



1

2

3

4

Document Number: DSP1002

Date: 2009-09-23

Version: 1.0.1

5 **Diagnostics Profile**

6 **Document Type: Specification**

7 **Document Status: DMTF Standard**

8 **Document Language: E**

9 Copyright Notice

10 Copyright © 2009 Distributed Management Task Force, Inc. (DMTF). All rights reserved.

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

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25 withdrawn or modified after publication, and shall be indemnified and held harmless by any party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

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

31

CONTENTS

33	1	Scope	9
34	2	Normative References.....	9
35	2.1	Approved References	9
36	2.2	Other References.....	9
37	3	Terms and Definitions	9
38	4	Symbols and Abbreviated Terms	11
39	5	Synopsis.....	11
40	6	Description	12
41	7	Implementation.....	13
42	7.1	CIM_DiagnosticTest.....	14
43	7.1.1	CIM_DiagnosticTest.Name.....	14
44	7.1.2	CIM_DiagnosticTest.ElementName.....	14
45	7.1.3	CIM_DiagnosticTest.Characteristics.....	14
46	7.1.4	Looping Tests	15
47	7.1.5	Test Effectiveness.....	16
48	7.2	CIM_AvailableDiagnosticService.....	16
49	7.2.1	CIM_AvailableDiagnosticService.EstimatedDurationOfService	16
50	7.2.2	CIM_AvailableDiagnosticService.EstimatedDurationQualifier	16
51	7.3	CIM_DiagnosticServiceCapabilities.....	16
52	7.4	CIM_DiagnosticSetting	17
53	7.5	CIM_ConcreteJob	17
54	7.5.1	CIM_ConcreteJob.TimeBeforeRemoval.....	17
55	7.5.2	CIM_ConcreteJob.PercentComplete	17
56	7.6	CIM_DiagnosticsLog.....	17
57	7.6.1	Logging Results	17
58	7.7	CIM_DiagnosticRecord.....	18
59	7.7.1	CIM_DiagnosticRecord.ExpirationDate	18
60	7.8	CIM_ServiceComponent.....	18
61	8	Methods.....	19
62	8.1	CIM_DiagnosticService.RunDiagnostic().....	19
63	8.2	CIM_ConcreteJob.RequestStateChange().....	20
64	8.3	CIM_Log.ClearLog()	21
65	8.4	CIM_HelpService.GetHelp()	21
66	8.5	Profile Conventions for Operations.....	22
67	8.6	CIM_DiagnosticTest.....	22
68	8.7	CIM_AvailableDiagnosticService.....	23
69	8.8	CIM_ServiceAffectsElement	23
70	8.9	CIM_HelpService	23
71	8.10	CIM_ServiceAvailableToElement	23
72	8.11	CIM_DiagnosticSetting	24
73	8.12	CIM_DiagnosticServiceCapabilities.....	24
74	8.13	CIM_ElementCapabilities	24
75	8.14	CIM_ConcreteJob	25
76	8.15	CIM_OwningJobElement	25
77	8.16	CIM_AffectedJobElement	25
78	8.17	CIM_JobSettingData	25
79	8.18	CIM_DefaultSetting.....	26
80	8.19	CIM_DiagnosticsLog.....	26
81	8.20	CIM_UseOfLog	26
82	8.21	CIM_DiagnosticServiceRecord.....	27
83	8.22	CIM_DiagnosticSettingRecord.....	27
84	8.23	CIM_LogManagesRecord.....	27

85	8.24	CIM_RecordAppliesToElement	27
86	8.25	CIM_ServiceComponent.....	28
87	8.26	CIM_LogRecord.....	28
88	8.27	CIM_ElementSettingData	28
89	8.28	CIM_CorrespondingSettingsRecord	29
90	8.29	CIM_HostedService	29
91	9	Use Cases.....	29
92	9.1	Profile Conformance	29
93	9.2	Use Case Summary.....	30
94	9.3	Diagnostic Services Object Diagram	33
95	9.4	Discover Available Diagnostics.....	34
96	9.4.1	GetAllDiagnostics.....	34
97	9.4.2	GetAllDiagnosticMEPairs.....	34
98	9.4.3	GetDiagnosticsForME.....	34
99	9.4.4	GetMEsForDiagnostic.....	34
100	9.4.5	GetCapabilitiesOfDiagnostic.....	34
101	9.4.6	GetCharacteristicsOfDiagnostic.....	35
102	9.4.7	GetDiagnosticswithCharacteristicsForME	35
103	9.4.8	GetDiagnosticswithCapabilitiesForME.....	35
104	9.4.9	GetPackageSubtests	35
105	9.5	Configure Diagnostic.....	35
106	9.5.1	GetDefaultDiagnosticSettings.....	36
107	9.5.2	CreateDiagnosticSettings	36
108	9.5.3	GetDefaultJobSettings	36
109	9.5.4	CreateJobSettings	37
110	9.6	Execute and Control Diagnostic	37
111	9.6.1	RunDiagnostic.....	38
112	9.6.2	SuspendDiagnostic.....	38
113	9.6.3	ResumeDiagnostic.....	39
114	9.6.4	AbortDiagnostic.....	39
115	9.6.5	KillDiagnostic	39
116	9.7	Discover Diagnostic Executions	40
117	9.7.1	GetAffectedMEs.....	40
118	9.7.2	GetAllDiagnosticExecutionsForME.....	40
119	9.7.3	GetSpecificDiagnosticExecutions	40
120	9.7.4	GetSpecificDiagnosticExecutionsForME	41
121	9.8	Discover Diagnostic Results (In Progress and Final)	41
122	9.8.1	GetLogRecordsForDiagnostic	43
123	9.8.2	GetLogRecordsForME	44
124	9.8.3	GetLogRecordsForMEAndDiagnostic.....	44
125	9.8.4	GetDiagnosticExecutionResults	45
126	9.8.5	GetDiagnosticExecutionSettings	46
127	9.8.6	GetDiagnosticProgress.....	46
128	9.8.7	GetDiagnosticExecutionFinalResults.....	47
129	10	CIM Elements.....	47
130	10.1	CIM_AffectedJobElement	50
131	10.2	CIM_AvailableDiagnosticService.....	50
132	10.3	CIM_ConcreteJob	50
133	10.4	CIM_DiagnosticsLog.....	51
134	10.5	CIM_DiagnosticServiceCapabilities.....	51
135	10.6	CIM_DiagnosticServiceRecord.....	52
136	10.7	CIM_DiagnosticSetting	54
137	10.8	CIM_DiagnosticSettingRecord.....	55
138	10.9	CIM_DiagnosticTest.....	56
139	10.10	CIM_ElementCapabilities	57
140	10.11	CIM_DefaultSetting (DiagnosticSetting)	57
141	10.12	CIM_DefaultSetting (JobSettingData).....	58

142 10.13 CIM_HelpService 58
 143 10.14 CIM_JobSettingData 59
 144 10.15 CIM_LogManagesRecord 59
 145 10.16 CIM_OwningJobElement 59
 146 10.17 CIM_RecordAppliesToElement 60
 147 10.18 CIM_ServiceAffectsElement 60
 148 10.19 CIM_ServiceAvailableToElement 60
 149 10.20 CIM_ServiceComponent 61
 150 10.21 CIM_ElementSettingData 61
 151 10.22 CIM_UseOfLog 61
 152 10.23 CIM_CorrespondingSettingsRecord 62
 153 10.24 CIM_HostedService (DiagnosticTest) 62
 154 10.25 CIM_HostedService (HelpService) 62
 155 10.26 CIM_RegisteredProfile 63
 156 Bibliography 66

157

158 **Figures**

159 Figure 1 – Diagnostics Profile: Class Diagram 13
 160 Figure 2 – Registered Profile 30
 161 Figure 3 – Diagnostic Services Object Diagram 33
 162 Figure 4 – Job Example 38
 163 Figure 5 – Diagnostic Logging Object Diagram 43

164

165 **Tables**

166 Table 1 – Related Profiles 12
 167 Table 2 – RunDiagnostic() Method: Return Code Values 19
 168 Table 3 – RunDiagnostic() Method: Parameters 19
 169 Table 4 – RequestStateChange() Method: Return Code Values 20
 170 Table 5 – RequestStateChange() Method: Parameters 20
 171 Table 6 – ClearLog() Method: Return Code Values 21
 172 Table 7 – GetHelp() Method: Return Code Values 21
 173 Table 8 – GetHelp() Method: Parameters 22
 174 Table 9 – Operations: CIM_DiagnosticTest 22
 175 Table 10 – Operations: CIM_AvailableDiagnosticService 23
 176 Table 11 – Operations: CIM_ServiceAffectsElement 23
 177 Table 12 – Operations: CIM_HelpService 23
 178 Table 13 – Operations: CIM_ServiceAvailableToElement 23
 179 Table 14 – Operations: CIM_DiagnosticSetting 24
 180 Table 15 – Operations: CIM_DiagnosticServiceCapabilities 24
 181 Table 16 – Operations: CIM_ElementCapabilities 24
 182 Table 17 – Operations: CIM_ConcreteJob 25
 183 Table 18 – Operations: CIM_OwningJobElement 25
 184 Table 19 – Operations: CIM_AffectedJobElement 25
 185 Table 20 – Operations: CIM_JobSettingData 26
 186 Table 21 – Operations: CIM_DefaultSetting 26
 187 Table 22 – Operations: CIM_DiagnosticsLog 26

188	Table 23 – Operations: CIM_UseOfLog.....	26
189	Table 24 – Operations: CIM_DiagnosticServiceRecord	27
190	Table 25 – Operations: CIM_DiagnosticSettingRecord	27
191	Table 27 – Operations: CIM_RecordAppliesToElement.....	28
192	Table 28 – Operations: CIM_ServiceComponent	28
193	Table 29 – Operations: CIM_LogRecord	28
194	Table 30 – Operations: CIM_ElementSettingData.....	28
195	Table 31 – Operations: CIM_CorrespondingSettingsData	29
196	Table 32 – Operations: CIM_HostedService	29
197	Table 33 – Diagnostics Profile Use Cases.....	31
198	Table 34 – CIM Elements: Diagnostics Profile.....	48
199	Table 35 – Class: CIM_AffectedJobElement	50
200	Table 36 – Class: CIM_AvailableDiagnosticService.....	50
201	Table 37 – Class: CIM_ConcreteJob	50
202	Table 38 – Class: CIM_DiagnosticsLog.....	51
203	Table 39 – Class: CIM_DiagnosticServiceCapabilities	51
204	Table 40 – Class: CIM_DiagnosticServiceRecord	52
205	Table 41 – Class: CIM_DiagnosticSetting	54
206	Table 42 – Class: CIM_DiagnosticSettingRecord.....	55
207	Table 43 – Class: CIM_DiagnosticTest.....	56
208	Table 44 – Class: CIM_ElementCapabilities.....	57
209	Table 45 – Class: CIM_DefaultSetting (DiagnosticSetting)	57
210	Table 46 – Class: CIM_DefaultSetting (JobSettingData).....	58
211	Table 47 – Class: CIM_HelpService	58
212	Table 48 – Class: CIM_JobSettingData.....	59
213	Table 49 – Class: CIM_LogManagesRecord.....	59
214	Table 50 – Class: CIM_OwningJobElement	59
215	Table 51 – Class: CIM_RecordAppliesToElement	60
216	Table 52 – Class: CIM_ServiceAffectsElement	60
217	Table 53 – Class: CIM_ServiceAvailableToElement	60
218	Table 54 – Class: CIM_ServiceComponent.....	61
219	Table 55 – Class: CIM_ElementSettingData	61
220	Table 56 – Class: CIM_UseOfLog	61
221	Table 57 – Class: CIM_CorrespondingSettingsRecord	62
222	Table 58 – Class: CIM_HostedService (DiagnosticTest).....	62
223	Table 59 – Class: CIM_HostedService (HelpService).....	62
224	Table 60 – Class: CIM_RegisteredProfile.....	63
225		

226

Foreword

227 The *Diagnostics Profile* (DSP1002) was prepared by the Diagnostics Working Group.

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

230

Introduction

231 A *profile* is a collection of Common Information Model (CIM) elements and behavior rules that represent a
232 specific area of management. The purpose of a profile is to ensure interoperability in the use of web-
233 based enterprise management (WBEM) services for a specific subset of the Distributed Management
234 Task Force (DMTF) CIM schema for a specific management area—in this case, diagnostics.

235 Diagnostics is a critical component of systems management. Diagnostic services are used in problem
236 containment to maintain availability, achieve fault isolation for system recovery, establish system integrity
237 during boot, increase system reliability, and perform routine proactive system verification. The goal of the
238 Common Diagnostic Model (CDM) is to define industry-standard building blocks, based on and consistent
239 with the DMTF CIM, that enables seamless integration of vendor-supplied diagnostic services into system
240 and SAN management frameworks.

241 The CDM is an architecture and methodology for exposing system diagnostic instrumentation through the
242 CIM standard interfaces.

243 The ability to transparently run diagnostic tests and exercisers while the user operating system is
244 functional (no reboot required) may significantly contribute to the reduction of Total Cost of Ownership
245 (TCO) and will also lower warranty costs by reducing the return of defect-free parts for service. This
246 functionality is referred to as *OS-Present Diagnostics* (also known as On-line Diagnostics and Concurrent
247 Diagnostics).

248 A primary objective of the CDM is to standardize the interfaces that diagnostic developers create for their
249 OS-Present Diagnostics in the operating environment, making the diagnostics accessible to all
250 applications that query CIM for diagnostic data or register with CIM to execute diagnostic methods and
251 receive results.

252 Standardization of these interfaces means that clients, providers, and tests gain a certain degree of
253 portability and, in many cases, need only be written once to satisfy multiple environments and platforms.
254 OEMs can differentiate their diagnostic offerings by how effectively their applications use the information
255 and capabilities available through CIM to maintain and service their systems.

256 Reduced cost through standardization is accompanied by the initial investment of coding to a new
257 interface. The CDM Forum intends to ease this burden by developing tools to generate most of the
258 interface code necessary to communicate with CIM.

259

Diagnostics Profile

260 1 Scope

261 The information in this specification should be sufficient for a provider or consumer of this data to identify
262 unambiguously the classes, properties, methods, and values that shall be instantiated and manipulated to
263 represent and manage the diagnostic service components of systems and subsystems that are modeled
264 using the DMTF CIM core and extended model definitions.

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

267 2 Normative References

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

271 2.1 Approved References

272 DMTF DSP0200, *CIM Operations over HTTP 1.2.0*,
273 http://www.dmtf.org/standards/published_documents/DSP200.pdf

274 DMTF DSP0004, *CIM Infrastructure Specification 2.3.0*,
275 http://www.dmtf.org/standards/published_documents/DSP0004V2.3_final.pdf

276 DMTF DSP1001, *Management Profile Specification Usage Guide 1.0.0*,
277 http://www.dmtf.org/standards/published_documents/DSP1001.pdf

278 DMTF DSP1033, *Profile Registration Profile 1.0.0*,
279 http://www.dmtf.org/standards/published_documents/DSP1033_1.0.0.pdf

280 2.2 Other References

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

283 *Unified Modeling Language (UML) from the Open Management Group (OMG)*,
284 <http://www.omg.org/docs/formal/07-11-04.pdf>

285 3 Terms and Definitions

286 For the purposes of this document, the following terms and definitions apply. The terms and definitions
287 given in [DSP1033](#) and [DSP1001](#) also apply.

288 3.1

289 can

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

291 3.2

292 cannot

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

- 294 **3.3**
295 **conditional**
296 indicates requirements to be followed strictly in order to conform to the document when the specified
297 conditions are met
- 298 **3.4**
299 **mandatory**
300 indicates requirements to be followed strictly in order to conform to the document and from which no
301 deviation is permitted
- 302 **3.5**
303 **may**
304 indicates a course of action permissible within the limits of the document
- 305 **3.6**
306 **need not**
307 indicates a course of action permissible within the limits of the document
- 308 **3.7**
309 **optional**
310 indicates a course of action permissible within the limits of the document
- 311 **3.8**
312 **referencing profile**
313 indicates a profile that owns the definition of this class and can include a reference to this profile in its
314 "Related Profiles" table
- 315 **3.9**
316 **shall**
317 indicates requirements to be followed strictly in order to conform to the document and from which no
318 deviation is permitted
- 319 **3.10**
320 **shall not**
321 indicates requirements to be followed strictly in order to conform to the document and from which no
322 deviation is permitted
- 323 **3.11**
324 **should**
325 indicates that among several possibilities, one is recommended as particularly suitable, without
326 mentioning or excluding others, or that a certain course of action is preferred but not necessarily required
- 327 **3.12**
328 **should not**
329 indicates that a certain possibility or course of action is deprecated but not prohibited
- 330 **3.13**
331 **unspecified**
332 indicates that this profile does not define any constraints for the referenced CIM element or operation

333 4 Symbols and Abbreviated Terms

334 The following abbreviations are used in this document.

335 4.1

336 **CDM**

337 Common Diagnostic Model

338 4.2

339 **CIM**

340 Common Information Model

341 4.3

342 **CRU**

343 Customer Replaceable Unit

344 4.4

345 **FRU**

346 Field Replaceable Unit

347 4.5

348 **ME**

349 Managed Element

350 4.6

351 **MOF**

352 Managed Object Format

353 4.7

354 **PD**

355 Problem Determination

356 4.8

357 **PFA**

358 Predictive Failure Analysis

359 4.9

360 **SAN**

361 Storage Area Network

362 4.10

363 **WBEM**

364 Web-Based Enterprise Management

365 5 Synopsis

366 **Profile Name:** Diagnostics

367 **Version:** 1.0.1

368 **Organization:** DMTF

369 **CIM schema version:** 2.9.0

370 **Central Class:** CIM_DiagnosticTest

371 **Scoping Class:** CIM_ComputerSystem

372 The *Diagnostics Profile* extends the management capability of referencing profiles by adding the
 373 capability to run diagnostic services in a managed system. This profile includes a specification of the
 374 Diagnostic Test Service, its configuration, its associated capabilities, its logging mechanisms, and its
 375 profile registration information.

