



1
2
3
4

Document Number: DSP1010

Date: 2007-02-13

Version: 1.0.0c

5 **Record Log Profile**

6 **Document Type: Specification**
7 **Document Status: Preliminary Standard**
8 **Document Language: E**
9

Record Log Profile

10 Copyright notice

11 Copyright © 2007 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 for uses consistent with this purpose, provided that correct attribution is given. As DMTF
15 specifications may be revised from time to time, the particular version and release date should always be
16 noted.

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27 withdrawn or modified after publication, and shall be indemnified and held harmless by any party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

CONTENTS

31	Foreword	5
32	Introduction	6
33	1 Scope	7
34	2 Normative References.....	7
35	2.1 Approved References	7
36	2.2 References under Development	7
37	2.3 Other References.....	7
38	3 Terms and Definitions.....	7
39	4 Symbols and Abbreviated Terms.....	9
40	5 Synopsis	9
41	6 Description	9
42	7 Implementation.....	10
43	7.1 Representing Logs.....	10
44	7.2 CIM_EnabledLogicalElementCapabilities.....	11
45	7.3 CIM_RecordLog.RequestedState.....	11
46	7.4 Representing Log State	12
47	7.5 CIM_UseOfLog	13
48	7.6 CIM_RecordLog.OverwritePolicy Property	13
49	8 Methods.....	13
50	8.1 CIM_RecordLog.ClearLog().....	14
51	8.2 CIM_RecordLog.RequestStateChange().....	14
52	8.3 Profile Conventions for Operations.....	15
53	8.4 CIM_ElementCapabilities	16
54	8.5 CIM_EnabledLogicalElementCapabilities.....	16
55	8.6 CIM_RecordLog.....	16
56	8.7 CIM_LogEntry	17
57	8.8 CIM_UseOfLog	17
58	8.9 CIM_LogManagesRecord.....	17
59	9 Use Cases	18
60	9.1 Object Diagrams	18
61	9.2 Identify the Log by the Name.....	20
62	9.3 Browse the Records of the Log	20
63	9.4 Sort the Log Records Based on the Time Stamp of the Log Entry	20
64	9.5 Delete a Log Entry	20
65	9.6 Clear the Log	21
66	10 CIM Elements.....	21
67	10.1 CIM_ElementCapabilities	21
68	10.2 CIM_EnabledLogicalElementCapabilities.....	22
69	10.3 CIM_LogManagesRecord.....	22
70	10.4 CIM_LogEntry	22
71	10.5 CIM_RecordLog.....	23
72	10.6 CIM_RegisteredProfile.....	23
73	10.7 CIM_UseOfLog	23
74	ANNEX A (informative) Change Log	24
75	ANNEX B (informative) Acknowledgements	25
76		

Record Log Profile

77 **Figures**

78	Figure 1 – Record Log Profile: Class Diagram	10
79	Figure 2 – RecordLog Instance.....	18
80	Figure 3 – RecordLog Instance Before the Log Is Cleared	19
81	Figure 4 – RecordLog Instance after the Log Is Cleared.....	20

82

83 **Tables**

84	Table 1 – Referenced Profiles	9
85	Table 2 – EnabledState Value Description	12
86	Table 3 – LogState Value Description and Mapping to EnabledState Value	13
87	Table 4 – CIM_RecordLog.ClearLog() Method: Return Code Values	14
88	Table 5 – CIM_RecordLog.RequestStateChange() Method: Return Code Values	14
89	Table 6 – CIM_RecordLog.RequestStateChange() Method: Parameters	15
90	Table 7 – Operations: CIM_ElementCapabilities	16
91	Table 8 – Operations: CIM_RecordLog	16
92	Table 9 – Operations: CIM_LogEntry	17
93	Table 10 – Operations: CIM_UseOfLog.....	17
94	Table 11 – Operations: CIM_LogManagesRecord	17
95	Table 12 – CIM Elements: Record Log Profile	21
96	Table 13 – Class: CIM_ElementCapabilities.....	21
97	Table 14 – Class: CIM_EnabledLogicalElementCapabilities.....	22
98	Table 15 – Class: CIM_LogManagesRecord	22
99	Table 16 – Class: CIM_LogEntry	22
100	Table 17 – Class: CIM_RecordLog.....	23
101	Table 18 – Class: CIM_RegisteredProfile.....	23
102	Table 19 – Class: CIM_UseOfLog	23

