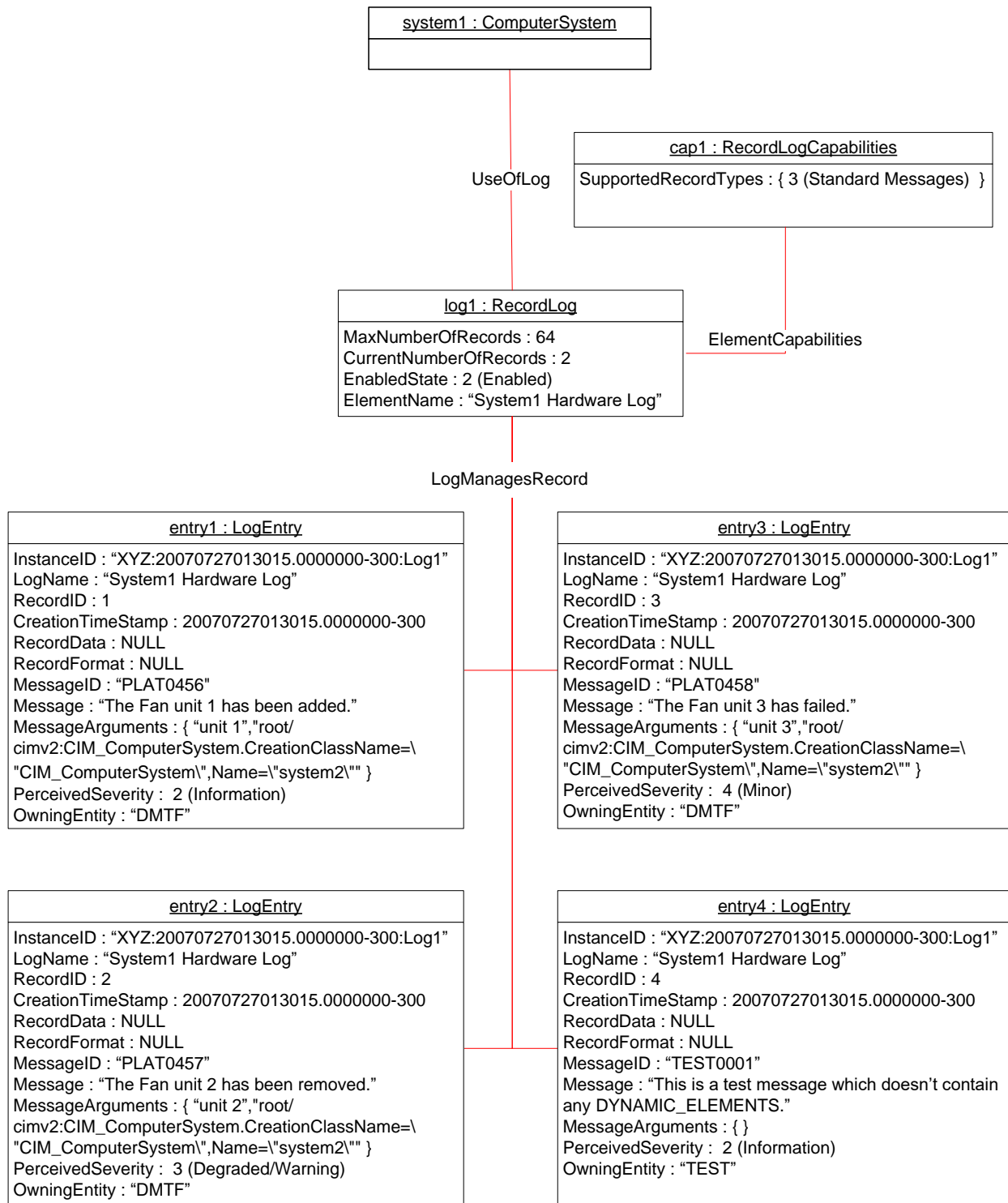
9.9 RecordLog Instance for standard messages with MessageArguments supported

597

598

599 Figure 6 shows four messages using the standard message record log type. For these record log entries
600 that support standard messages both the Message and MessageArguments properties are supported.
601 entry4 shows a potential message that does not contain any DYNAMIC_ELEMENTS. For this case, the
602 MessageArguments property is an empty array.