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

377 CIM_DiagnosticTest shall be the Central Class of this profile. The instance of CIM_DiagnosticTest shall
 378 be the Central Instance of this profile. CIM_ComputerSystem shall be the Scoping Class of this profile.
 379 The instance of CIM_ComputerSystem with which the Central Instance is associated through an instance
 380 of CIM_HostedService shall be the Scoping Instance of this profile.

381

Table 1 – Related Profiles

Profile Name	Organization	Version	Relationship	Behavior
Profile Registration	DMTF	1.0	Mandatory	

382 6 Description

383 This profile describes the CIM schema extensions that compose the Common Diagnostic Model (CDM)
 384 and provides guidelines for the development of diagnostic clients and providers that will promote
 385 seamless integration of option diagnostics into Problem Determination and Systems Management
 386 applications. Using this profile as a guide, WBEM clients can discover diagnostic services that have been
 387 installed on the system and invoke these services to run on their respective devices. The client can
 388 monitor the progress of the service, obtain and modify the status of the service, and query for results.

389 The architecture of the CDM is described in the [CIM Diagnostic Model White Paper](#). This profile is a
 390 normative presentation of the model described in the white paper, and it suggests implementation
 391 techniques that will result in the highest degree of interoperability. It is targeted at developers of
 392 diagnostic applications (WBEM clients) and hardware instrumentation (for the WBEM server) to help them
 393 understand the CDM.

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

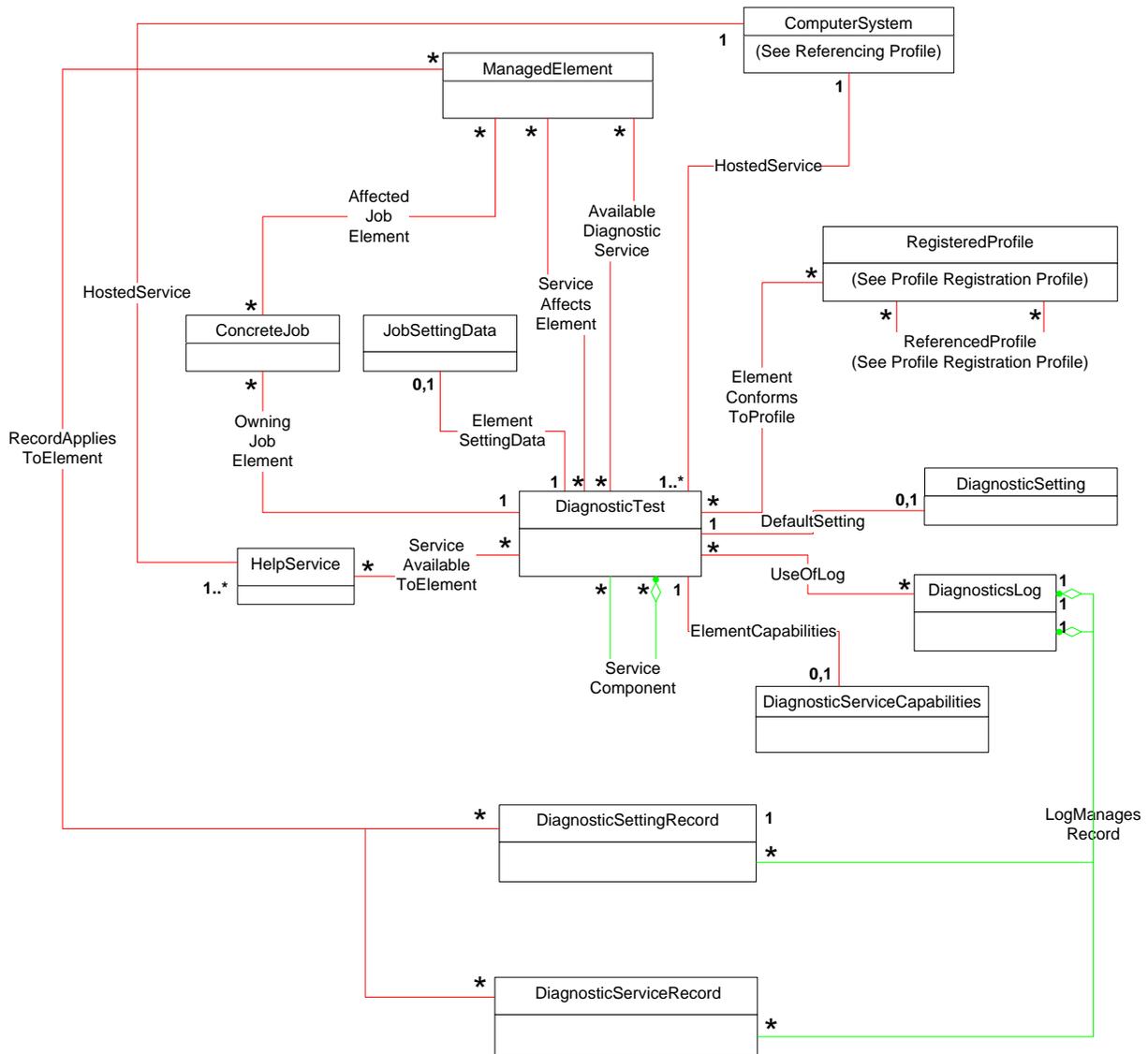


Figure 1 – Diagnostics Profile: Class Diagram

396

397

398 7 Implementation

399 This section details the requirements related to the arrangement of instances and their properties for
 400 implementations of this profile.

401 The *Diagnostics Profile* consists of definitions for classes related to the CIM_DiagnosticService class,
 402 such as CIM_DiagnosticTest, CIM_DiagnosticSetting, and CIM_DiagnosticServiceCapabilities. It also
 403 defines the CIM_DiagnosticsLog class and its related classes, CIM_DiagnosticRecord,
 404 CIM_DiagnosticServiceRecord, and CIM_DiagnosticSettingRecord. Requirements for propagating and
 405 formulating certain properties of these classes and their parents are discussed in this section. Required
 406 methods are listed in section 8, and properties are listed in section 10.

407 **7.1 CIM_DiagnosticTest**

408 CIM_DiagnosticTest is the only defined subclass of CIM_DiagnosticService. CIM_DiagnosticTest inherits
409 the RunDiagnostic() method, which is used to execute a diagnostic test on a managed element.

410 Each diagnostic test shall be represented by an instance of either CIM_DiagnosticTest or a subclass.
411 Note that a test that actually packages multiple subtests shall also be represented by such an instance
412 and shall set the IsPackage characteristic for that instance (see section 7.1.3.5).

413 Depending on the implementation, a provider may use

- 414 • an instance of CIM_DiagnosticTest for each test
- 415 • an instance of a single subclass (for example, ST_DiskDiagnosticTest) for each test
- 416 • a different subclass and its instance (for example, ST_DiskDiagnosticSelfTest,
417 ST_DiskDiagnosticRWVTest) for each test

418 The same provider may use a combination of the preceding approaches.

419 **7.1.1 CIM_DiagnosticTest.Name**

420 The Name property uniquely identifies the service and provides an indication of the functionality that is
421 managed. The value of the Name property shall be unique and should indicate the nature of the service
422 (for example, EjectTest).

423 **7.1.2 CIM_DiagnosticTest.ElementName**

424 The ElementName property shall be used to provide a user-friendly name for the service. This name shall
425 be used by clients to identify the service to the user.

426 **7.1.3 CIM_DiagnosticTest.Characteristics**

427 This section defines the values of the Characteristics property.

428 **7.1.3.1 2 (Is Exclusive)**

429 Use this value to indicate that only one instance of the diagnostic test may be running at one time, even if
430 more than one target device exists.

431 **7.1.3.2 3 (Is Interactive)**

432 Use this value to indicate that the test requires some interaction with the client at the system under test
433 (for example, when media is required in a device for the test to run).

434 **7.1.3.3 4 (Is Destructive)**

435 Use this value to indicate that the test has the potential for destroying data, permanently altering the
436 state, or reconfiguring the device.

437 **7.1.3.4 5 (Is Risky)**

438 Use this value to indicate that data loss, state change, or reconfiguration may occur if the test is
439 interrupted. For example, a test saves some device data or configuration, changes the device state,
440 performs some operation, and then restores the saved data. If this process is interrupted, the device may
441 be left in an altered state.

442 7.1.3.5 6 (Is Package)

443 Use this value to indicate that the test is actually a set of lower-level diagnostics that are packaged
444 together by the test. This packaging is implemented by the test, not aggregated by CIM. Information and
445 results associated with the individual tests in the package may be requested by using the Subtests value
446 in the CIM_DiagnosticSetting.LogOptions array.

447 If the lower-level diagnostics are themselves CIM_DiagnosticTest instances, the packaging test shall be
448 associated to those lower-level diagnostics through an instance of the CIM_ServiceComponent
449 association. See section 7.8.

450 7.1.3.6 7 (Reserved)

451 This value originally contained "Supports PercentOfTestCoverage", which was deprecated and added to
452 the CIM_DiagnosticServiceCapabilities class.

453 7.1.3.7 8 (Is Synchronous)

454 Use this value to indicate that this diagnostic service will complete before the RunDiagnostic() method
455 returns to the caller. A job is still created that the client may access for accounting purposes, but the
456 ability to track the progress and status of the job are lost. Additionally, in certain environments, the client
457 may be "blocked" from further action until the service completes. Development of synchronous diagnostic
458 services is not recommended.

459 7.1.3.8 9 (Media Required)

460 Use this value to indicate that media shall be inserted into the device to perform the service.

461 7.1.3.9 10 (Additional Hardware Required)

462 Use this value to indicate that some additional hardware (for example, a wrap plug) shall be installed to
463 perform the service.

464 7.1.4 Looping Tests

465 Looping tests or groups of tests is useful for detecting intermittent faults. The client, provider, or test may
466 control looping, and the method chosen depends on many factors, a few of which follow:

- 467 • A client may want to loop a test that does not support looping.
- 468 • A provider may choose to support looping even though its tests do not.
- 469 • A stress test may, by its nature, want to repeat a certain operation a large number of times.

470 Looping in the provider and test is under control of the LoopControl() and LoopControlParameter()
471 properties of the CIM_DiagnosticSetting class. These properties are used to specify the number of
472 iterations in the loop, either directly or through a termination condition. If more than one control is set, the
473 first one that reaches its condition terminates the loop.

474 Looping in the client is entirely under the control of the client and would generally not affect the
475 CIM_DiagnosticSetting object.

476 **Note:** A remote client may incur network delays and CIM Server delays during every iteration of its loop,
477 and this is not an effective way to stress a device.

478 It is recommended that all diagnostic tests support looping. Exceptions exist where looping a test leads to
479 an undesirable condition (for example, a risky test, certain user interactions, or excessive mechanical
480 wear).

481 **7.1.5 Test Effectiveness**

482 Although the focus of this profile is use of the CIM schema, the CDM includes the notion of test
483 effectiveness. A perfectly implemented CDM provider wrapped around an ineffective test is not very
484 useful.

485 Diagnostic tests should provide support for all properties in the CIM_DiagnosticSetting class.

486 The QuickMode property of the CIM_DiagnosticSetting class shall be supported for “long-running” tests
487 (that is, tests with running times in excess of what would be considered compatible with a quick system
488 “health check” of a few minutes). QuickMode need not be supported for interactive, risky, or destructive
489 tests, because these tests would not be useful as a health check.

490 **Note:** QuickMode is distinct from PercentOfTestCoverage in that it is a Boolean property that may be set
491 by a client without any particular knowledge of the test. Use of PercentOfTestCoverage requires that the
492 client be aware of the effects and expected outcome of this “throttling” setting control. Diagnostic tests
493 should support the ability to surface logs that may be useful in the problem-determination process.

494 **7.2 CIM_AvailableDiagnosticService**

495 An instance of CIM_AvailableDiagnosticService shall associate a managed element with a diagnostic
496 service that is available for that element. This instance is the means by which clients discover the
497 diagnostic services that are installed for a particular managed element.

498 **7.2.1 CIM_AvailableDiagnosticService.EstimatedDurationOfService**

499 All tests shall attempt to accurately set the EstimatedDurationOfService property. As stated in the MOF
500 file for this class, this property is an estimation of magnitude, not absolute time, and is to be used as a
501 guide for the client.

502 The CIM_DiagnosticSetting.LoopControl property allows a client to indicate how long a test should run.
503 Tests should use their default values for the LoopControl properties when determining a value for
504 EstimatedDurationOfService.

505 Interactive tests have an additional complication because their test execution depends on the responses
506 from the user. However, this type of test is not much different than a test whose execution depends on
507 information from a device and the response time of the hardware, or even on how much CPU time or
508 other system resources are allocated to the test. Interactive tests should assume a user response time. If
509 a test cannot reasonably determine an EstimatedDurationOfService value (for example, a completely
510 interactive test that does not know anything about what it will do until a user tells it what tests to run), it
511 can set the value to 0 (Unknown).

512 **7.2.2 CIM_AvailableDiagnosticService.EstimatedDurationQualifier**

513 The EstimatedDurationQualifier property allows for more accurate quantification of the value specified for
514 the EstimatedDurationOfService property. This property shall be implemented only if further quantification
515 is possible.

516 **7.3 CIM_DiagnosticServiceCapabilities**

517 CIM_DiagnosticServiceCapabilities is the means by which a diagnostic service may publish its support for
518 various options—in particular, settings. If a setting is supported, the client may assign it, usually in
519 satisfaction of a user request. The client gains access to an instance of
520 CIM_DiagnosticServiceCapabilities through an instance of CIM_ElementCapabilities.

521 **7.4 CIM_DiagnosticSetting**

522 This class defines specific diagnostic service parameters and execution instructions. To provide more
523 detailed settings for a type of test (that is, additional properties), subclassing is appropriate.

524 The default settings for a diagnostic service are obtained by using the CIM_DefaultSetting association to
525 an instance of (a subclass of) CIM_DiagnosticSetting. If a service does not publish defaults in this
526 manner, the client should either avoid settings altogether or use only those settings supported by an
527 instance of CIM_DiagnosticServiceCapabilities.

528 Note that the CIM_DiagnosticSetting subclass may have extensions. If the client is aware of the
529 extensions, these may be modified as well. If the client is unaware, the default values should be used.

530 If a client chooses to accept the default settings (published or not), the CIM_DiagnosticSetting object may
531 be excluded from the method parameter list (entered as NULL).

532 **7.5 CIM_ConcreteJob**

533 This section defines the properties of the CIM_ConcreteJob class. All executing diagnostics will be
534 represented by instances of CIM_ConcreteJob so that a client can track the progress and control the
535 execution of the executing diagnostic.

536 **7.5.1 CIM_ConcreteJob.TimeBeforeRemoval**

537 To properly implement the functionality implied by this property, the job completion time shall be
538 determined. The algorithm is

539 If JobState=Completed OR Terminated OR Killed OR Exception OR ShuttingDown, then Completion
540 Time=StartTime+ElapsedTime.

541 The job may be deleted at Completion Time+TimeBeforeRemoval.

542 **7.5.2 CIM_ConcreteJob.PercentComplete**

543 This property indicates the percentage of the job that has completed at the time that this value is
544 requested.

545 Implementation of this property is mandatory in order to provide progress indication to clients.

546 The value of this property shall be kept current to be useful. Service providers should update this property
547 within one second of becoming aware of a progress change.

548 The PercentComplete property shall always report the actual percent complete of how much testing was
549 done. It shall be set to 100 percent only when the test is complete. It shall not be set to 100 percent if the
550 test stops for any other reason (for example, the test stopped or was killed by user, the test exited due to
551 a critical failure, or the test found an error and HaltOnError is TRUE) because the actual percent complete
552 is not 100 percent.

553 **7.6 CIM_DiagnosticsLog**

554 All diagnostic result messages may be represented by instances of CIM_DiagnosticRecord subclasses.
555 Moreover, those records may be aggregated to an instance of CIM_DiagnosticsLog. A diagnostic service
556 may also implement other additional logging mechanisms. Any other implemented logging mechanism
557 shall be indicated in the LogStorage property of the published capabilities.

558 **7.6.1 Logging Results**

559 The ways to record the results of running a diagnostic service are specified by the LogOptions and
560 LogStorage properties of the CIM_DiagnosticsSetting class. Use LogOptions to specify *what* to log and

561 LogStorage to specify *where* to log it. The MOF file describes these properties in some detail, but it is
562 useful to emphasize the mandatory mechanism here.

563 *Diagnostic Records aggregated to the Diagnostic Log* are highly recommended for several reasons:

- 564 • The heterogeneous nature of the log entries more easily fits into a self-describing record
565 paradigm.
- 566 • Keyed records are easier to manage and retrieve.

567 **7.7 CIM_DiagnosticRecord**

568 CIM_DiagnosticRecord has two subclasses: CIM_DiagnosticServiceRecord and
569 CIM_DiagnosticSettingRecord.

570 CIM_DiagnosticServiceRecord is structured to hold the information that is generated while a particular
571 service is running.

572 CIM_DiagnosticSettingRecord is structured to hold the attributes of the setting object that was used as an
573 input parameter to the RunDiagnostic() method.