103

104

Foreword

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

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

Introduction

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

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

117

Record Log Profile

118 1 Scope

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

122 2 Normative References

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

126 2.1 Approved References

127 DMTF [DSP0004](#), *CIM Infrastructure Specification 2.3.0*

128 DMTF [DSP0200](#), *CIM Operations over HTTP 1.2.0*

129 DMTF [DSP1000](#), *Management Profile Specification Template*

130 DMTF [DSP1001](#), *Management Profile Specification Usage Guide*

131 2.2 References under Development

132 DMTF [DSP0215](#), *SM Managed Element Addressing Specification (SM ME Addressing)*

133 DMTF [DSP1033](#), *Profile Registration Profile 1.0*

134 2.3 Other References

135 ISO/IEC Directives, Part 2, [Rules for the structure and drafting of International Standards](#)

136 [Unified Modeling Language \(UML\) from the Open Management Group \(OMG\)](#)

137 3 Terms and Definitions

138 For the purposes of this document, the following terms and definitions apply. The terms and definitions in
139 DSP1033 and DSP1001 also apply.

140 3.1

141 can

142 used for statements of possibility and capability, whether material, physical, or causal

143 3.2

144 cannot

145 used for statements of possibility and capability, whether material, physical or causal

Record Log Profile

- 146 **3.3**
147 **conditional**
148 indicates requirements to be followed strictly in order to conform to the document when the specified
149 conditions are met
- 150 **3.4**
151 **mandatory**
152 indicates requirements to be followed strictly in order to conform to the document and from which no
153 deviation is permitted
- 154 **3.5**
155 **may**
156 indicates a course of action permissible within the limits of the document
- 157 **3.6**
158 **need not**
159 indicates a course of action permissible within the limits of the document
- 160 **3.7**
161 **optional**
162 indicates a course of action permissible within the limits of the document
- 163 **3.8**
164 **referencing profile**
165 indicates a profile that owns the definition of this class and can include a reference to this profile in its
166 "Referenced Profiles" table
- 167 **3.9**
168 **shall**
169 indicates requirements to be followed strictly in order to conform to the document and from which no
170 deviation is permitted
- 171 **3.10**
172 **shall not**
173 indicates requirements to be followed strictly in order to conform to the document and from which no
174 deviation is permitted
- 175 **3.11**
176 **should**
177 indicates that among several possibilities, one is recommended as particularly suitable, without
178 mentioning or excluding others, or that a certain course of action is preferred but not necessarily required
- 179 **3.12**
180 **should not**
181 indicates that a certain possibility or course of action is deprecated but not prohibited
- 182 **3.13**
183 **unspecified**
184 indicates that this profile does not define any constraints for the referenced CIM element or operation

185 4 Symbols and Abbreviated Terms

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

187 4.1

188 LIFO

189 Last In, First Out

190 5 Synopsis

191 **Profile Name:** *Record Log Profile*

192 **Version:** 1.0.0

193 **Organization:** Server Management Working Group

194 **CIM schema version:** 2.14

195 **Central Class:** CIM_RecordLog

196 **Scoping Class:** CIM_RecordLog

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

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

200

Table 1 – Referenced Profiles

Profile Name	Organization	Version	Relationship	Behavior
<i>Profile Registration Profile</i>	DMTF	1.0	Mandatory	None