603



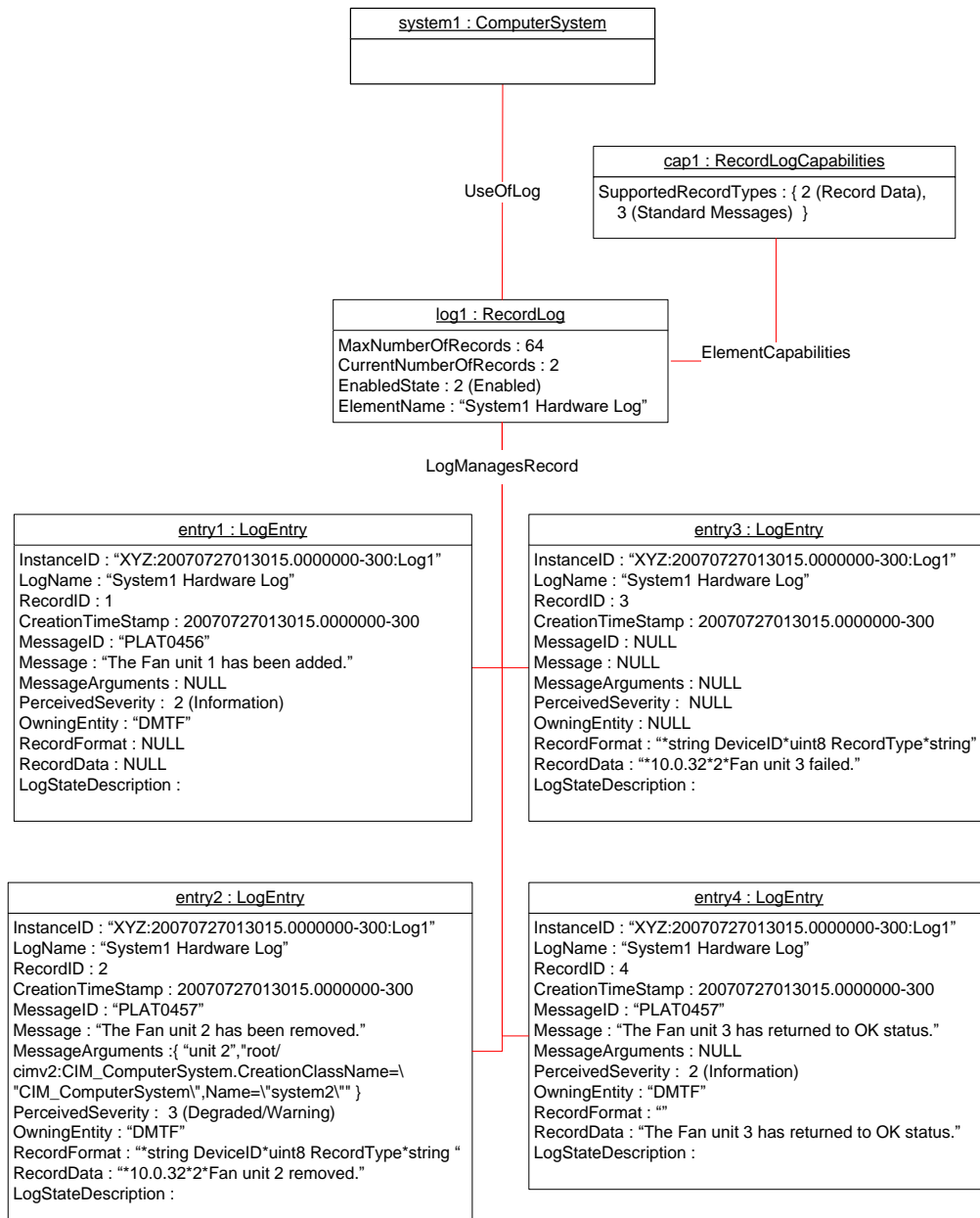
604

605

Figure 6 – RecordLog with standard message and MessageArguments

606 **9.10 RecordLog Instance for Record Data and Standard Messages**

607 Figure 7 shows a Record Log that supports both the Record Data and Standard Message formats. In
608 addition, for the standard messages both the Message and MessageArguments properties are supplied.
609 The entry1 log entry contains only the Standard Message Format with the Message property, but not the
610 MessageArguments property. The entry2 log entry contains both the Record Data and Standard Message
611 Formats. The entry3 log entry contains only the Record Data Format. And the entry4 log entry contains
612 both the Record Data and Standard Message Formats. The entry4 log entry shows properties for the
613 Record Data Format where the RecordFormat property is an empty string, which indicates that the
614 RecordData is a free-format string.