574 **7.7.1 CIM_DiagnosticRecord.ExpirationDate**

575 After a diagnostic service produces results, the result objects need to persist for a minimum amount of
576 time to allow diagnostic CIM clients to capture what the application needs. When the data has been
577 captured, the containing objects need to be deleted in a timely fashion.

578 CIM_DiagnosticSetting.ResultPersistence shall be used by the client to specify to the diagnostic service
579 provider how long the results generated by that service shall persist. A value shall be chosen that allows
580 the minimum time needed by the client to record the data. When the timeout value has been reached, the
581 provider shall expire the data objects that contain the results.

582 The value of CIM_DiagnosticRecord.ExpirationDate shall be calculated by the provider to account for the
583 persistence setting value, time zone, and other applicable factors. When this expiration value has been
584 reached, the record is eligible for immediate deletion by the provider. It is the provider's responsibility to
585 manage the logs to prevent accumulation of expired records.

586 A ResultPersistence value of 0 (zero) indicates that the result does not need to persist; the
587 ExpirationDate is set to the current date and time. A ResultPersistence value of 0xFFFFFFFF indicates
588 that the result shall persist until it is explicitly deleted by a client DeleteInstance or ClearLog call; the
589 ExpirationDate is set to NULL, indicating no expiration date.

590 **7.8 CIM_ServiceComponent**

591 CIM_ServiceComponent is the means by which clients discover any individual tests that are also subtests
592 within a packaging test. This association does not imply any order, number, or method of subtest
593 execution, nor that all subtests executed within a packaging test shall be individual tests, nor even that all
594 the subtests would be executed for any specific execution of the packaging test.

595 The packaging test shall ensure that the values in CIM_DiagnosticTest.Characteristics of the packaging
596 test are consistent with the values in CIM_DiagnosticTest.Characteristics of the subtests unless the
597 packaging test can execute the subtest such that it does not have those characteristics. For example, if a
598 subtest sets the values Is Destructive or Is Interactive, the packaging test values in
599 CIM_DiagnosticTest.Characteristics should reflect those same characteristics, unless the packaging test
600 can execute the subtest so that it is not destructive or interactive.

601 **8 Methods**

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

604 **8.1 CIM_DiagnosticService.RunDiagnostic()**

605 The RunDiagnostic() method is invoked to commence execution of a diagnostic service on a specific
606 managed element. The input parameters specify this managed element and the settings that are to be
607 applied to the diagnostic service and the resultant job. The method returns a reference to the
608 CIM_ConcreteJob instance that is created.

609 Before invoking this method, clients examine the appropriate capabilities and create valid
610 CIM_DiagnosticSetting and CIM_JobSettingData instances to apply as input parameters. The
611 RunDiagnostic() method shall capture the attributes of CIM_DiagnosticSetting in an instance of
612 CIM_DiagnosticSettingRecord. This information is useful for post-mortem analysis of diagnostic results.

613 A job may be instantiated to monitor the diagnostic service as it runs and to provide useful accounting
614 and status information when the diagnostic service has completed.

615 RunDiagnostic() return values are specified in Table 2 and parameters are specified in Table 3. No
616 standard messages are defined.

617 **Table 2 – RunDiagnostic() Method: Return Code Values**

Value	Description
0	Job completed with no error
1	Not supported
2	Unknown
3	Timeout
4	Failed
5	Invalid parameter
0x1000	Method parameters checked – job started
0x1001..0x7FFF	Method reserved
0x8000..0xFFFF	Vendor Specific

618 **Table 3 – RunDiagnostic() Method: Parameters**

Qualifiers	Name	Type	Description/Values
IN	ManagedElement	CIM_ManagedElement	A reference that specifies the element upon which to run the diagnostic service This parameter is Mandatory.
IN	DiagSetting	CIM_DiagnosticSetting	A reference that specifies the settings to be applied to the diagnostic service. If NULL, the diagnostic service's defaults are used.
IN	JobSetting	CIM_JobSettingData	A reference that specifies the settings to be applied to the resulting job. If NULL, the job's defaults are used.
OUT	Job	CIM_ConcreteJob	Returns a reference to the resulting job

619 8.2 CIM_ConcreteJob.RequestStateChange()

620 All CIM_DiagnosticService.RunDiagnostic() calls will return a reference to a CIM_ConcreteJob instance,
 621 which represents the diagnostic execution. The CIM_ConcreteJob.RequestStateChange() method is
 622 invoked to control the diagnostic program execution. The input parameters specify the execution control
 623 to be performed (Suspend, Kill, Terminate) and a timeout period that specifies the maximum amount of
 624 time that the client expects the transition to the new state to take.

625 Before invoking this method, clients examine the appropriate capabilities to verify whether the execution
 626 control is supported. The RequestStateChange() method shall change the JobState value if the transition
 627 is successfully performed.

628 RequestStateChange() return values are specified in Table 4 and parameters are specified in Table 5.
 629 No standard messages are defined.

630 **Table 4 – RequestStateChange() Method: Return Code Values**

Value	Description
0	Completed with No Error
1	Not Supported
2	Unknown/Unspecified Error
3	Can NOT complete within Timeout Period
4	Failed
5	Invalid parameter
6	In Use
7..4095	DMTF reserved
4096	Method parameters checked – transition started
4097	Invalid state transition
4098	Use of timeout parameter not supported
4099	Busy
4100 - 32767	Method reserved
32768-65535	Vendor specific

631 **Table 5 – RequestStateChange() Method: Parameters**

Qualifiers	Name	Type	Description/Values
IN	RequestedState	uint16	The requested state of a job, which may be one of the following values: Start (2), Suspend (3), Terminate (4), Kill (5), or Service (6)
IN	TimeoutPeriod	datetime	A timeout period that specifies the maximum amount of time that the client expects the transition to the new state to take. The interval format shall be used to specify the TimeoutPeriod.

632 **8.3 CIM_Log.ClearLog()**

633 The ClearLog() method is invoked to delete all records (instances of CIM_DiagnosticRecord subclasses)
 634 that are associated with the log instance through the CIM_LogManagesRecord association. This method
 635 has no parameters, and no standard messages are defined.

636 ClearLog return values are specified in Table 6.

637 **Table 6 – ClearLog() Method: Return Code Values**

Value	Description
0	Completed with no error
1	Not supported
2	Unspecified Error
3	Timeout
4	Failed
5	Invalid parameter
“6..0x0FFF”	DMTF reserved
0x1000..0x7FFF	Method_Reserved
0x8000..0xFFFF	Vendor_Reserved

638 **8.4 CIM_HelpService.GetHelp()**

639 The GetHelp() method is invoked to obtain documentation about a diagnostic service. The input
 640 parameters provide the name, format, and delivery type of a document.

641 The CIM_HelpService class has some attributes that publish the available documents, supported delivery
 642 types, and formats. See Table 8 for additional information. Before invoking this method, clients check
 643 these attributes in order to request an available document, format, and delivery type.

644 GetHelp() return values are specified in Table 7 and parameters are specified in Table 8. No standard
 645 messages are defined.

646 **Table 7 – GetHelp() Method: Return Code Values**

Value	Description
0	Document returned with no error
1	Not supported
2	Unspecified Error
3	Timeout
4	Failed
5	Invalid parameter
6..0x0FFF	DMTF reserved
0x1000	Busy
0x1001	Requested Document not found
0x1002..0x7FFF	Method Reserved
0x8000..0xFFFF	Vendor Reserved

647

Table 8 – GetHelp() Method: Parameters

Qualifiers	Name	Type	Description/Values
IN	RequestedDocument	string	The document that should be made available to the client. The available documents are published in the DocumentsAvailable attribute.
IN	Format	uint16	The format that the document should have. The supported formats are published in the DocumentFormat attribute.
IN	RequestedDelivery	uint16	The way in which the document should be made available (fully specified path, launch a program, file contents, and so on)
OUT	DocumentInfo	string	This parameter returns information about the document. The format and content will depend on the RequestedDelivery parameter.

648 8.5 Profile Conventions for Operations

649 Support for operations for each Profile class (including associations) is specified in the following
 650 subclauses. Each subclause includes either the statement “All operations in the default list in section 8.5
 651 are supported as described by [DSP0200](#) version 1.2” or a table listing all of the operations that are not
 652 supported by this Profile or where the Profile requires behavior other than that described by [DSP0200](#)
 653 version 1.2.

654 The default list of operations is as follows:

- 655 • GetInstance
- 656 • Associators
- 657 • AssociatorNames
- 658 • References
- 659 • ReferenceNames
- 660 • EnumerateInstances
- 661 • EnumerateInstanceNames

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

664 8.6 CIM_DiagnosticTest

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

667

Table 9 – Operations: CIM_DiagnosticTest

Operation	Requirement	Messages
References	Unspecified	None
ReferenceNames	Unspecified	None

668 **8.7 CIM_AvailableDiagnosticService**

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

671 **Table 10 – Operations: CIM_AvailableDiagnosticService**

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

672 **8.8 CIM_ServiceAffectsElement**

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

675 **Table 11 – Operations: CIM_ServiceAffectsElement**

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

676 **8.9 CIM_HelpService**

677 Table 12 lists operations that either have special requirements beyond those from [DSP0200](#) version 1.2
678 or shall not be supported.

679 **Table 12 – Operations: CIM_HelpService**

Operation	Requirement	Messages
References	Unspecified	None
ReferenceNames	Unspecified	None

680 **8.10 CIM_ServiceAvailableToElement**

681 Table 13 lists operations that either have special requirements beyond those from [DSP0200](#) version 1.2
682 or shall not be supported.

683 **Table 13 – Operations: CIM_ServiceAvailableToElement**

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

684 8.11 CIM_DiagnosticSetting

685 Table 14 lists operations that either have special requirements beyond those from [DSP0200](#) version 1.2
686 or shall not be supported.

687 CreateInstance, DeleteInstance and ModifyInstance shall be supported if the provider supports the
688 DiagnosticServiceCapabilities class and the DiagnosticServiceCapabilities class indicates that settings
689 other than default are supported. DeleteInstance and ModifyInstance operations shall return
690 CIM_ERR_ACCESS_DENIED if the DiagnosticSetting instance is associated to the DiagnosticTest
691 instance via the DefaultSetting association.

692 **Table 14 – Operations: CIM_DiagnosticSetting**

Operation	Requirement	Messages
CreateInstance	Conditional	None
ModifyInstance	Conditional	None
DeleteInstance	Conditional	None
References	Unspecified	None
ReferenceNames	Unspecified	None

693 8.12 CIM_DiagnosticServiceCapabilities

694 Table 15 lists operations that either have special requirements beyond those from [DSP0200](#) version 1.2
695 or shall not be supported.

696 **Table 15 – Operations: CIM_DiagnosticServiceCapabilities**

Operation	Requirement	Messages
References	Unspecified	None
ReferenceNames	Unspecified	None

697 8.13 CIM_ElementCapabilities

698 Table 16 lists operations that either have special requirements beyond those from [DSP0200](#) version 1.2
699 or shall not be supported.

700 **Table 16 – Operations: CIM_ElementCapabilities**

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

701 8.14 CIM_ConcreteJob

702 Table 17 lists operations that either have special requirements beyond those from [DSP0200](#) version 1.2
703 or shall not be supported.

704 **Table 17 – Operations: CIM_ConcreteJob**

Operation	Requirement	Messages
ModifyInstance	Optional	None
References	Unspecified	None
ReferenceNames	Unspecified	None

705 8.15 CIM_OwningJobElement

706 Table 18 lists operations that either have special requirements beyond those from [DSP0200](#) version 1.2
707 or shall not be supported.

708 **Table 18 – Operations: CIM_OwningJobElement**

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

709 8.16 CIM_AffectedJobElement

710 Table 19 lists operations that either have special requirements beyond those from [DSP0200](#) version 1.2
711 or shall not be supported.

712 **Table 19 – Operations: CIM_AffectedJobElement**

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

713 8.17 CIM_JobSettingData

714 Table 20 lists operations that either have special requirements beyond those from [DSP0200](#) version 1.2
715 or shall not be supported.

716 CreateInstance, DeleteInstance and ModifyInstance shall be supported if the provider supports job
717 settings other than default. DeleteInstance and ModifyInstance operations shall return
718 CIM_ERR_ACCESS_DENIED if the JobSettingData instance is associated to the DiagnosticTest instance
719 via the ElementSettingData association where ElementSettingData.IsDefault property is True.

720

Table 20 – Operations: CIM_JobSettingData

Operation	Requirement	Messages
CreateInstance	Conditional	None
ModifyInstance	Conditional	None
DeleteInstance	Conditional	None
References	Unspecified	None
ReferenceNames	Unspecified	None

721 8.18 CIM_DefaultSetting

722 Table 21 lists operations that either have special requirements beyond those from [DSP0200](#) version 1.2
723 or shall not be supported.

724

Table 21 – Operations: CIM_DefaultSetting

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

725 8.19 CIM_DiagnosticsLog

726 Table 22 lists operations that either have special requirements beyond those from [DSP0200](#) version 1.2
727 or shall not be supported.

728

Table 22 – Operations: CIM_DiagnosticsLog

Operation	Requirement	Messages
References	Unspecified	None
ReferenceNames	Unspecified	None

729 8.20 CIM_UseOfLog

730 Table 23 lists operations that either have special requirements beyond those from [DSP0200](#) version 1.2
731 or shall not be supported.

732

Table 23 – Operations: CIM_UseOfLog

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

733 **8.21 CIM_DiagnosticServiceRecord**

734 Table 24 lists operations that either have special requirements beyond those from [DSP0200](#) version 1.2
735 or shall not be supported.

736 **Table 24 – Operations: CIM_DiagnosticServiceRecord**

Operation	Requirement	Messages
CreateInstance	Optional	None
DeleteInstance	Mandatory	None
References	Unspecified	None
ReferenceNames	Unspecified	None

737 **8.22 CIM_DiagnosticSettingRecord**

738 Table 25 lists operations that either have special requirements beyond those from [DSP0200](#) version 1.2
739 or shall not be supported.

740 **Table 25 – Operations: CIM_DiagnosticSettingRecord**

Operation	Requirement	Messages
CreateInstance	Optional	None
DeleteInstance	Mandatory	None
References	Unspecified	None
ReferenceNames	Unspecified	None

741 **8.23 CIM_LogManagesRecord**

742 Table 26 lists operations that either have special requirements beyond those from [DSP0200](#) version 1.2
743 or shall not be supported.

744 **Table 26 – Operations: CIM_LogManagesRecord**

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

745 **8.24 CIM_RecordAppliesToElement**

746 Table 27 lists operations that either have special requirements beyond those from [DSP0200](#) version 1.2
747 or shall not be supported.

748

Table 27 – Operations: CIM_RecordAppliesToElement

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

749 **8.25 CIM_ServiceComponent**

750 Table 28 lists operations that either have special requirements beyond those from [DSP0200](#) version 1.2
751 or shall not be supported.

752

Table 28 – Operations: CIM_ServiceComponent

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

753 **8.26 CIM_LogRecord**

754 Table 29 lists operations that either have special requirements beyond those from [DSP0200](#) version 1.2
755 or shall not be supported.

756

Table 29 – Operations: CIM_LogRecord

Operation	Requirement	Messages
CreateInstance	Optional	None
DeleteInstance	Mandatory	None
References	Unspecified	None
ReferenceNames	Unspecified	None

757 **8.27 CIM_ElementSettingData**

758 Table 30 lists operations that either have special requirements beyond those from [DSP0200](#) version 1.2
759 or shall not be supported.

760

Table 30 – Operations: CIM_ElementSettingData

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

761 **8.28 CIM_CorrespondingSettingsRecord**

762 Table 31 lists operations that either have special requirements beyond those from [DSP0200](#) version 1.2
763 or shall not be supported.

764 **Table 31 – Operations: CIM_CorrespondingSettingsData**

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

765 **8.29 CIM_HostedService**

766 Table 32 lists operations that either have special requirements beyond those from [DSP0200](#) version 1.2
767 or shall not be supported.

768 **Table 32 – Operations: CIM_HostedService**

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

769 **9 Use Cases**

770 This section contains object diagrams and use cases for the *Diagnostics Profile*.

771 **9.1 Profile Conformance**

772 Conformance of a central class instance and its associated instances to a particular profile may be
773 identified by examining instances of the CIM_ElementConformsToProfile association class according to
774 the Central Class Methodology. In some environments, an alternative method that relies on the Scoping
775 Class Methodology through the scoping class instance may be desirable.