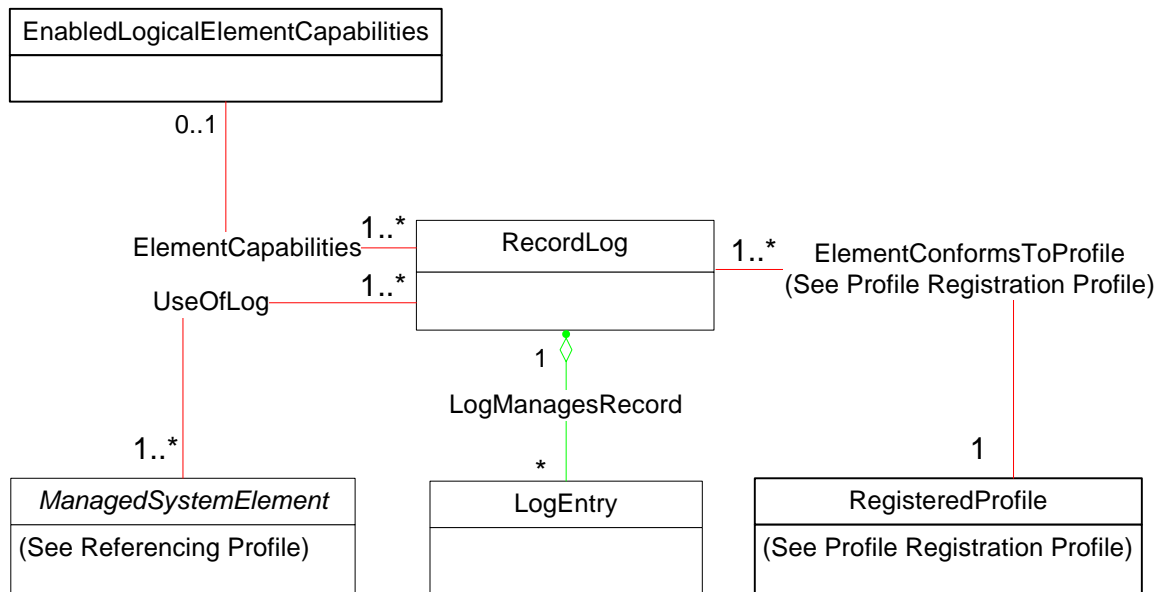
201 6 Description

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

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

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

Record Log Profile



211

212

Figure 1 – Record Log Profile: Class Diagram

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

216 7 Implementation

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

220 7.1 Representing Logs

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

225 7.1.1 CIM_LogEntry.LogInstanceID

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

229 7.1.2 CIM_LogEntry.LogName

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

233 7.2 CIM_EnabledLogicalElementCapabilities

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

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

240 7.2.1 CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported

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

247 7.2.2 CIM_EnabledLogicalElementCapabilities.ElementNameEditSupported

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

251 7.2.3 CIM_EnabledLogicalElementCapabilities.MaxElementNameLen

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

254 7.2.4 Log State Management (Optional)

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

257 7.2.5 Log State Management Support

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

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

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

266 7.3 CIM_RecordLog.RequestedState

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

270 When log state management is supported and the RequestStateChange() method is successfully
271 executed, the RequestedState property shall be set to the value of the RequestedState parameter of the
272 RequestStateChange() method. After the RequestStateChange() method has successfully executed, the

Record Log Profile

273 RequestedState and EnabledState parameters shall have equal values, with the exception of the
274 transitional requested state 11 (Reset). The value of the RequestedState property may also change as a
275 result of a non-CIM implementation's request for a change to the log's enabled state.

276 7.3.1 RequestedState—12 (Not Applicable) Value

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

279 7.3.2 RequestedState—5 (No Change) Value

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

282 7.4 Representing Log State

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

285 7.4.1 CIM_RecordLog.EnabledState

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

292 **Table 2 – EnabledState Value Description**

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

293 7.4.1.1 CIM_RecordLog.EnabledState—6 (Enabled but Offline) Value

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

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

299 7.4.2 CIM_RecordLog.LogState

300 The CIM_RecordLog.LogState property is used to describe a more granular state of the log than that of
301 the CIM_RecordLog.EnabledState property. Table 3 describes the mapping between the values of the
302 CIM_RecordLog.LogState property and the corresponding description of the granular state of the log. The
303 CIM_RecordLog.LogState property shall match the values that are specified in Table 3. Additionally,
304 Table 3 describes the mapping between the LogState property and the EnabledState property. When the

305 CIM_RecordLog.LogState property has a value that matches the value in the “LogState Value” column in
 306 Table 3, the CIM_RecordLog.EnabledState property shall have a value that matches the value in the
 307 “EnabledState Value” column for that row.