615

616

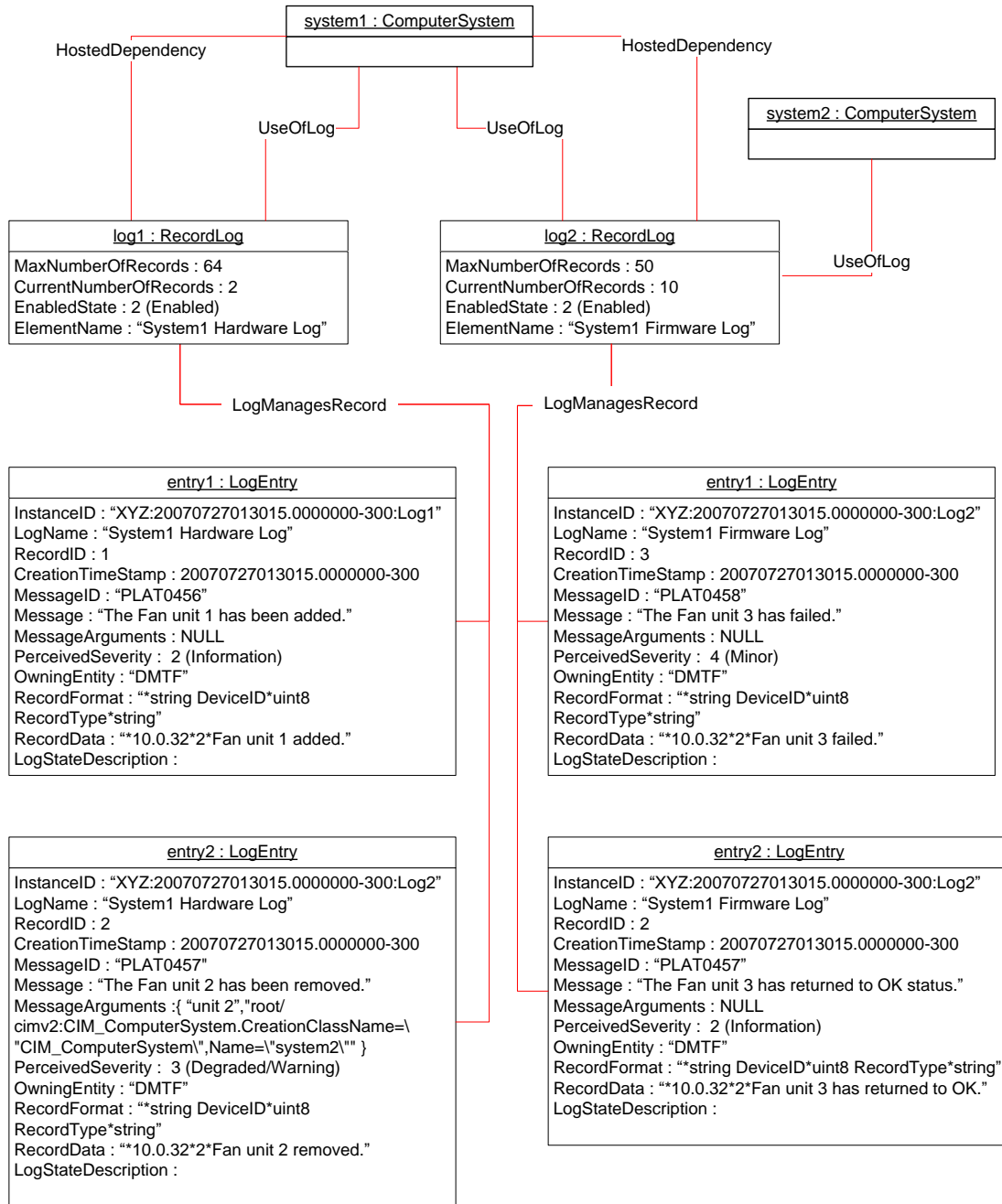
Figure 7 – RecordLog instances for both record types

617 9.11 List all logs hosted on this system

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

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

- 621 1) From the CIM_ComputerSystem instance for system1 enumerate all CIM_HostedDependency
622 association instances where CIM_RecordLog is the result class.