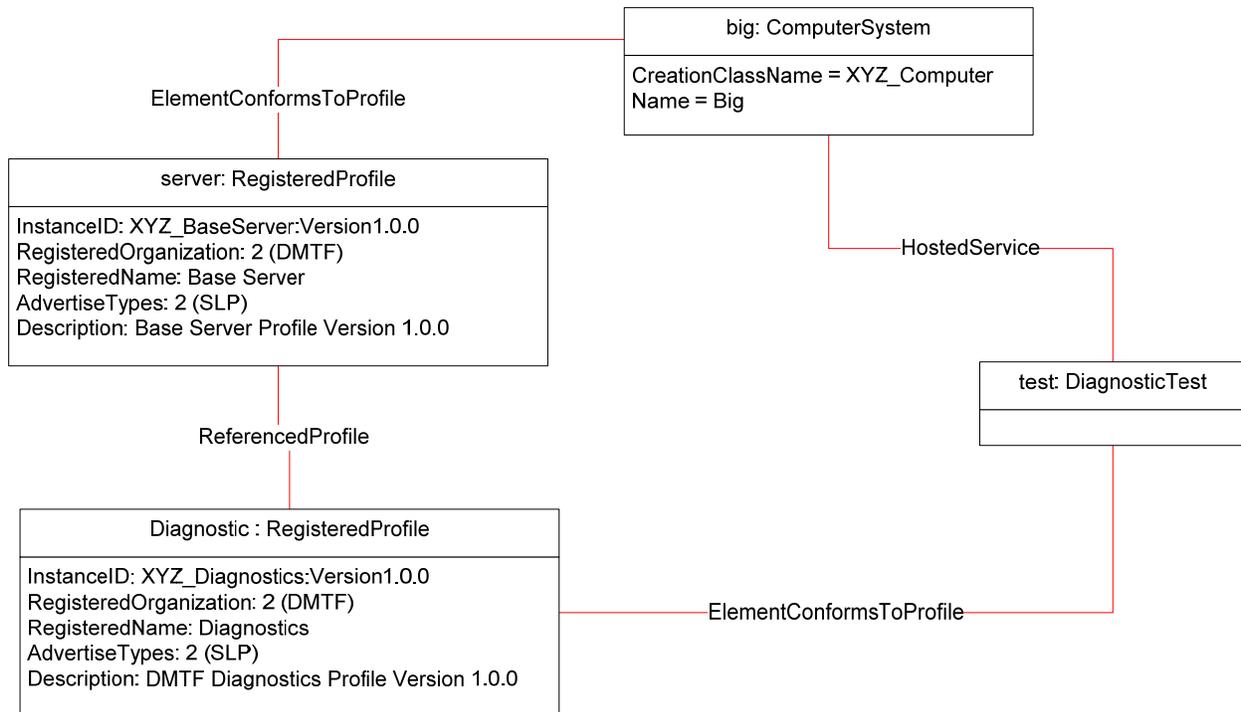
776 With CIM_ComputerSystem as the Scoping Class of this profile, the object diagram in Figure 2 shows
777 how instances of CIM_RegisteredProfile may be used to identify the version of the *Diagnostics Profile* to
778 which an instance of CIM_DiagnosticTest and its associated instances conform. In this example (using
779 BaseServer as the system configuration), one instance of CIM_RegisteredProfile identifies the “Base
780 Server Profile v1.0” and the other instance identifies the “*Diagnostics Profile v1.0*.”

781 To support the Scoping Class Methodology for advertising profile implementation conformance, a
782 CIM_DiagnosticTest instance is associated to an instance of the Scoping Class, CIM_ComputerSystem,
783 through an instance of CIM_HostedService. This instance of CIM_ComputerSystem is advertised as
784 being in implementation conformance with the *Base Server Profile v1.0* as indicated by the
785 CIM_ElementConformsToProfile association to the "server" CIM_RegisteredProfile instance. The
786 CIM_ReferencedProfile relationship between "server" and "diagnostic" places the CIM_DiagnosticTest
787 instance within the scope of "diagnostic." Thus, the CIM_DiagnosticTest instance is conformant with the
788 *Diagnostics Profile v1.0*.

789 To support the Central Class Methodology for advertising profile implementation conformance, a
 790 CIM_ElementConformsToProfile association is established between the CIM_DiagnosticTest central
 791 class instance and the instance of CIM_RegisteredProfile that represents the *Diagnostics Profile*.

792 For these methodologies to be successful, profiles for systems that can support diagnostics need to
 793 reference the *Diagnostics Profile*. In this example, the *Base Server Profile* would need to include the
 794 *Diagnostics Profile* in its “Related Profiles” table.

795 The CIM_ prefix has been omitted from the class names in Figure 2 for simplicity and readability.



796
797

798 **Figure 2 – Registered Profile**

799 **9.2 Use Case Summary**

800 Table 33 summarizes the use cases that are described in this section. The use cases are categorized
 801 and named, and references are provided to the body text that describes the use case.

802 **Note:** Although use case names follow the convention for naming classes, properties, and methods in the
 803 schema, this naming was done for readability only and does not imply any functionality attached to the
 804 name.

805 The CIM_ prefix has been omitted from the class names in the use cases for readability.

Table 33 – Diagnostics Profile Use Cases

Category	Name	Description
Discover Available Diagnostics See section 9.4.	GetAllDiagnostics	Find all diagnostics available on a system. See section 9.4.
	GetAllDiagnosticMEPairs	Find all diagnostic/managed elements pairs available on a system. See section 9.4.2.
	GetDiagnosticsForME	Find all the diagnostics available on a system, for a managed element. See section 9.4.3.
	GetMEsForDiagnostic	Find all the managed elements that support a particular diagnostic. See section 9.4.4.
	GetCapabilitiesOfDiagnostic	Find the capabilities of a particular diagnostic. See section 9.4.5.
	GetCharacteristicsOfDiagnostic	Find the characteristics of a particular diagnostic. See section 9.4.6.
	GetDiagnosticsWithCharacteristicsForME	Find all the diagnostics available on a system, for a managed element, with certain characteristics. See section 9.4.7.
	GetDiagnosticsWithCapabilitiesForME	Find all the diagnostics available on a system, for a managed element, with certain capabilities. See section 9.4.8.
Configure Diagnostic See section 9.5.	GetPackageSubtests	Find the subtests for a diagnostic test with the value of the DiagnosticTest.Characteristics property set to Is Package. See section 9.4.9.
	GetDefaultDiagnosticSettings	Find the default diagnostic settings for a diagnostic. See section 9.5.1.
	CreateDiagnosticSettings	Create a unique setting for a diagnostic. See section 9.5.2.
	GetDefaultJobSettings	Find the default job settings for a diagnostic. See section 9.5.3.
Execute and Control Diagnostic See section 9.6.	CreateJobSettings	Create a unique setting for a diagnostic job. See section 9.5.4.
	RunDiagnostic	Run a diagnostic with default and unique settings. See section 9.6.1.
	SuspendDiagnostic	Suspend a running diagnostic. See section 9.6.2.
	ResumeDiagnostic	Resume a running diagnostic. See section 9.6.3.
	AbortDiagnostic	Abort a running diagnostic. See section 9.6.4.
	KillDiagnostic	Abort a running diagnostic immediately, with no attempt to perform a clean shutdown. See section 9.6.5.

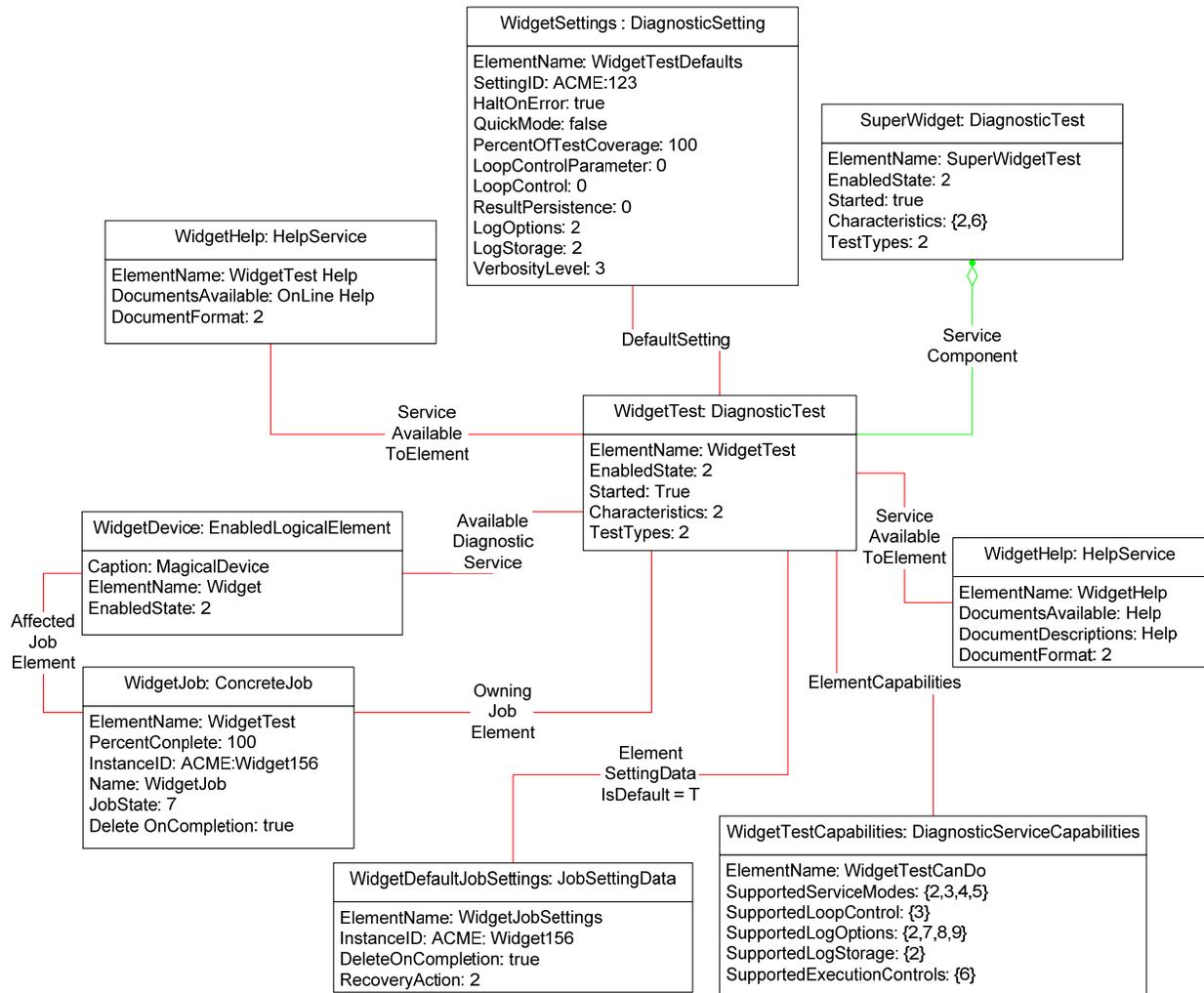
Category	Name	Description
Discover Diagnostic Executions See section 9.7.	GetAffectedMEs	Find all the managed elements affected by a running diagnostic. See section 9.7.1.
	GetAllDiagnosticExecutionsForME	Find all the diagnostic executions on a system for a managed element. See section 9.7.2.
	GetSpecificDiagnosticExecutions	Find all the executions of a specific diagnostic. See section 9.7.3.
	GetSpecificDiagnosticExecutionsForME	Find all the executions of a specific diagnostic for a particular managed element. See section 9.7.4.
Discover Diagnostic Results (in-progress and final) See section 9.8.	GetLogRecordsForDiagnostic	Find all the diagnostic log records for a particular diagnostic. See section 9.8.1.
	GetLogRecordsForME	Find all the diagnostic log records for a particular managed element. See section 9.8.2.
	GetLogRecordsForMEAndDiagnostic	Find all the diagnostic log records for a particular diagnostic run on a particular managed element. See section 9.8.3.
	GetDiagnosticExecutionResults	Find all diagnostic log records for a particular execution (job). See section 9.8.4.
	GetDiagnosticExecutionSettings	Find the settings used in a diagnostic execution. See section 9.8.5.
	GetDiagnosticProgress	Get the progress of a running diagnostic. See section 9.8.6.
	GetDiagnosticExecutionFinalResults	Find the diagnostic log record with final results for a particular execution (job). See section 9.8.7.

807 **9.3 Diagnostic Services Object Diagram**

808 Figure 3 is an object diagram for diagnostic services for a fictitious device called “Widget.” Only classes,
 809 properties, and methods that are of particular interest for the diagnostic model are shown. Refer to this
 810 diagram for the use cases in this section.

811 The CIM_ prefix has been omitted from the class names in the diagram for readability.

812



813

814

Figure 3 – Diagnostic Services Object Diagram

815 **9.4 Discover Available Diagnostics**

816 The use cases in this section describe how the client can find available diagnostics. The CIM_ prefix has
817 been omitted from the class names in the use cases for readability.

818 **9.4.1 GetAllDiagnostics**

819 The client can find all of the diagnostics that are available on a system as follows:

- 820 1) The client calls the EnumerateInstances (or EnumerateInstanceNames) operation using the
821 DiagnosticTest class.
- 822 2) The operation returns DiagnosticTest instances that represent a diagnostic that is available on
823 the system.

824 **9.4.2 GetAllDiagnosticMEPairs**

825 The client can find all of the diagnostics/managed element pairs that are available on a system as follows.
826 Each pair comprises a diagnostic and a ManagedElement (device) that is supported by the diagnostic.

- 827 1) The client calls the EnumerateInstances (or EnumerateInstanceNames) operation using the
828 AvailableDiagnosticService class.
- 829 2) The operation returns AvailableDiagnosticService instances that have a reference to the
830 DiagnosticTest instance and another reference to the ManagedElement instance.

831 **9.4.3 GetDiagnosticsForME**

832 The client can find all of the diagnostics on a system that can be launched against a specific device
833 (managed element) as follows. Assume that the client starts at a known ManagedElement instance,
834 which represents the device to be tested.

- 835 1) From the ManagedElement instance, the client calls the Associators operation
836 using AvailableDiagnosticService as the association class.
- 837 2) The operation returns DiagnosticTest instances that represent a diagnostic that can be
838 launched against the ManagedElement.

839 **9.4.4 GetMEsForDiagnostic**

840 The client can find all managed elements (devices) that are supported by a specific diagnostic as follows.
841 Assume that the client starts at a known DiagnosticTest instance.

- 842 1) From the DiagnosticTest instance, the client calls the Associators operation
843 using AvailableDiagnosticService as the association class.
- 844 2) The operation returns ManagedElement instances that represent a device that is supported by
845 the DiagnosticTest.

846 **9.4.5 GetCapabilitiesOfDiagnostic**

847 A diagnostic service publishes its support for various options—in particular, settings—through a
848 DiagnosticServiceCapabilities instance. If a setting is supported, the client can assign it, usually to satisfy
849 a user request. The client should be able to find the capabilities of a diagnostic as follows. Assume that
850 the client starts at a known DiagnosticTest instance.

- 851 1) From the DiagnosticTest instance, the client calls the Associators operation
852 using ElementCapabilities as the association class and DiagnosticServiceCapabilities as the
853 result class.
- 854 2) The operation should return only one DiagnosticServiceCapabilities instance, which represents
855 the diagnostic capabilities.

856 **Note:** Because the implementation of DiagnosticServiceCapabilities is optional, it may not be available. In
857 this case, no assumptions should be made regarding the diagnostic capabilities.

858 **9.4.6 GetCharacteristicsOfDiagnostic**

859 The client can discover all of the characteristics (is destructive, is interactive, is synchronous, and so on)
860 of a diagnostic. From the DiagnosticTest instance, the client reads just the Characteristics and
861 OtherCharacteristicsDescriptions attributes, which contain the diagnostic characteristics.

862 **9.4.7 GetDiagnosticswithCharacteristicsForME**

863 The client can find all of the diagnostics that can be launched against a specific device (managed
864 element) and have specific characteristics as follows. Assume that the client starts at a known
865 ManagedElement instance, which represents the device to be tested.

- 866 1) The client discovers all of the diagnostics that are available for the specific ManagedElement.
867 The GetDiagnosticsForME use case (section 9.4.3) describes the necessary steps.
- 868 2) For each DiagnosticTest instance, the client checks the diagnostic characteristics. The
869 GetCharacteristicsOfDiagnostic use case (section 9.4.6) describes the necessary steps.
- 870 3) If the characteristics of the DiagnosticTest instance match the desired characteristics, the
871 DiagnosticTest instance is the one desired.

872 **9.4.8 GetDiagnosticswithCapabilitiesForME**

873 The client can find all of the diagnostics that can be launched against a specific device (managed
874 element) and have specific capabilities as follows. Assume that the client starts at a known
875 ManagedElement instance, which represents the device to be tested.

- 876 1) The client discovers all of the diagnostics that are available for the specific ManagedElement.
877 The GetDiagnosticsForME use case (section 9.4.3) describes the necessary steps.
- 878 2) For each DiagnosticTest instance, the client checks the diagnostic capabilities. The
879 GetCapabilitiesOfDiagnostic use case (section 9.4.5) describes the necessary steps.
- 880 3) If the capabilities of the DiagnosticTest instance match the desired capabilities, the
881 DiagnosticTest instance is the one desired.

882 **9.4.9 GetPackageSubtests**

883 The client can find the subtests for a diagnostic test with the IsPackage value set in the
884 DiagnosticTest.Characteristics property, using the following procedure. Assume that the client starts at a
885 known DiagnosticTest instance.

- 886 1) The client checks the DiagnosticTest.Characteristics property for the IsPackage value.
- 887 2) If the IsPackage value is present, the client calls the Associators operation using
888 ServiceComponent as the association class and DiagnosticTest as the result class.
- 889 3) The operation returns the DiagnosticTest instances that are subtests of the known
890 DiagnosticTest.

891 **9.5 Configure Diagnostic**

892 The use cases in this section describe how the client can find and create settings for diagnostics. The
893 CIM_ prefix has been omitted from the class names in the use cases for readability.

894 **9.5.1 GetDefaultDiagnosticSettings**

895 The client can obtain the default settings for a diagnostic service as follows. Assume that the client starts
896 at a known DiagnosticTest instance.

- 897 1) From the DiagnosticTest instance, the client calls the Associators operation
898 using DefaultSetting as the association class and DiagnosticSetting as the result class.
- 899 2) The operation should return only one DiagnosticSetting instance, which represents the default
900 settings of the diagnostic.

901 **Note:** Because the implementation of DiagnosticSetting is optional, it may not be available. In this case,
902 no assumptions should be made regarding the default settings of diagnostic.

903 **9.5.2 CreateDiagnosticSettings**

904 The client may modify the diagnostic settings as follows if it wants to use settings different than the
905 default settings. Note that the diagnostic default settings are represented by a DiagnosticSetting subclass
906 that may have extensions. If the client is aware of the extensions, they may be modified as well. If the
907 client is unaware, the default values should be used. Assume that the client starts at a known
908 DiagnosticTest instance.