308 **Table 3 – LogState Value Description and Mapping to EnabledState Value**

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

309 7.4.2.1 CIM_RecordLog.LogState—2 (Erasing) Value

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

314 7.5 CIM_UseOfLog

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

317 7.6 CIM_RecordLog.OverwritePolicy Property

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

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

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

326 8 Methods

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

Record Log Profile

329 8.1 CIM_RecordLog.ClearLog()

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

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

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

335 **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

336 8.2 CIM_RecordLog.RequestStateChange()

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

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

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

344 When the RequestedState parameter is set to 2 (Enabled) but the CIM_RecordLog.EnabledState
345 property has a value of 6 (Enabled but Offline), the RequestStateChange() method invocation shall return
346 2 (Error Occurred).

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

349 No standard messages are defined for this method.

350 **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

351

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

352 8.3 Profile Conventions for Operations

353 Support for operations for each profile class (including associations) is specified in the following
 354 subclauses. Each subclause includes either the statement “All operations in the default list in section 8.3
 355 are supported as described by [DSP0200 version 1.2](#)” or a table listing all of the operations that are not
 356 supported by this profile or where the profile requires behavior other than that described by [DSP0200](#)
 357 [version 1.2](#).

358 The default list of operations is as follows:

- 359 • GetInstance
- 360 • Associators
- 361 • AssociatorNames
- 362 • References
- 363 • ReferenceNames
- 364 • EnumerateInstances
- 365 • EnumerateInstanceNames

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

Record Log Profile

368 8.4 CIM_ElementCapabilities

369 Table 7 lists operations that either have special requirements beyond those from [DSP0200 version 1.2](#) or
370 shall not be supported.

371 **Table 7 – Operations: CIM_ElementCapabilities**

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

372 8.5 CIM_EnabledLogicalElementCapabilities

373 All operations in the default list in section 8.3 are supported as described by [DSP0200 version 1.2](#).

374 8.6 CIM_RecordLog

375 Table 8 lists operations that either have special requirements beyond those from [DSP0200 version 1.2](#) or
376 shall not be supported.

377 **Table 8 – Operations: CIM_RecordLog**

Operation	Requirement	Messages
ModifyInstance	Optional: See section 8.6.1.	None

378 8.6.1 CIM_RecordLog—ModifyInstance

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

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

384 8.6.1.1 CIM_RecordLog.ElementName

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

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

394 8.7 CIM_LogEntry

395 Table 9 lists operations that either have special requirements beyond those from [DSP0200 version 1.2](#) or
396 shall not be supported.

397 **Table 9 – Operations: CIM_LogEntry**

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

398 8.7.1 CIM_LogEntry DeleteInstance

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

402 8.8 CIM_UseOfLog

403 Table 10 lists operations that either have special requirements beyond those from [DSP0200 version 1.2](#)
404 or shall not be supported.

405 **Table 10 – Operations: CIM_UseOfLog**

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

406 8.9 CIM_LogManagesRecord

407 Table 11 lists operations that either have special requirements beyond those from [DSP0200 version 1.2](#)
408 or shall not be supported.