623

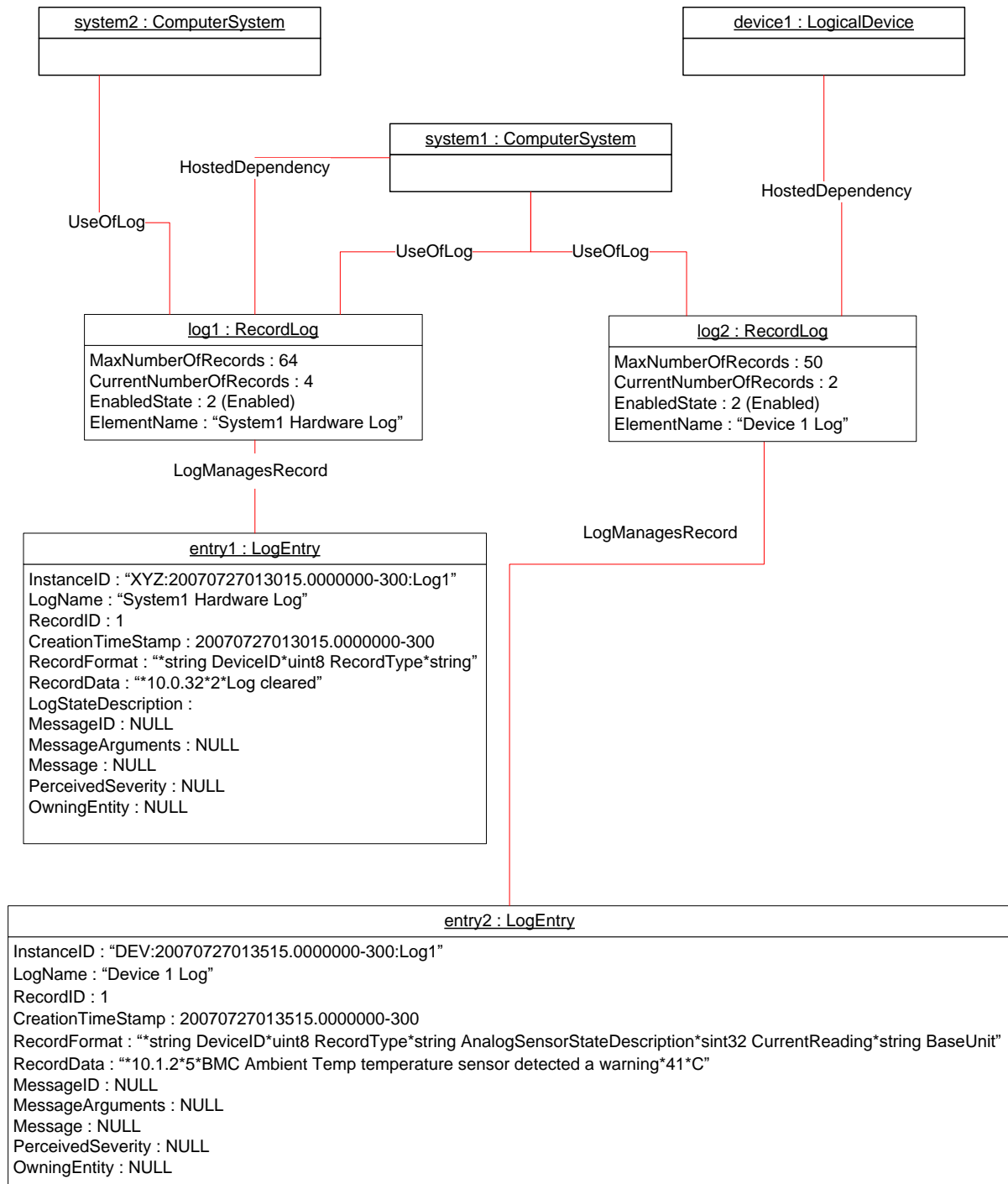
624

Figure 8 – Record log hosted on system1

625

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

626



627

628

Figure 9 – Record logs hosted on system1 and device1

629 **10 CIM Elements**

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

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

Element Name	Requirement	Description
Classes		
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.
Indications		
None defined in this profile		