- 909 1) The client discovers the diagnostic capabilities of the DiagnosticTest instance. The
910 GetCapabilitiesOfDiagnostic use case (section 9.4.5) describes the necessary steps. If no
911 capability information is available, the client shall use the default settings (empty string or NULL)
912 because it cannot assume any diagnostic capability.
- 913 2) The client discovers the diagnostic default settings of the DiagnosticTest instance. The
914 GetDefaultDiagnosticSettings use case (section 9.5.1) describes the necessary steps. If no
915 instance is returned, the client shall use the default settings (empty string or NULL) because it
916 cannot assume support for any diagnostic setting..
- 917 3) The client modifies the created DiagnosticSetting instance as necessary. However, the client
918 should consider the diagnostic capabilities during the changes and shall modify the SettingID
919 attribute.
- 920 4) Finally, the client calls CreateInstance operation passing the instance modified on step 3.

921 **9.5.3 GetDefaultJobSettings**

922 The client can obtain the default job settings for a diagnostic service as follows. Assume that the client
923 starts at a known DiagnosticTest instance.

- 924 1) From the DiagnosticTest instance, the client calls the Associators operation
925 using ElementSettingData as the association class and JobSettingData as the result class. The
926 operation returns JobSettingData instances.
- 927 2) For each JobSettingData instance, the client calls the References operation using
928 ElementSettingData as the result class. The operation returns ElementSettingData instances.
- 929 3) For each ElementSettingData instance, the client determines whether the value of the
930 ElementSettingData.ManagedElement property matches the reference to the instance of
931 DiagnosticTest and the value of the ElementSettingData.IsDefault property is 1 ("Is Default"). If
932 so, the JobSettingData instance represents the default job settings. The name of this
933 JobSettingData instance may also be retrieved from ElementSettingData.SettingData property.

934 **Note:** Because the implementation of JobSettingData is optional, it may not be available. In this case, no
935 assumptions should be made regarding the default job settings of diagnostic.

936 9.5.4 CreateJobSettings

937 The client can modify the diagnostic job settings as follows if it wants to use settings different than the
938 default job settings. Note that the diagnostic default job settings are represented by a JobSettingData
939 subclass that may have extensions. If the client is aware of the extensions, they may be modified as well.
940 If the client is unaware, the default values should be used. Assume that the client starts at a known
941 DiagnosticTest instance.

- 942 1) The client discovers the diagnostic capabilities of the DiagnosticTest instance. The
943 GetCapabilitiesOfDiagnostic use case (section 9.4.5) describes the necessary steps. If no
944 capability information is available, the client shall use the default settings (empty string or NULL)
945 because it cannot assume any diagnostic capability.
- 946 2) The client discovers the diagnostic default settings of the DiagnosticTest instance. The
947 GetDefaultJobSettings use case (section 9.5.3) describes the necessary steps. If no instance is
948 returned, the client shall use the default settings (empty string or NULL) because it cannot
949 assume support for any job setting.
- 950 3) The client modifies the created JobSettingData instance as necessary. However, the client
951 should consider the diagnostic capabilities during the changes and shall modify the InstanceID
952 attribute.
- 953 4) Finally, the client calls CreateInstance operation passing the instance modified on step 3.

954 9.6 Execute and Control Diagnostic

955 The RunDiagnostic() method is invoked to start the diagnostic service. Input parameters are the
956 ManagedElement being tested and the settings (optional). A reference to a ConcreteJob instance is
957 returned.

958 An instance of ConcreteJob is created by the diagnostic provider to allow monitoring and control of the
959 running service. By invoking the RequestStateChange method, the client may start, stop, suspend, and
960 resume the job. By inspecting the value of PercentComplete, the client may determine the job's progress.

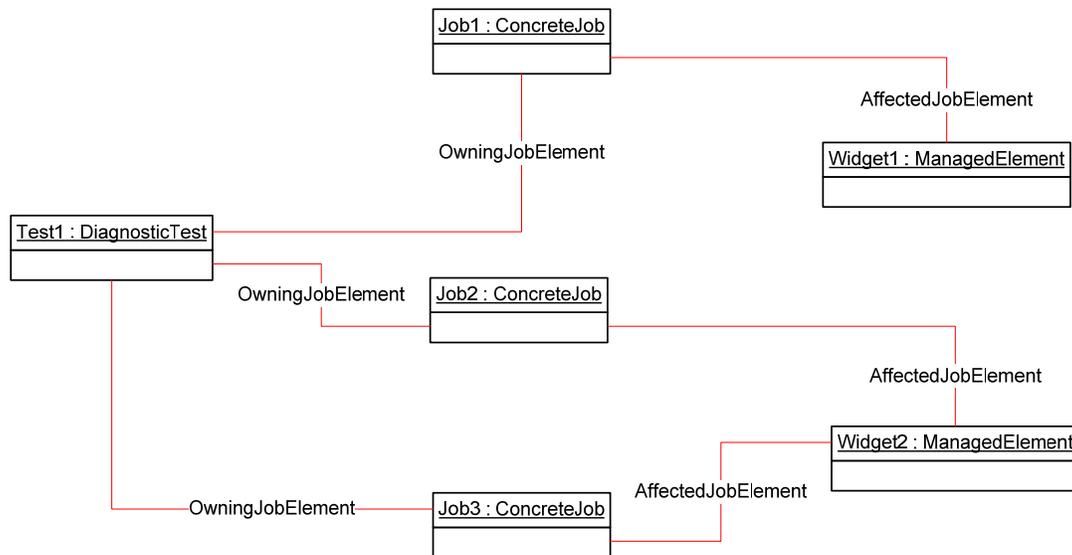
961 The ManagedElement being tested and the DiagnosticTest instance that launched the test are related to
962 the job instance through the OwningJobElement and the AffectedJobElement associations. The client
963 may find jobs associated with services or managed elements of interest by using these associations.

964 Figure 4 is an object diagram that shows the state of instances when a DiagnosticTest RunDiagnostic()
965 method has been called three times. Two of the times were to run a test on the same device,
966 ManagedElement2.

967 **Note:** Not all diagnostic tests are capable of running on the same device simultaneously. If this had been
968 the case in this example, the DiagnosticTest would have returned an error on the second
969 RunDiagnostic() method call to run a test on ManagedElement2.

970 **Note:** Diagnostic tests that do not return a reference to a ConcreteJob instance are assumed to have
971 completed the execution of the test upon return from the RunDiagnostic() method and thus there is no
972 need for a ConcreteJob reference to provide execution status or control execution. In this case, only the
973 RunDiagnostic use case is valid.

974 The CIM_ prefix has been omitted from the class names in the diagram and the use cases for readability.



975

976

Figure 4 – Job Example

977 9.6.1 RunDiagnostic

978 The client can run a diagnostic with default and unique settings as follows. (See section 9.4 for use cases
 979 related to finding diagnostics that can be initiated. See section 9.5 for use cases related to creating and
 980 modifying diagnostic settings to configure diagnostic execution.)

- 981 1) The client calls the RunDiagnostic() method, passing in references of DiagnosticSetting and
 982 JobSetting to use to execute the test as well as the reference to the ManagedElement to test. If
 983 the client passes in NULL or an empty string for these classes, the default values are used.
- 984 2) The diagnostic service creates a Job instance to represent that test running on that managed
 985 element and returns a reference to it in the return call from RunDiagnostic(). In addition, the
 986 diagnostic service creates the OwningJobElement association between the Job and the Service
 987 and the AffectedJobElement association between the Job and the ManagedElement.
- 988 3) If the diagnostic service does not create and return a Job instance, the test is assumed to have
 989 completed execution upon return from the RunDiagnostic() method.

990 9.6.2 SuspendDiagnostic

991 The client can suspend the execution of the test by using the RequestStateChange() method call on the
 992 Job instance that is returned from the RunDiagnostic() method, as shown in the following procedure.
 993 Assume that the client starts at a known DiagnosticTest instance.

- 994 1) The client follows the ElementCapabilities association from the DiagnosticTest to the
 995 DiagnosticServiceCapabilities for the service.
- 996 2) The client checks the DiagnosticServiceCapabilities.SupportedExecutionControls() property for
 997 the value of "Suspend Job". If the value exists, the Job supports suspending.
- 998 3) The client finds the appropriate Job instances. The GetSpecificDiagnosticExecutions use case
 999 (section 9.7.3) describes the necessary steps.
- 1000 4) The client calls the RequestStateChange() method, passing in a RequestedState value of
 1001 "Suspend".
- 1002 5) When the transition completes successfully, the ConcreteJob that represents the test will set the
 1003 value of the JobState property to "Suspended" and set the value of TimeOfLastStateChange to
 1004 the current time.

1005 9.6.3 ResumeDiagnostic

1006 The client can resume the execution of a test by using the RequestStateChange() method call on the Job
1007 instance that is returned from the RunDiagnostic() method, as shown in the following procedure. Assume
1008 that the client starts at a known DiagnosticTest instance.

- 1009 1) The client follows the ElementCapabilities association from the DiagnosticTest to the
1010 DiagnosticServiceCapabilities for the service.
- 1011 2) The client checks the DiagnosticServiceCapabilities.SupportedExecutionControls() property for
1012 the value of "Suspend Job". If the value exists, the Job supports suspending and resuming.
- 1013 3) The client finds the appropriate Job instances. The GetSpecificDiagnosticExecutions use case
1014 (section 9.7.3) describes the necessary steps.
- 1015 4) The client calls the RequestStateChange() method, passing in a RequestedState value of
1016 "Start".
- 1017 5) When the transition completes successfully, the ConcreteJob that represents the test will set the
1018 value of the JobState property to "Running" and set the value of TimeOfLastStateChange to the
1019 current time.

1020 **Note:** The JobState property may transition to "Starting" before the final transition to "Running".

1021 9.6.4 AbortDiagnostic

1022 The client can cleanly abort the execution of a test by using the RequestStateChange() method call on
1023 the Job instance that is returned from the RunDiagnostic() method, as shown in the following procedure.
1024 Assume that the client starts at a known DiagnosticTest instance.

- 1025 1) The client follows the ElementCapabilities association from the DiagnosticTest to the
1026 DiagnosticServiceCapabilities for the service.
- 1027 2) The client checks the DiagnosticServiceCapabilities.SupportedExecutionControls() property for
1028 the value of "Terminate Job". If the value exists, the Job supports termination.
- 1029 3) The client finds the appropriate Job instances. The GetSpecificDiagnosticExecutions use case
1030 (section 9.7.3) describes the necessary steps.
- 1031 4) The client calls the RequestStateChange() method, passing in a RequestedState value of
1032 "Terminate".
- 1033 5) When the transition completes successfully, the ConcreteJob that represents the test will set the
1034 value of the EnabledState property to "Terminated" and set the value of
1035 TimeOfLastStateChange to the current time.

1036 **Note:** The JobState property may transition to "Shutting Down" before the final transition to "Terminated".

1037 9.6.5 KillDiagnostic

1038 The client can immediately abort the execution of a test, with no attempt to perform a clean shutdown, by
1039 using the RequestStateChange() method call on the Job instance that is returned from the
1040 RunDiagnostic() method, as shown in the following procedure. Assume that the client starts at a known
1041 DiagnosticTest instance.

- 1042 1) The client follows the ElementCapabilities association from the DiagnosticTest to the
1043 DiagnosticServiceCapabilities for the service.
- 1044 2) The client checks the DiagnosticServiceCapabilities.SupportedExecutionControls() property for
1045 the value of "Kill Job". If the value exists, the Job supports kill.
- 1046 3) The client finds the appropriate Job instances. The GetSpecificDiagnosticExecutions use case
1047 (section 9.7.3) describes the necessary steps.

- 1048 4) The client calls the RequestStateChange() method, passing in a RequestedState value of "Kill".
- 1049 5) When the transition completes successfully, the ConcreteJob that represents the test will set the
- 1050 value of the EnabledState property to "Killed" and set the value of TimeOfLastStateChange to
- 1051 the current time.

1052 9.7 Discover Diagnostic Executions

1053 In the following use cases, the term *execution* refers to an instance of the ConcreteJob class created to

1054 control a diagnostic service that was started on a managed element. The job may be in any of the states

1055 represented by the JobState property value, not necessarily active and running. The use cases in this

1056 section only apply to implementations that return a reference to ConcreteJob upon execution of the

1057 RunDiagnostic() method.

1058 The CIM_ prefix has been omitted from the class names in the use cases for readability.

1059 9.7.1 GetAffectedMEs

1060 The client can find all of the managed elements that are affected by a diagnostic execution as follows.

1061 Assume that the client starts at a known DiagnosticTest instance.

- 1062 1) From the DiagnosticTest instance, the client calls the Associators operation using
- 1063 OwningJobElement as the association class and ConcreteJob as the result class. The operation
- 1064 returns the ConcreteJob instances launched by the DiagnosticTest.
- 1065 2) For each ConcreteJob instance, the client calls the Associators operation using
- 1066 AffectedJobElement as the association class and ManagedElement as the result class. The
- 1067 operation returns the ManagedElement instances that this DiagnosticTest affects.

1068 **Note:** This use case depends on the optional AffectedJobElement association. If that association does

1069 not exist, this use case is invalid.

1070 9.7.2 GetAllDiagnosticExecutionsForME

1071 The client can find all of the diagnostic executions on a system for a managed element as follows.

1072 Assume that the client starts at a known ManagedElement instance.

- 1073 1) From the ManagedElement instance, the client calls the Associators operation
- 1074 using AffectedJobElement as the association class. The operation returns the ConcreteJob
- 1075 instances launched against this ManagedElement.
- 1076 2) For each ConcreteJob instance, the client calls the AssociatorNames operation using
- 1077 OwningJobElement as the association class and DiagnosticTest as the result class. The
- 1078 operation returns the instance paths to the DiagnosticTest instances that launched the
- 1079 ConcreteJob against this ManagedElement.
- 1080 3) Each ConcreteJob instance that is associated with a DiagnosticTest represents an execution of
- 1081 a diagnostic service on that ManagedElement.

1082 **Note:** This use case depends on the optional AffectedJobElement association. If that association does

1083 not exist, this use case is invalid.

1084 9.7.3 GetSpecificDiagnosticExecutions

1085 The client can find all of the executions of a specific diagnostic as follows. Assume that the client starts at

1086 a known DiagnosticTest instance.

- 1087 1) From the DiagnosticTest instance, the client calls the Associators operation
- 1088 using OwningJobElement as the association class. The operation returns the ConcreteJob
- 1089 instances launched by the DiagnosticTest.
- 1090 2) Each ConcreteJob instance represents an execution of that diagnostic service.

1091 **9.7.4 GetSpecificDiagnosticExecutionsForME**

1092 The client can find all of the executions of a specific diagnostic for a particular managed element using
1093 either of the following methods:

- 1094 • starting at the known ManagedElement instance
- 1095 • starting at the known DiagnosticTest instance

1096 **9.7.4.1 Starting at the Managed Element**

1097 **Note:** This use case depends on the optional AffectedJobElement association. If that association does
1098 not exist, this use case is invalid.

1099 Assume that the client starts at the known ManagedElement instance and knows the particular
1100 DiagnosticTest instance.

- 1101 1) From the ManagedElement instance, the client calls the Associators operation
1102 using AffectedJobElement as the association class and ConcreteJob as the result class. The
1103 operation returns the ConcreteJob instances that are running against this ManagedElement.
- 1104 2) For each ConcreteJob instance, the client calls the AssociatorNames operation using
1105 OwingJobElement as the association class and DiagnosticTest as the result class. The
1106 operation returns the instance paths to the DiagnosticTest instances that launched the
1107 ConcreteJob instances against this ManagedElement.
- 1108 3) For each DiagnosticTest instance path returned, the client determines if it is the instance path of
1109 the known DiagnosticTest instance. If the instance path matches, the ConcreteJob instance
1110 represents an execution of that diagnostic service on that ManagedElement.

1111 **9.7.4.2 Starting at the DiagnosticTest**

1112 **Note:** This use case depends on the optional AffectedJobElement association. If that association does
1113 not exist, this use case is invalid.