409 **Table 11 – Operations: CIM_LogManagesRecord**

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

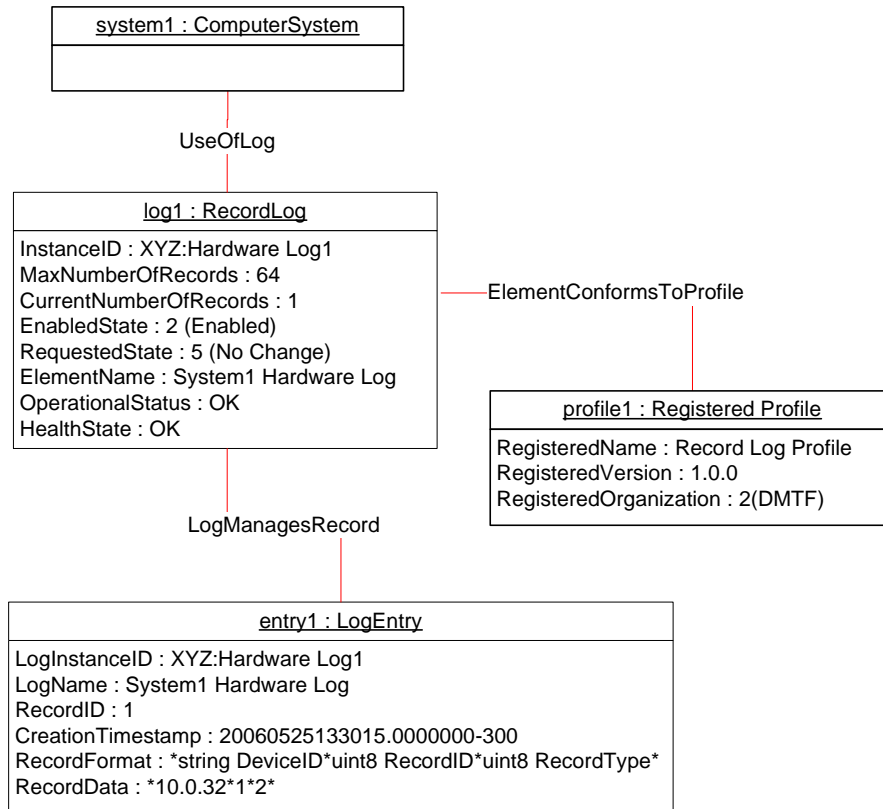
Record Log Profile

410 9 Use Cases

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

412 9.1 Object Diagrams

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

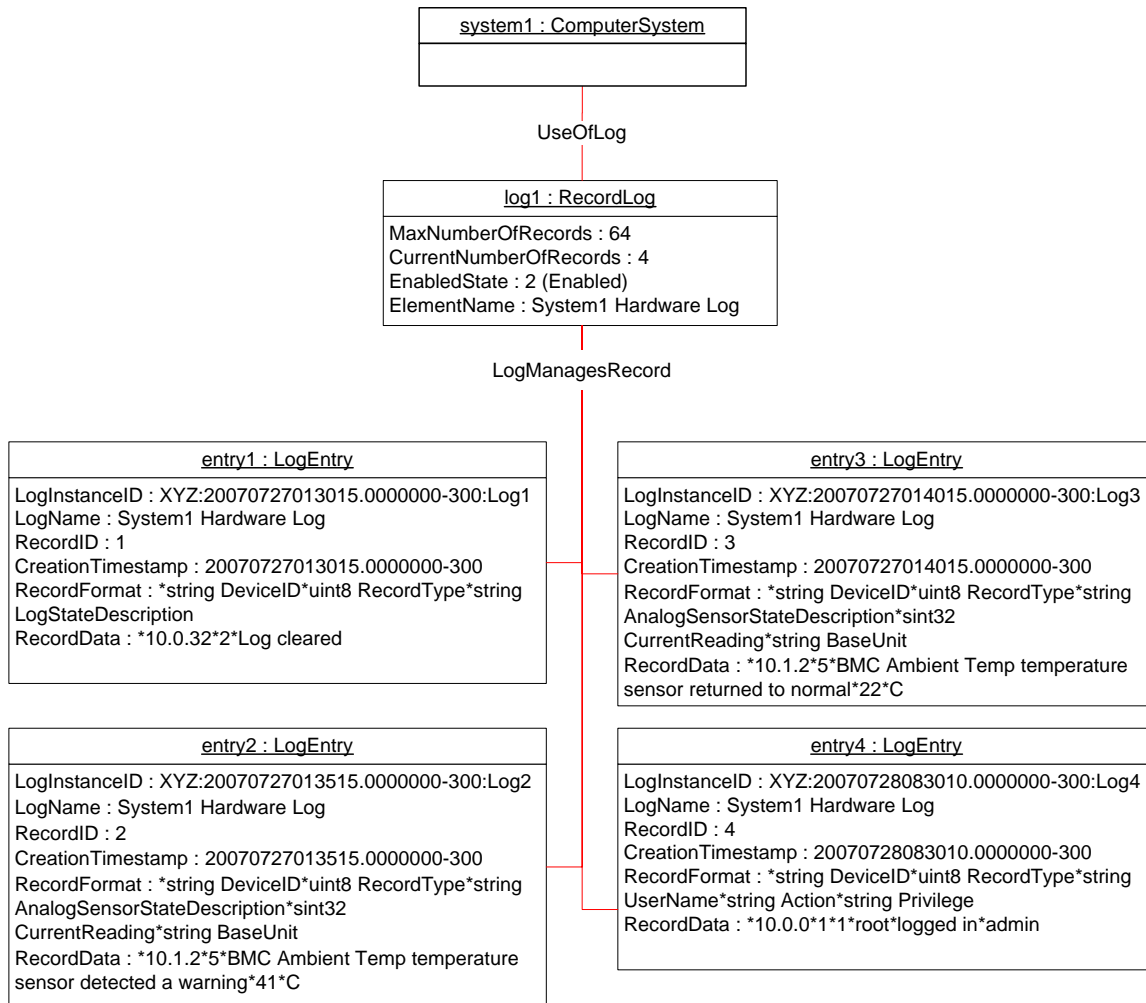


417

418

Figure 2 – RecordLog Instance

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



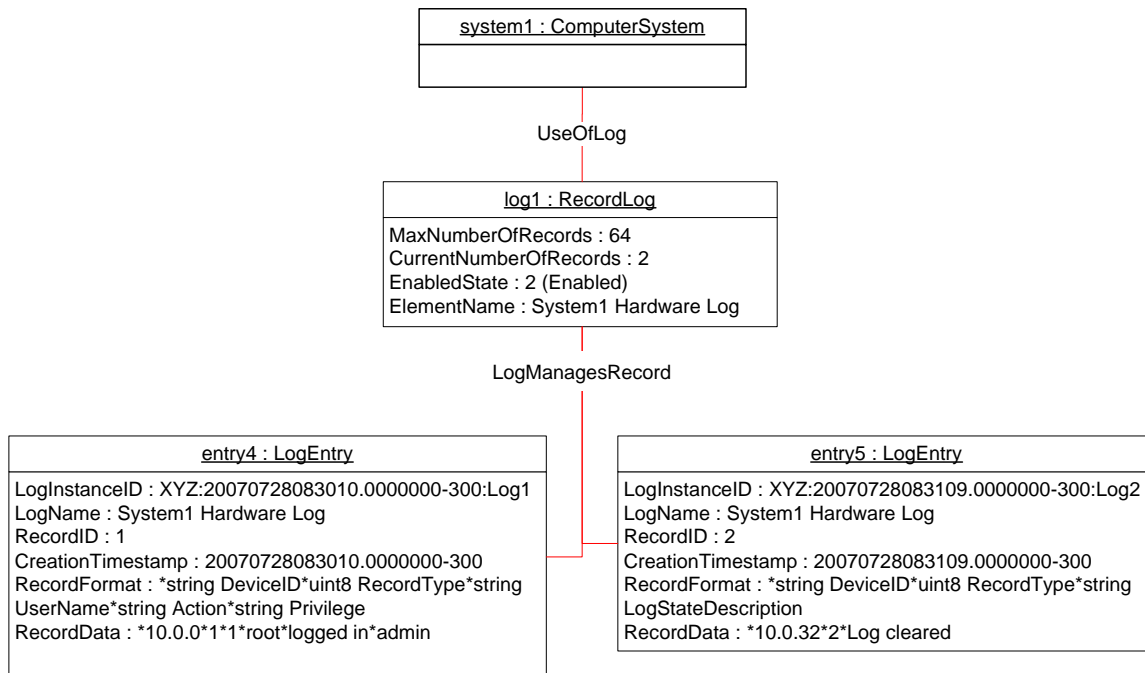
426

427

Figure 3 – RecordLog Instance Before the Log Is Cleared

Record Log Profile

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



433 **Figure 4 – RecordLog Instance after the Log Is Cleared**

434 9.2 Identify the Log by the Name

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

437 9.3 Browse the Records of the Log

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

441 9.4 Sort the Log Records Based on the Time Stamp of the Log Entry

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

- 443 1. Iterate through all the instances of CIM_LogEntry that are associated through the
444 CIM_LogManagesRecord association to the given instance of CIM_RecordLog that represents
445 the log record.
- 446 2. Sort the instances of CIM_LogEntry based on the CreationTimeStamp property value in LIFO
447 order.

448 9.5 Delete a Log Entry

449 A client can delete a log entry as follows:

- 450 1. Select the instance of CIM_LogEntry that represents the desired log entry to be deleted.