634 **10.1 CIM_ElementCapabilities**

635 CIM_ElementCapabilities associates an instance of CIM_RecordLog with an instance of
 636 CIM_EnabledLogicalElementCapabilities that describes the capabilities of CIM_RecordLog. Table 14
 637 provides information about the properties of CIM_ElementCapabilities.

638 **Table 14 – 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

639 **10.2 CIM_RecordLogCapabilities**

640 CIM_RecordLogCapabilities represents the capabilities of the log. Table 15 provides information about
 641 the properties of CIM_RecordLogCapabilities.

642 **Table 15 – Class: CIM_RecordLogCapabilities**

Elements	Requirement	Notes
InstanceID	Mandatory	Key
RequestedStatesSupported	Mandatory	See 7.2.2.
ElementNameEditSupported	Mandatory	See 7.2.3.
MaxElementNameLen	Conditional	See 7.2.4.
SupportedRecordTypes	Mandatory	See 7.2.1.
MethodsSupported	Optional	See 7.2.5 .

643 **10.3 CIM_LogManagesRecord**

644 CIM_LogManagesRecord associates the CIM_RecordLog instance, which represents the log, with an
 645 instance of CIM_LogEntry, which represents an entry within the log. Table 16 provides information about
 646 the properties of CIM_LogManagesRecord.

647 **Table 16 – 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

648 **10.4 CIM_LogEntry**

649 CIM_LogEntry represents the log entry within the log in the managed system. Table 17 provides
 650 information about the properties of CIM_LogEntry.

651 **Table 17 – Class: CIM_LogEntry**

Elements	Requirement	Notes
InstanceID	Mandatory	Key
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.

Elements	Requirement	Notes
ElementName	Mandatory	The property shall match pattern “.*”.
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.

652 10.5 CIM_RecordLog

653 CIM_RecordLog represents the log in the managed system. Table 18 provides information about the
654 properties of CIM_RecordLog.

655 **Table 18 – Class: CIM_RecordLog**

Elements	Requirement	Notes
InstanceID	Mandatory	Key
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

656 10.6 CIM_RegisteredProfile

657 CIM_RegisteredProfile identifies the *Record Log Profile* in order for a client to determine the conformance
658 with the profile. The CIM_RegisteredProfile class is defined by the [Profile Registration Profile](#). With the
659 exception of the mandatory values specified for the properties in Table 19, the behavior of the
660 RegisteredProfile instance is per the [Profile Registration Profile](#).

661 **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 “2.1.0”.
RegisteredOrganization	Mandatory	This property shall have a value of 2 (DMTF).

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

665 **10.7 CIM_UseOfLog**

666 CIM_UseOfLog associates CIM_RecordLog, which represents the log, with a subclass of
 667 CIM_ManagedSystemElement, which represents the element that uses or populates the log. Table 20
 668 provides information about the properties of CIM_UseOfLog.

669 **Table 20 – 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

670 **10.8 CIM_HostedDependency**

671 CIM_HostedDependency associates CIM_RecordLog, which represents the log, with a subclass of
 672 CIM_ManagedElement, which represents the element that hosts the log. Table 21 provides information
 673 about the properties of CIM_HostedDependency.

674 **Table 21 – Class: CIM_HostedDependency**

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 hosts the log. Cardinality 1..*, indicating one or many references

675
676
677
678

ANNEX A (informative)

Change log

Version	Date	Description
1.0.0	2007-10-04	
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">• Add Standard Message support. Either Record Data or Standard Message Format must be implemented.• The new class CIM_RecordLogCapabilities, which subclasses (and replaces) CIM_EnabledLogicalElementCapabilities, must now be implemented.
2.1.0	2016-08-15	New property CIM_RecordLogCapabilities.MethodsSupported was introduced to enable a client to intrinsically discover which extrinsic methods are supported by an implementation of this profile

679