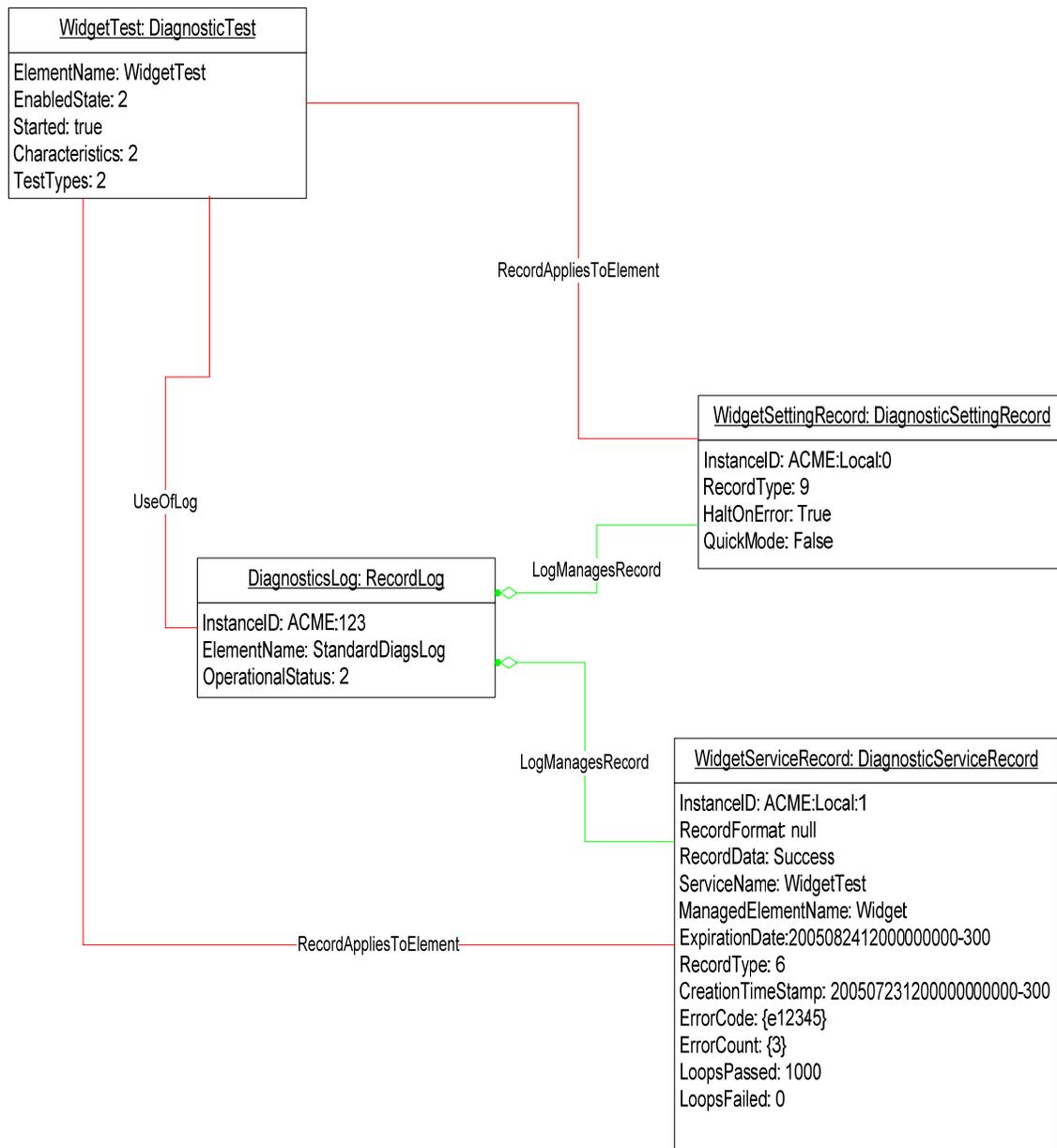
1114 Assume that the client starts at the known DiagnosticTest instance and knows the particular
1115 ManagedElement instance.

- 1116 1) From the DiagnosticTest instance, the client calls the Associators operation using
1117 OwingJobElement as the association class and ConcreteJob as the result class. The operation
1118 returns the ConcreteJob instances launched by the DiagnosticTest.
- 1119 2) For each ConcreteJob instance, the client calls the AssociatorNames operation using
1120 AffectedJobElement as the association class and ManagedElement as the result class. The
1121 operation returns the instance paths to the ManagedElement instances against which this
1122 DiagnosticTest launched the ConcreteJob instances.
- 1123 3) For each ManagedElement instance path returned, the client determines if it is the instance
1124 path of the known ManagedElement instance. If the instance path matches, the ConcreteJob
1125 instance represents an execution of that diagnostic service on that ManagedElement.

1126 **9.8 Discover Diagnostic Results (In Progress and Final)**

1127 In the following use cases, the term *execution* refers to an instance of the ConcreteJob class created to
1128 control a diagnostic service that was started on a managed element. The job may be in any of the states
1129 represented by the JobState property value, not necessarily active and running. Some of the use cases in
1130 this section only apply to implementations that return a reference to ConcreteJob upon execution of the
1131 RunDiagnostic() method.

- 1132 Figure 5 is an object diagram that represents the results logging process for a diagnostic service on a
1133 fictitious device called "Widget". Only classes, properties, and methods that are of particular interest for
1134 the diagnostic model are shown.
- 1135 Figure 5 shows the logging implementation, using the DiagnosticsLog class. DiagnosticsLog is a special
1136 subclass of RecordLog that supports a standard mechanism for organizing and retrieving the records that
1137 diagnostics services generate. Use of this common logging mechanism can substantially increase
1138 interoperability and simplify client design.
- 1139 The diagnostic provider will store the results of running the diagnostic in the manner selected through the
1140 LogStorage setting. The most common mechanism is for the provider to create instances of
1141 DiagnosticRecord to record the results and status of running diagnostic services. DiagnosticRecord has
1142 two subclasses: DiagnosticServiceRecord for recording test results, and DiagnosticSettingRecord for
1143 preserving the test settings. The providers for these classes can implement ExecQuery to simplify the
1144 retrieval of records. The use cases below provide alternatives that use ExecQuery as well as do not.
- 1145 The records are aggregated to a log by the LogManagesRecord association.
- 1146 The CIM_ prefix has been omitted from the class names in the diagram and use cases for readability.



1147

1148

Figure 5 – Diagnostic Logging Object Diagram

1149 **9.8.1 GetLogRecordsForDiagnostic**

1150 The client can find all of the diagnostic log records for a particular diagnostic as follows. Assume that the
 1151 client starts at the known DiagnosticTest instance and that the DiagnosticRecord.ServiceName property
 1152 is implemented according to this Profile.

1153 1) The client calls the ExecQuery operation as follows:

```

1154 SELECT * FROM CIM_DiagnosticRecord
1155 WHERE ServiceName = '<DiagnosticTest.Name>'
    
```

1156 2) The operation returns the DiagnosticRecord instances created for the specific DiagnosticTest,
 1157 independently if they are related to different managed elements or executions.

1158 An alternate method without using ExecQuery can be done by the following:

1159 Assume that the client starts at the known DiagnosticTest instance.

- 1160 1) From the DiagnosticTest instance, the client calls the Associators operation using UseOfLog as
1161 the association class and DiagnosticsLog as the result class. The operation returns the
1162 DiagnosticsLog instances that contain records for the DiagnosticTest.
- 1163 2) For each DiagnosticsLog instance, the client calls the Associators operation using
1164 LogManagesRecord as the association class and DiagnosticRecord as the result class. The
1165 operation returns the DiagnosticRecord instances in the Log.
- 1166 3) For each returned instance, the client compares DiagnosticRecord.ServiceName with
1167 DiagnosticTest.Name to determine if the instance is one created for the specific DiagnosticTest.

1168 9.8.2 GetLogRecordsForME

1169 The client can find all of the diagnostic log records for a particular managed element as follows. Assume
1170 that the client starts at the known ManagedElement instance and that the
1171 DiagnosticRecord.ManagedElementName property is implemented according to this Profile.

1172 1) The client calls the ExecQuery operation as follows:

```
1173 SELECT * FROM CIM_DiagnosticRecord
1174 WHERE ManagedElementName = '<ManagedElement.ElementName>'
```

1175 2) The operation returns the DiagnosticRecord instances created for the specific
1176 ManagedElement, independently if they are related to different diagnostics or executions.

1177 An alternate method without using ExecQuery can be done by the following:

1178 Assume that the client starts at the known ManagedElement instance.

- 1179 1) From the ManagedElement instance, the client calls the Associators operation using
1180 ServiceAvailableToElement as the association class and DiagnosticTest as the result class. The
1181 operation returns the DiagnosticTest instances for the ManagedElement.
- 1182 2) For each DiagnosticTest instance, the client calls the Associators operation using UseOfLog as
1183 the association class and DiagnosticsLog as the result class. The operation returns the
1184 DiagnosticsLog instances that contain records for the DiagnosticTest.
- 1185 3) For each DiagnosticsLog instance, the client calls the Associators operation using
1186 LogManagesRecord as the association class and DiagnosticRecord as the result class. The
1187 operation returns the DiagnosticRecord instances in the Log.
- 1188 4) For each returned instance, the client compares DiagnosticRecord.ManagedElementName with
1189 ManagedElement.ElementName to determine if the instance is one created for the specific
1190 ManagedElement.

1191 9.8.3 GetLogRecordsForMEAndDiagnostic

1192 The client can find all of the diagnostic log records for a particular diagnostic run on a particular managed
1193 element as follows.

1194 Assume that the client starts at the known DiagnosticTest and ManagedElement instances and that the
1195 DiagnosticRecord.ServiceName and DiagnosticRecord.ManagedElementName properties are
1196 implemented according to this Profile.

1197 1) The client calls the ExecQuery operation as follows:

```
1198 SELECT * FROM CIM_DiagnosticRecord
1199 WHERE ManagedElementName = '<ManagedElement.ElementName>' and ServiceName =
1200 '<DiagnosticTest.Name>'
```

- 1201 2) The operation returns the DiagnosticRecord instances created for the specific ManagedElement
1202 and DiagnosticTest, independently if they were created in different executions.

1203 An alternate method without using ExecQuery can be done by the following:

1204 Assume that the client starts at the known DiagnosticTest instance.

- 1205 1) From the DiagnosticTest instance, the client calls the Associators operation using UseOfLog as
1206 the association class and DiagnosticsLog as the result class. The operation returns the
1207 DiagnosticsLog instances that contain records for the DiagnosticTest.
- 1208 2) For each DiagnosticsLog instance, the client calls the Associators operation using
1209 LogManagesRecord as the association class and DiagnosticRecord as the result class. The
1210 operation returns the DiagnosticRecord instances in the Log.
- 1211 3) For each returned instance, the client compares DiagnosticRecord.ServiceName with
1212 DiagnosticTest.Name and DiagnosticRecord.ManagedElementName with
1213 ManagedElement.ElementName to determine if the instance is one created for the specific
1214 DiagnosticTest and ManagedElement.

1215 **9.8.4 GetDiagnosticExecutionResults**

1216 The client can find all diagnostic log records for a particular execution (job) as follows.

1217 Assume that the client starts at the known ConcreteJob instance and that the
1218 DiagnosticRecord.InstanceID property follows the format defined in this Profile
1219 (CIM_DiagnosticRecord.InstanceID *should* be <ConcreteJob.InstanceID>:<n>). This use case is also
1220 applicable after the job completes and is removed if the client knows the original ConcreteJob.InstanceID.

- 1221 1) The client calls the ExecQuery operation as follows:

```
1222           SELECT * FROM CIM_DiagnosticRecord
1223           WHERE InstanceID LIKE '<ConcreteJob.InstanceID>%'
```

- 1224 2) The operation returns the DiagnosticRecord instances created for the specific ConcreteJob.

1225 An alternate method without using ExecQuery can be done as follows:

1226 Assume that the client starts at the known DiagnosticTest instance.

- 1227 1) From the DiagnosticTest instance, the client calls the Associators operation using UseOfLog as
1228 the association class and DiagnosticsLog as the result class. The operation returns the
1229 DiagnosticsLog instances that contain records for the DiagnosticTest.
- 1230 2) For each DiagnosticsLog instance, the client calls the Associators operation using
1231 LogManagesRecord as the association class and DiagnosticRecord as the result class. The
1232 operation returns the DiagnosticRecord instances in the Log.
- 1233 3) For each returned instance, the client compares portion of DiagnosticRecord.InstanceID that
1234 contains the ConcreteJob.InstanceID with ConcreteJob.InstanceID to determine if the instance
1235 is one created for the specific execution of the DiagnosticTest.

1236 An alternate method without using ExecQuery can be done as follows:

1237 **Note:** This alternative use case depends upon the implementation of DiagnosticSettingRecord
1238 and CorrespondingSettingsRecord.

1239 Assume that the client starts at the known DiagnosticTest instance.

- 1240 1) From the DiagnosticTest instance, the client calls the Associators operation using UseOfLog as
1241 the association class and DiagnosticsLog as the result class. The operation returns the
1242 DiagnosticsLog instances that contain records for the DiagnosticTest.

- 1243 2) For each DiagnosticsLog instance, the client calls the Associators operation using
 1244 LogManagesRecord as the association class and DiagnosticSettingRecord as the result class.
 1245 The operation returns the DiagnosticSettingRecord instances in the Log.
- 1246 3) For each returned instance, the client compares portion of DiagnosticSettingRecord.InstanceID
 1247 that contains the ConcreteJob.InstanceID with ConcreteJob.InstanceID to determine if the
 1248 instance is the one created for the specific execution of the DiagnosticTest.
- 1249 4) From the DiagnosticSettingRecord instance, the client calls the Associators operation using
 1250 CorrespondingSettingsRecord as the association class and DiagnosticServiceRecord as the
 1251 result class. The operation returns the DiagnosticServiceRecord instances created for the
 1252 specific execution of the DiagnosticTest
- 1253 **Note:** All these alternatives only apply to implementations that return a reference to ConcreteJob
 1254 from the RunDiagnostic() method.

1255 9.8.5 GetDiagnosticExecutionSettings

1256 The client can find the settings used to execute a diagnostic as follows.

1257 Assume that the client starts at the known ConcreteJob instance and that the
 1258 DiagnosticRecord.InstanceID property follows the format defined in this Profile
 1259 (CIM_DiagnosticRecord.InstanceID *should* be <ConcreteJob.InstanceID>:<n>). This use case is also
 1260 applicable after the job completes and is removed if the client knows the original ConcreteJob.InstanceID.

- 1261 1) The client calls the ExecQuery operation as follows:
- ```
1262 SELECT * FROM CIM_DiagnosticSettingRecord
1263 WHERE InstanceID LIKE '<ConcreteJob.InstanceID>%'
```
- 1264 2) The operation returns the DiagnosticSettingRecord instance created for the specific  
 1265 ConcreteJob.
- 1266 3) The client reads the DiagnosticSettingRecord properties, such as HaltOnError or QuickMode,  
 1267 which are a copy of the properties from the DiagnosticSetting instance that passed as a  
 1268 parameter in the RunDiagnostic() method.

1269 An alternate method without using ExecQuery can be done as follows:

1270 Assume that the client starts at the known DiagnosticTest instance.

- 1271 1) From the DiagnosticTest instance, the client calls the Associators operation using UseOfLog as  
 1272 the association class and DiagnosticsLog as the result class. The operation returns the  
 1273 DiagnosticsLog instances that contain records for the DiagnosticTest.
- 1274 2) For each DiagnosticsLog instance, the client calls the Associators operation using  
 1275 LogManagesRecord as the association class and DiagnosticSettingRecord as the result class.  
 1276 The operation returns the DiagnosticSettingRecord instances in the Log.
- 1277 3) For each returned instance, the client compares portion of DiagnosticSettingRecord.InstanceID  
 1278 that contains the ConcreteJob.InstanceID with ConcreteJob.InstanceID to determine if the  
 1279 instance is the one created for the specific execution of the DiagnosticTest.

1280 **Note:** This use case only applies to implementations that return a reference to ConcreteJob from the  
 1281 RunDiagnostic() method.

### 1282 9.8.6 GetDiagnosticProgress

1283 The client can get the progress of a running diagnostic as follows.

1284 The client may poll the ConcreteJob.PercentComplete property to determine test progress or register for  
 1285 an indication that this property has changed. The value of this property shall be kept current to be useful.

1286 Service providers should update this property within one second of becoming aware of a progress  
1287 change.

1288 1) The client may use any of the Discover Diagnostic Execution use cases (section 9.7) to find the  
1289 desired ConcreteJob instances.

1290 2) The client reads the ConcreteJob.PercentComplete property to determine test progress.

1291 Assuming CIM\_InstModification indications are supported, the client may register to receive indications  
1292 when the particular ConcreteJob.PercentComplete property changes value.

1293 1) The client can use any of the Discover Diagnostic Execution use cases (section 9.7) to find the  
1294 desired ConcreteJob instances.

1295 2) The client can register to receive a CIM\_InstModification indication by creating an indication  
1296 subscription using the following CIM\_IndicationFilter.Query:

```
1297 SELECT * FROM CIM_InstModification
1298 WHERE "SourceInstance.ISA("CIM_ConcreteJob") and SourceInstance.InstanceID =
1299 <ConcreteJob.InstanceID> and PreviousInstance.PercentComplete <>
1300 SourceInstance.PercentComplete
```

1301 3) The indication received will notify the client that the PercentComplete property for the specific  
1302 ConcreteJob has changed. The client can use the SourceInstance property in the indication to  
1303 see the actual PercentComplete value to determine test progress.

1304 **Note:** This use case only applies to implementations that return a reference to ConcreteJob from the  
1305 RunDiagnostic() method.

### 1306 9.8.7 GetDiagnosticExecutionFinalResults

1307 The client can find the final results log record for a particular execution (job) as follows.

1308 Assume that the client starts at the known ConcreteJob instance and that the  
1309 DiagnosticRecord.InstanceID property follows the format defined in this Profile  
1310 (CIM\_DiagnosticRecord.InstanceID *should* be <ConcreteJob.InstanceID>:<n>). This use case is also  
1311 applicable after the job completes and is removed if the client knows the original ConcreteJob.InstanceID.

1312 1) Client determines that the Job has completed by examining ConcreteJob.JobState. Value  
1313 should be Completed, Terminated or Killed.

1314 2) The client uses GetDiagnosticExecutionResults to get the DiagnosticServiceRecord instances  
1315 for the particular execution of the DiagnosticTest.

1316 3) For each returned DiagnosticServiceRecord, the client reads the  
1317 DiagnosticServiceRecord.RecordType to find the instance with RecordType = "Results" to find  
1318 the final results record.

1319 4) Client reads properties of interest to determine the final result, such as LoopsPassed,  
1320 LoopsFailed, ErrorCode[], ErrorCount[] and RecordData.

1321 **Note:** This use case only applies to implementations that return a reference to ConcreteJob from the  
1322 RunDiagnostic() method.