 451 2. Execute DeleteInstance operation on the selected instance of CIM_LogEntry.
 452 Upon successful execution, the instance of CIM_LogEntry and the instance of CIM_LogManagesRecord
 453 that associates the log entry to the instance of CIM_RecordLog are deleted.

454 9.6 Clear the Log

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

456 10 CIM Elements

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

460 **Table 12 – CIM Elements: Record Log Profile**

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

461 10.1 CIM_ElementCapabilities

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

466 **Table 13 – Class: CIM_ElementCapabilities**

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

Record Log Profile

467 10.2 CIM_EnabledLogicalElementCapabilities

468 CIM_EnabledLogicalElementCapabilities represents the capabilities of the log.

469 **Table 14 – Class: CIM_EnabledLogicalElementCapabilities**

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

470 10.3 CIM_LogManagesRecord

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

474 **Table 15 – Class: CIM_LogManagesRecord**

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

475 10.4 CIM_LogEntry

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

477 **Table 16 – Class: CIM_LogEntry**

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

478 **10.5 CIM_RecordLog**

479 CIM_RecordLog represents the log in the managed system.

480

Table 17 – Class: CIM_RecordLog

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

481 **10.6 CIM_RegisteredProfile**

482 CIM_RegisteredProfile identifies the *Record Log Profile* in order for a client to determine the conformance
 483 with the profile. The CIM_RegisteredProfile class is defined by the *Profile Registration Profile*. With the
 484 exception of the mandatory values specified for the properties in Table 18, the behavior of the
 485 RegisteredProfile instance is per the *Profile Registration Profile*.

486

Table 18 – Class: CIM_RegisteredProfile

Elements	Requirement	Description
RegisteredName	Mandatory	This property shall have a value of “Record Log Profile”.
RegisteredVersion	Mandatory	This property shall have a value of “1.0.0”.
RegisteredOrganization	Mandatory	This property shall have a value of 2 (DMTF).

487 **10.7 CIM_UseOfLog**

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

490

Table 19 – Class: CIM_UseOfLog

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

491
492
493
494
495

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.

496
497
498
499
500

ANNEX B (informative)

Acknowledgements

501 The authors wish to acknowledge the following people.

502 Editor:

- 503 • Jon Hass – Dell
- 504 • Khachatur Papanyan – Dell

505 Contributors:

- 506 • Jon Hass – Dell
- 507 • Khachatur Papanyan – Dell
- 508 • Jeff Hilland – HP
- 509 • Christina Shaw – HP
- 510 • Aaron Merkin – IBM
- 511 • Jeff Lynch – IBM
- 512 • Perry Vincent – Intel
- 513 • John Leung – Intel