## 1323 10 CIM Elements

1324 Table 34 shows the instances of CIM Elements for this Profile. Instances of the CIM Elements shall be  
1325 implemented as described in Table 34. Section 8 ("Methods") may impose additional requirements on  
1326 these elements.

1327

**Table 34 – CIM Elements: Diagnostics Profile**

| Element Name                           | Requirement | Description                                                                                                                                                                                      |
|----------------------------------------|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Classes</b>                         |             |                                                                                                                                                                                                  |
| CIM_AffectedJobElement                 | Optional    | Association to link a job to a managed element<br>See section 10.1.                                                                                                                              |
| CIM_AvailableDiagnosticService         | Mandatory   | Association to link diagnostic services which can be launched against managed elements<br>See section 10.2.                                                                                      |
| CIM_ConcreteJob                        | Mandatory   | Used by the client to monitor and control the execution of a diagnostic service<br>See section 10.3.                                                                                             |
| CIM_DiagnosticsLog                     | Optional    | Although several legitimate mechanisms for logging results exist (see CIM_DiagnosticSetting.LogStorage), aggregation of diagnostic records to a diagnostic log is expected.<br>See section 10.4. |
| CIM_DiagnosticServiceCapabilities      | Conditional | This class is mandatory if CIM_DiagnosticSetting is implemented and non-default settings are supported. See sections 7.2 and 10.5.                                                               |
| CIM_DiagnosticServiceRecord            | Conditional | Records that contain the results of running a diagnostic service. If CIM_DiagnosticsLog is implemented, this class is mandatory.<br>See section 10.6.                                            |
| CIM_DiagnosticSetting                  | Optional    | See section 10.7.                                                                                                                                                                                |
| CIM_DiagnosticSettingRecord            | Conditional | Records that contain the settings that were used by the diagnostic service. If CIM_DiagnosticSetting and CIM_DiagnosticsLog are implemented, this class is Mandatory.<br>See section 10.8.       |
| CIM_DiagnosticTest                     | Mandatory   | See section 10.9.                                                                                                                                                                                |
| CIM_ElementCapabilities                | Optional    | Association to link a Capabilities object to a diagnostic service. If Capabilities is implemented, this association is Mandatory.<br>See section 10.10.                                          |
| CIM_DefaultSetting (DiagnosticSetting) | Optional    | If CIM_DiagnosticSetting is implemented, this association is Mandatory.<br>See section 10.11.                                                                                                    |
| CIM_DefaultSetting (JobSettingData)    | Optional    | See section 10.12                                                                                                                                                                                |
| CIM_ElementSettingData                 | Optional    | If CIM_JobSettingData is implemented, this association is Mandatory.<br>See section 10.21.                                                                                                       |
| CIM_HelpService                        | Optional    | See section 10.13.                                                                                                                                                                               |

| Element Name                       | Requirement | Description                                                                                                                                                                                                                                                                                                         |
|------------------------------------|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CIM_JobSettingData                 | Optional    | See section 10.14.                                                                                                                                                                                                                                                                                                  |
| CIM_LogManagesRecord               | Optional    | If CIM_DiagnosticsLog is implemented, this class is Mandatory.<br>See section 10.15.                                                                                                                                                                                                                                |
| CIM_OwningJobElement               | Mandatory   | Association to link a job to a diagnostic service<br>See section 10.16.                                                                                                                                                                                                                                             |
| CIM_RecordAppliesToElement         | Optional    | Association to link records to the managed element to which they apply<br>See section 10.17.                                                                                                                                                                                                                        |
| CIM_ServiceAffectsElement          | Optional    | See section 10.18.                                                                                                                                                                                                                                                                                                  |
| CIM_ServiceAvailableToElement      | Optional    | See section 10.19.                                                                                                                                                                                                                                                                                                  |
| CIM_ServiceComponent               | Optional    | Association to link a packaging test to its subtests. If a CIM_DiagnosticTest.Characteristic property contains the IsPackage value and the subtests are also instances of CIM_DiagnosticTest, this association shall be used to associate those subtests to the packaging CIM_DiagnosticTest.<br>See section 10.20. |
| CIM_UseOfLog                       | Optional    | See section 10.22.                                                                                                                                                                                                                                                                                                  |
| CIM_CorrespondingSettingsRecord    | Optional    | If CIM_DiagnosticsLog and CIM_DiagnosticSetting are implemented, this class is Mandatory.<br>See section 10.23.                                                                                                                                                                                                     |
| CIM_HostedService (DiagnosticTest) | Mandatory   | The association between CIM_DiagnosticTest and CIM_ComputerSystem instance is Mandatory.<br>See section 10.24.                                                                                                                                                                                                      |
| CIM_HostedService (HelpService)    | Optional    | If the HelpService class is implemented for the associated system, this class is mandatory.<br>See section 10.25.                                                                                                                                                                                                   |
| CIM_RegisteredProfile              | Mandatory   | See section 10.26.                                                                                                                                                                                                                                                                                                  |
| <b>Indications</b>                 |             |                                                                                                                                                                                                                                                                                                                     |
| None defined in this Profile       |             |                                                                                                                                                                                                                                                                                                                     |

1328 **10.1 CIM\_AffectedJobElement**

1329 CIM\_AffectedJobElement is used to associate a job with its affected managed elements (devices). Table  
1330 35 provides information about the properties of CIM\_AffectedJobElement.

1331 **Table 35 – Class: CIM\_AffectedJobElement**

| Properties       | Requirement | Notes                                                                               |
|------------------|-------------|-------------------------------------------------------------------------------------|
| AffectedElement  | Mandatory   | <b>Key</b> This property shall be a reference to an instance of CIM_ManagedElement. |
| AffectingElement | Mandatory   | <b>Key</b> This property shall be a reference to an instance of CIM_ConcreteJob.    |

1332 **10.2 CIM\_AvailableDiagnosticService**

1333 CIM\_AvailableDiagnosticService is used to discover the diagnostic services that are installed for a  
1334 particular managed element. Table 36 provides information about the properties of  
1335 CIM\_AvailableDiagnosticService.

1336 **Table 36 – Class: CIM\_AvailableDiagnosticService**

| Properties      | Requirement | Notes                                                                                  |
|-----------------|-------------|----------------------------------------------------------------------------------------|
| ServiceProvided | Mandatory   | <b>Key</b> This property shall be a reference to an instance of CIM_DiagnosticService. |
| UserOfService   | Mandatory   | <b>Key</b> This property shall be a reference to an instance of CIM_ManagedElement.    |

1337 **10.3 CIM\_ConcreteJob**

1338 Each successful RunDiagnostic() call will return a CIM\_ConcreteJob instance. Each CIM\_ConcreteJob  
1339 instance represents a diagnostic execution. Table 37 provides information about the properties of  
1340 CIM\_ConcreteJob.

1341 **Table 37 – Class: CIM\_ConcreteJob**

| Properties        | Requirement | Notes                                                                                                                                                                         |
|-------------------|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| InstanceID        | Mandatory   | <b>Key</b><br>InstanceID should be constructed using the following preferred algorithm:<br><OrgID>:<LocalID><br>(See the MOF file for more detail.)<br>(pattern "^.*[:].*\$") |
| Name              | Mandatory   | The property will be formatted as a free-form string of variable length.<br>(pattern ".*")                                                                                    |
| JobState          | Mandatory   | None                                                                                                                                                                          |
| TimeBeforeRemoval | Mandatory   |                                                                                                                                                                               |
| StartTime         | Mandatory   | None                                                                                                                                                                          |

| Properties           | Requirement | Notes                                                                             |
|----------------------|-------------|-----------------------------------------------------------------------------------|
| ElapsedTime          | Mandatory   | This property should be updated periodically so as to be useful as a "heartbeat." |
| PercentComplete      | Mandatory   |                                                                                   |
| DeleteOnCompletion   | Optional    | The default value for this property is TRUE.                                      |
| ErrorCode            | Optional    | 0 if the Job completed without error.                                             |
| ErrorDescription     | Conditional | If ErrorCode has a value other than 0, , ErrorDescription is Mandatory.           |
| RequestStateChange() | Mandatory   | See section 8.2.                                                                  |

1342 **10.4 CIM\_DiagnosticsLog**

1343 CIM\_DiagnosticsLog represents a log that aggregates all of the results (records) that the execution of a  
 1344 diagnostic generates. Table 38 provides information about the properties of CIM\_DiagnosticsLog.

1345 **Table 38 – Class: CIM\_DiagnosticsLog**

| Properties | Requirement | Notes                                                                                                                                                                         |
|------------|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| InstanceID | Mandatory   | <b>Key</b><br>InstanceID should be constructed using the following preferred algorithm:<br><OrgID>:<LocalID><br>(See the MOF file for more detail.)<br>(pattern "^.*[:].*\$") |
| ClearLog() | Mandatory   | See section 8.3.                                                                                                                                                              |

1346 **10.5 CIM\_DiagnosticServiceCapabilities**

1347 CIM\_DiagnosticServiceCapabilities publishes the diagnostic service's capabilities, such as settings and  
 1348 execution controls that are supported. Table 39 provides information about the properties of  
 1349 CIM\_DiagnosticServiceCapabilities.

1350 **Table 39 – Class: CIM\_DiagnosticServiceCapabilities**

| Properties | Requirement | Notes                                                                                                                                                                                             |
|------------|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| InstanceID | Mandatory   | <b>Key</b><br>InstanceID shall be unique and should be constructed using the following preferred algorithm:<br><OrgID>:<LocalID><br>(See the MOF file for more detail.)<br>(pattern "^.*[:].*\$") |

| Properties                                   | Requirement | Notes                                                                                                                                                                               |
|----------------------------------------------|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ElementName                                  | Optional    | This property shall contain the value of the Service's ElementName property.<br><br>The property will be formatted as a free-form string of variable length.<br><br>(pattern ".**") |
| SupportedServiceModes                        | Optional    | If service modes are supported, they shall be published using this property.                                                                                                        |
| OtherSupportedServiceModesDescriptions       | Conditional | If SupportedServiceModes has the value 1 (Other), this property is Mandatory.                                                                                                       |
| SupportedLoopControl                         | Optional    | If looping is supported, its controls shall be published using this property.                                                                                                       |
| OtherSupportedLoopControlDescriptions        | Conditional | If SupportedLoopControl has the value 1 (Other), this property is Mandatory.                                                                                                        |
| SupportedLogOptions                          | Optional    | If any log options are supported, they shall be published using this property.                                                                                                      |
| OtherSupportedLogOptionsDescriptions         | Conditional | If SupportedLogOptions has the value 1(Other), this property is Mandatory.                                                                                                          |
| SupportedLogStorage                          | Optional    | If any log storage options are supported, they shall be published using this property.                                                                                              |
| OtherSupportedLogStorageDescriptions         | Conditional | If SupportedLogStorage has the value 1 (Other), this property is Mandatory.                                                                                                         |
| SupportedExecutionControls                   | Optional    | If any execution controls are supported, they shall be published using this property.                                                                                               |
| OtherSupportedExecutionControls Descriptions | Conditional | If SupportedExecutionControls has the value 1 (Other), this property is Mandatory.                                                                                                  |

1351 **10.6 CIM\_DiagnosticServiceRecord**

1352 CIM\_DiagnosticServiceRecord is used to report diagnostic service messages such as results, errors,  
 1353 warnings, and status when CIM\_DiagnosticsLog is implemented. Table 40 provides information about the  
 1354 properties of CIM\_DiagnosticServiceRecord.

1355 **Table 40 – Class: CIM\_DiagnosticServiceRecord**

| Properties        | Requirement | Notes                                                                                                                                                                                                                                                                                                                                                                                                                             |
|-------------------|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| InstanceID        | Mandatory   | <b>Key</b><br><br>InstanceID should be constructed using the following preferred algorithm:<br><ConcreteJob.InstanceID>:<n> Where <ConcreteJob.InstanceID> is <OrgID>:<LocalID> as described in ConcreteJob and <n> is an increment value that provides uniqueness. <n> should be set to \"0\" for the first record created by the job, and incremented for each subsequent record.<br><br>(pattern "^.*[:].*[:][0123456789]*\$") |
| CreationTimeStamp | Mandatory   | None                                                                                                                                                                                                                                                                                                                                                                                                                              |
| RecordData        | Mandatory   | None                                                                                                                                                                                                                                                                                                                                                                                                                              |

| Properties   | Requirement | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|--------------|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| RecordFormat | Mandatory   | None                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| LoopsPassed  | Conditional | <p>This property shall contain the number of loops passed when RecordType is a "Results", 3 (Subtests), 7 (Device Errors) or 8 (Service Errors) at the time when the record was logged.</p> <p>This property may be NULL if RecordType is different than the aforementioned record types.</p>                                                                                                                                                                                                                                                                           |
| LoopsFailed  | Conditional | <p>This property shall contain the number of loops failed when RecordType is a "Results", 3 (Subtests), 7 (Device Errors) or 8 (Service Errors) at the time when the record was logged.</p> <p>This property may be NULL if RecordType is different than the aforementioned record types.</p>                                                                                                                                                                                                                                                                           |
| ErrorCode    | Conditional | <p>This property is Mandatory and shall contain only the error code number when RecordType is 7 (Device Errors) or 8 (Service Errors).</p> <p>If the RecordType value is 2 (Results), this property shall be an array that contains the error codes of all errors generated by the diagnostic service or subtest execution at the time when the record was logged.</p> <p>This property may be NULL if RecordType is different than the aforementioned record types.</p> <p>The property will be formatted as a free-form string of variable length. (pattern ".*")</p> |
| ErrorCount   | Conditional | <p>This property is Mandatory and shall contain only "1" in the first position of that array when RecordType is 7 (Device Errors) or 8 (Service Errors).</p> <p>This property is Mandatory when the RecordType value is 2 (Results), this property shall be an array where each position shall contain the number of times that an error (which can be identified by the same position of ErrorCode array) happened.</p> <p>This property may be NULL if RecordType is different than the aforementioned record types.</p>                                              |
| ServiceName  | Mandatory   | <p>This property shall be constructed as follows: &lt;OrgID&gt;:&lt;TestName&gt;. It should be set to the value of the associated DiagnosticTest.Name.</p> <p>(pattern "^.*[:].*\$")</p>                                                                                                                                                                                                                                                                                                                                                                                |

| Properties                 | Requirement | Notes                                                                                                                                                                        |
|----------------------------|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ManagedElementName         | Mandatory   | This property shall be formatted as a free-form string of variable length. It should be set to the value of the associated ManagedElement.ElementName.<br><br>(pattern ".*") |
| RecordType                 | Mandatory   | A RecordType value of 2 (Results) shall be used to report the final results.<br><br>A RecordType value of 6 (Status) shall be used to report interim results.                |
| OtherRecordTypeDescription | Conditional | If the RecordType value is 1 (Other), this property is Mandatory.                                                                                                            |
| ExpirationDate             | Mandatory   |                                                                                                                                                                              |

## 1356 10.7 CIM\_DiagnosticSetting

1357 Diagnostic services use CIM\_DiagnosticSetting to publish default settings, and clients use this class to  
 1358 change defaults and run a diagnostic service using specific settings. Table 41 provides information about  
 1359 the properties of CIM\_DiagnosticSetting.

1360

**Table 41 – Class: CIM\_DiagnosticSetting**

| Properties                   | Requirement | Notes                                                                                                                                                                                                                                                                                                                                            |
|------------------------------|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SettingID                    | Mandatory   | <b>Key</b>                                                                                                                                                                                                                                                                                                                                       |
| ElementName                  | Optional    | This property shall be formatted as a free-form string of variable length. (pattern ".*")                                                                                                                                                                                                                                                        |
| HaltOnError                  | Optional    | When this property is TRUE, the service should halt after finding the first error.                                                                                                                                                                                                                                                               |
| QuickMode                    | Optional    | When this property is TRUE, the service should attempt to run in an accelerated fashion either by reducing the coverage or number of tests performed.                                                                                                                                                                                            |
| PercentOfTestCoverage        | Optional    | This property requests the service to reduce test coverage to the specified percentage.                                                                                                                                                                                                                                                          |
| LoopControl                  | Optional    | This property, which is used in conjunction with LoopControlParameter property, sets one or more loop control mechanisms that limit the number of times that a test should be repeated.                                                                                                                                                          |
| LoopControlParameter         | Conditional | If LoopControl has a value other than null, LoopControlParameter shall be filled in for corresponding LoopControl settings.<br><br>If LoopControl matches "Count", "Timer", or "ErrorCount", LoopControlParameter represents a uint32 numeric value.<br><br>(pattern "^b[01]*   ^d[0123456789]*   ^x[0123456789ABCDEFabcdef]*   ^[0123456789]*") |
| OtherLoopControlDescriptions | Conditional | If LoopControl has the value 1 (Other), this property is Mandatory.                                                                                                                                                                                                                                                                              |

| Properties                  | Requirement | Notes                                                                                                                                                                                 |
|-----------------------------|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ResultPersistence           | Mandatory   | This property specifies how many seconds the records should persist after service execution finishes. 0 (zero) indicates "no persistence" and 0xFFFFFFFF indicates "persist forever". |
| LogOptions                  | Optional    | This property specifies the types of data that should be logged by the diagnostic service.                                                                                            |
| OtherLogOptionsDescriptions | Conditional | If LogOptions has the value 1 (Other), this property is Mandatory.                                                                                                                    |
| LogStorage                  | Optional    | This property specifies the logging mechanism to store the diagnostic results.                                                                                                        |
| OtherLogStorageDescriptions | Conditional | If LogStorage has the value 1 (Other), this property is Mandatory.                                                                                                                    |
| VerbosityLevel              | Optional    | This property specifies the desired volume or detail logged by a diagnostic service.                                                                                                  |
| Locales                     | Optional    | None                                                                                                                                                                                  |

1361 **10.8 CIM\_DiagnosticSettingRecord**

1362 CIM\_DiagnosticSettingRecord stores the settings used in a specific diagnostic service execution when  
 1363 CIM\_DiagnosticsLog is implemented. Table 42 provides information about the properties of  
 1364 CIM\_DiagnosticSettingRecord.

1365 **Table 42 – Class: CIM\_DiagnosticSettingRecord**

| Properties         | Requirement | Notes                                                                                                                                                                                                                                                                                                                                                                                                                          |
|--------------------|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| InstanceID         | Mandatory   | <b>Key</b><br>InstanceID should be constructed using the following preferred algorithm:<br><ConcreteJob.InstanceID>:<n><br><br>< ConcreteJob.InstanceID> is <OrgID>:<LocalID> as described in CIM_ConcreteJob, and <n> is an increment value that provides uniqueness. <n> should be set to \"0\" for the first record created by the job, and incremented for each subsequent record.<br>(pattern "^.*[:].*:[0123456789]*\$") |
| CreationTimeStamp  | Mandatory   | None                                                                                                                                                                                                                                                                                                                                                                                                                           |
| ServiceName        | Mandatory   | This property shall be constructed as follows: <OrgID>:<TestName>. It should be set to the value of the associated DiagnosticTest.Name.<br>(pattern "^.*[:].*\$")                                                                                                                                                                                                                                                              |
| ManagedElementName | Mandatory   | This property will be formatted as a free-form string of variable length. It should be set to the value of the associated ManagedElement.ElementName.<br>(pattern ".**")                                                                                                                                                                                                                                                       |

| Properties                   | Requirement | Notes                                                                                                                                                                                                                                                                                                                                            |
|------------------------------|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ExpirationDate               | Mandatory   |                                                                                                                                                                                                                                                                                                                                                  |
| HaltOnError                  | Optional    | When this property is TRUE, the service should halt after finding the first error.                                                                                                                                                                                                                                                               |
| QuickMode                    | Optional    | When this property is TRUE, the service should attempt to run in an accelerated fashion either by reducing the coverage or number of tests performed.                                                                                                                                                                                            |
| PercentOfTestCoverage        | Optional    | This property requests the service to reduce test coverage to the specified percentage.                                                                                                                                                                                                                                                          |
| LoopControl                  | Optional    | This property, which is used in conjunction with the LoopControlParameter property, sets one or more loop control mechanisms that limit the number of times that a test should be repeated.                                                                                                                                                      |
| LoopControlParameter         | Conditional | If LoopControl has a value other than null, LoopControlParameter shall be filled in for corresponding LoopControl settings.<br><br>If LoopControl matches "Count", "Timer", or "ErrorCount", LoopControlParameter represents a uint32 numeric value.<br><br>(pattern "^b[01]*   ^d[0123456789]*   ^x[0123456789ABCDEFabcdef]*   ^[0123456789]*") |
| OtherLoopControlDescriptions | Conditional | If LoopControl has the value 1 (Other), this property is Mandatory.                                                                                                                                                                                                                                                                              |
| ResultPersistence            | Mandatory   | This property specifies how many seconds the records should persist after service execution finishes. 0 (zero) indicates "no persistence" and 0xFFFFFFFF indicates "persist forever".                                                                                                                                                            |
| LogOptions                   | Optional    | This property specifies the types of data that should be logged by the diagnostic service.                                                                                                                                                                                                                                                       |
| OtherLogOptionsDescriptions  | Conditional | If LogOptions has the value 1 (Other), this property is Mandatory.                                                                                                                                                                                                                                                                               |
| VerbosityLevel               | Optional    | This property specifies the desired volume or detail logged by a diagnostic service.                                                                                                                                                                                                                                                             |

## 1366 10.9 CIM\_DiagnosticTest

1367 CIM\_DiagnosticTest is a class that represents a diagnostic service developed to exercise and observe  
1368 the behavior of a device that is implicated in some level of system malfunction. It contains properties  
1369 useful in test configuration and the RunDiagnostic( ) method, a standard mechanism for invoking the test.

1370 Table 43 provides information about the properties of CIM\_DiagnosticTest.

1371

**Table 43 – Class: CIM\_DiagnosticTest**

| Properties              | Requirement | Notes      |
|-------------------------|-------------|------------|
| SystemCreationClassName | Mandatory   | <b>Key</b> |
| SystemName              | Mandatory   | <b>Key</b> |
| CreationClassName       | Mandatory   | <b>Key</b> |

| Properties                       | Requirement | Notes                                                                                                                 |
|----------------------------------|-------------|-----------------------------------------------------------------------------------------------------------------------|
| Name                             | Mandatory   | <b>Key</b><br>The Name property shall be constructed as follows:<br><OrgID>:<TestName>.<br><br>(pattern "^.*[:].*\$") |
| ElementName                      | Mandatory   | The property will be formatted as a free-form string of variable length.<br>(pattern ".*")                            |
| Characteristics                  | Mandatory   |                                                                                                                       |
| OtherCharacteristicsDescriptions | Conditional | If Characteristics has the value 1 (Other), this property is Mandatory.                                               |
| RunDiagnostic()                  | Mandatory   | See section 8.1.                                                                                                      |

1372 **10.10 CIM\_ElementCapabilities**

1373 CIM\_ElementCapabilities associates a diagnostic service with its capabilities. Table 44 provides  
1374 information about the properties of CIM\_ElementCapabilities.

1375 **Table 44 – Class: CIM\_ElementCapabilities**

| Properties     | Requirement | Notes                                                                                                                      |
|----------------|-------------|----------------------------------------------------------------------------------------------------------------------------|
| ManagedElement | Mandatory   | <b>Key</b> This property shall be a reference to an instance of CIM_DiagnosticService.                                     |
| Capabilities   | Mandatory   | <b>Key</b> This property shall be a reference to an instance of CIM_DiagnosticServiceCapabilities.<br><br>Cardinality 0..1 |

1376 **10.11 CIM\_DefaultSetting (DiagnosticSetting)**

1377 CIM\_DefaultSetting associates the diagnostic service with the settings for the service itself and the  
1378 resulting job. Table 45 provides information about the properties of CIM\_DefaultSetting.

1379 **Table 45 – Class: CIM\_DefaultSetting (DiagnosticSetting)**

| Properties | Requirement | Notes                                                                                                          |
|------------|-------------|----------------------------------------------------------------------------------------------------------------|
| Element    | Mandatory   | <b>Key</b> This property shall be a reference to an instance of CIM_DiagnosticService.<br><br>Cardinality 1    |
| Setting    | Mandatory   | <b>Key</b> This property shall be a reference to an instance of CIM_DiagnosticSetting.<br><br>Cardinality 0..1 |

1380 **10.12 CIM\_DefaultSetting (JobSettingData)**

1381 CIM\_DefaultSetting associates the diagnostic service with the settings for the service itself and the  
 1382 resulting job. Table 46 provides information about the properties of CIM\_DefaultSetting.

1383 **Table 46 – Class: CIM\_DefaultSetting (JobSettingData)**

| Properties | Requirement | Notes                                                                                                   |
|------------|-------------|---------------------------------------------------------------------------------------------------------|
| Element    | Mandatory   | <b>Key</b> This property shall be a reference to an instance of CIM_DiagnosticService.<br>Cardinality 1 |
| Setting    | Mandatory   | <b>Key</b> This property shall be a reference to an instance of CIM_JobSettingData.<br>Cardinality 0..1 |

1384 **10.13 CIM\_HelpService**

1385 CIM\_HelpService is the preferred way for a service to publish online help information. Table 47 provides  
 1386 information about the properties of CIM\_HelpService.

1387 **Table 47 – Class: CIM\_HelpService**

| Properties                     | Requirement | Notes                                                                                                  |
|--------------------------------|-------------|--------------------------------------------------------------------------------------------------------|
| SystemCreationClassName        | Mandatory   | <b>Key</b>                                                                                             |
| SystemName                     | Mandatory   | <b>Key</b>                                                                                             |
| CreationClassName              | Mandatory   | <b>Key</b>                                                                                             |
| Name                           | Mandatory   | <b>Key</b><br>This property will be formatted as a free-form string of variable length. (pattern ".*") |
| ElementName                    | Mandatory   | This property will be formatted as a free-form string of variable length. (pattern ".*")               |
| DeliveryOptions                | Mandatory   | None                                                                                                   |
| OtherDeliveryOptionDescription | Conditional | If DeliveryOptions has the value 1 (Other), this property is Mandatory.                                |
| DocumentsAvailable             | Mandatory   | This property will be formatted as a free-form string of variable length. (pattern ".*")               |
| DocumentDescriptions           | Mandatory   | None                                                                                                   |
| DocumentFormat                 | Mandatory   | None                                                                                                   |
| OtherDocumentFormatDescription | Conditional | If DocumentFormat has the value 1 (Other), this property is Mandatory.                                 |
| GetHelp()                      | Mandatory   | See section 8.4.                                                                                       |

1388 **10.14 CIM\_JobSettingData**

1389 Diagnostic services use CIM\_JobSettingData to publish default job settings (for the jobs that they launch),  
 1390 and clients use this class to change the default job settings when invoking the RunDiagnostic( ) method.  
 1391 Table 48 provides information about the properties of CIM\_JobSettingData.

1392 **Table 48 – Class: CIM\_JobSettingData**

| Properties         | Requirement | Notes                                                                                     |
|--------------------|-------------|-------------------------------------------------------------------------------------------|
| InstanceID         | Mandatory   | <b>Key</b>                                                                                |
| ElementName        | Mandatory   | This property shall be formatted as a free-form string of variable length. (pattern ".*") |
| DeleteOnCompletion | Mandatory   |                                                                                           |

1393 **10.15 CIM\_LogManagesRecord**

1394 CIM\_LogManagesRecord associates a DiagnosticsLog with its records (service records, setting records,  
 1395 or completion records). Table 49 provides information about the properties of CIM\_LogManagesRecord.

1396 **Table 49 – Class: CIM\_LogManagesRecord**

| Properties | Requirement | Notes                                                                                 |
|------------|-------------|---------------------------------------------------------------------------------------|
| Log        | Mandatory   | <b>Key</b> This property shall be a reference to an instance of CIM_Log.              |
| Record     | Mandatory   | <b>Key</b> This property shall be a reference to an instance of CIM_DiagnosticRecord. |

1397 **10.16 CIM\_OwningJobElement**

1398 CIM\_OwningJobElement associates a diagnostic service with its jobs (jobs that are launched by this  
 1399 diagnostic). Table 50 provides information about the properties of CIM\_OwningJobElement.

1400 **Table 50 – Class: CIM\_OwningJobElement**

| Properties    | Requirement | Notes                                                                                                   |
|---------------|-------------|---------------------------------------------------------------------------------------------------------|
| OwningElement | Mandatory   | <b>Key</b> This property shall be a reference to an instance of CIM_DiagnosticService.<br>Cardinality 1 |
| OwnedElement  | Mandatory   | <b>Key</b> This property shall be a reference to an instance of CIM_ConcreteJob.                        |

1401 **10.17 CIM\_RecordAppliesToElement**

1402 CIM\_RecordAppliesToElement associates a record with the managed elements (diagnostic service and  
 1403 device) that have a relationship with this record. Table 51 provides information about the properties of  
 1404 CIM\_RecordAppliesToElement.

1405 **Table 51 – Class: CIM\_RecordAppliesToElement**

| Properties | Requirement | Notes                                                                                 |
|------------|-------------|---------------------------------------------------------------------------------------|
| Antecedent | Mandatory   | <b>Key</b> This property shall be a reference to an instance of CIM_DiagnosticRecord. |
| Dependent  | Mandatory   | <b>Key</b> This property shall be a reference to an instance of CIM_ManagedElement.   |

1406 **10.18 CIM\_ServiceAffectsElement**

1407 CIM\_ServiceAffectsElement is used to associate to the diagnostic service any managed elements that  
 1408 are affected by the running of the service. Table 52 provides information about the properties of  
 1409 CIM\_ServiceAffectsElement.

1410 **Table 52 – Class: CIM\_ServiceAffectsElement**

| Properties       | Requirement | Notes                                                                                  |
|------------------|-------------|----------------------------------------------------------------------------------------|
| AffectedElement  | Mandatory   | <b>Key</b> This property shall be a reference to an instance of CIM_ManagedElement.    |
| AffectingElement | Mandatory   | <b>Key</b> This property shall be a reference to an instance of CIM_DiagnosticService. |

1411 **10.19 CIM\_ServiceAvailableToElement**

1412 CIM\_ServiceAvailableToElement associates the diagnostic service with its help service information. Table  
 1413 53 provides information about the properties of CIM\_ServiceAvailableToElement.

1414 **Table 53 – Class: CIM\_ServiceAvailableToElement**

| Properties      | Requirement | Notes                                                                                                   |
|-----------------|-------------|---------------------------------------------------------------------------------------------------------|
| ServiceProvided | Mandatory   | <b>Key</b> This property shall be a reference to an instance of CIM_HelpService.<br>Cardinality 1       |
| UserOfService   | Mandatory   | <b>Key</b> This property shall be a reference to an instance of CIM_DiagnosticService.<br>Cardinality 1 |

1415 **10.20 CIM\_ServiceComponent**

1416 CIM\_ServiceComponent associates a test that is also part of another test. Table 54 provides information  
 1417 about the properties of CIM\_ServiceComponent.

1418 **Table 54 – Class: CIM\_ServiceComponent**

| Properties     | Requirement | Notes                                                                                  |
|----------------|-------------|----------------------------------------------------------------------------------------|
| GroupComponent | Mandatory   | <b>Key</b> This property shall be a reference to an instance of CIM_DiagnosticService. |
| PartComponent  | Mandatory   | <b>Key</b> This property shall be a reference to an instance of CIM_DiagnosticService. |

1419 **10.21 CIM\_ElementSettingData**

1420 CIM\_ElementSettingData associates the diagnostic service with the settings for the resulting job. Table  
 1421 55 provides information about the properties of CIM\_ElementSettingData.

1422 **Table 55 – Class: CIM\_ElementSettingData**

| Properties     | Requirement | Notes                                                                                                   |
|----------------|-------------|---------------------------------------------------------------------------------------------------------|
| ManagedElement | Mandatory   | <b>Key</b> This property shall be a reference to an instance of CIM_DiagnosticService.<br>Cardinality 1 |
| SettingData    | Mandatory   | <b>Key</b> This property shall be a reference to an instance of CIM_JobSettingData<br>Cardinality 0..1  |
| IsDefault      | Mandatory   | None                                                                                                    |

1423 **10.22 CIM\_UseOfLog**

1424 CIM\_UseOfLog associates a log with a managed element (a device or diagnostic service) whose  
 1425 information is stored in the log. Table 56 provides information about the properties of CIM\_UseOfLog.

1426 **Table 56 – Class: CIM\_UseOfLog**

| Properties | Requirement | Notes                                                                                  |
|------------|-------------|----------------------------------------------------------------------------------------|
| Antecedent | Mandatory   | <b>Key</b> This property shall be a reference to an instance of CIM_Log.               |
| Dependent  | Mandatory   | <b>Key</b> This property shall be a reference to an instance of CIM_DiagnosticService. |

1427 **10.23 CIM\_CorrespondingSettingsRecord**

1428 CIM\_RecordAppliesToElement associates a setting record with the service records related to this setting  
 1429 record. Table 57 provides information about the properties of CIM\_CorrespondingSettingsRecord.

1430 **Table 57 – Class: CIM\_CorrespondingSettingsRecord**

| Properties     | Requirement | Notes                                                                                        |
|----------------|-------------|----------------------------------------------------------------------------------------------|
| DataRecord     | Mandatory   | <b>Key</b> This property shall be a reference to an instance of CIM_DiagnosticRecord.        |
| SettingsRecord | Mandatory   | <b>Key</b> This property shall be a reference to an instance of CIM_DiagnosticSettingRecord. |

1431 **10.24 CIM\_HostedService (DiagnosticTest)**

1432 CIM\_HostedService is used to associate an instance of CIM\_DiagnosticTest with an instance of  
 1433 CIM\_ComputerSystem to which the CIM\_DiagnosticTest is scoped. Table 58 provides information about  
 1434 the properties of CIM\_HostedService (DiagnosticTest).

1435 **Table 58 – Class: CIM\_HostedService (DiagnosticTest)**

| Properties | Requirement | Notes                                                                                                   |
|------------|-------------|---------------------------------------------------------------------------------------------------------|
| Antecedent | Mandatory   | <b>Key</b> This property shall be a reference to an instance of CIM_ComputerSystem.<br>Cardinality 1    |
| Dependent  | Mandatory   | <b>Key</b> This property shall be a reference to an instance of CIM_DiagnosticTest.<br>Cardinality 1..* |

1436 **10.25 CIM\_HostedService (HelpService)**

1437 CIM\_HostedService is used to associate an instance of CIM\_HelpService with an instance of  
 1438 CIM\_ComputerSystem to which the CIM\_HelpService is scoped. Table 59 provides information about the  
 1439 properties of CIM\_HostedService.

1440 **Table 59 – Class: CIM\_HostedService (HelpService)**

| Properties | Requirement | Notes                                                                                                |
|------------|-------------|------------------------------------------------------------------------------------------------------|
| Antecedent | Mandatory   | <b>Key</b> This property shall be a reference to an instance of CIM_ComputerSystem.<br>Cardinality 1 |
| Dependent  | Mandatory   | <b>Key</b> This property shall be a reference to an instance of CIM_HelpService.<br>Cardinality 1..* |

1441 **10.26 CIM\_RegisteredProfile**

1442 CIM\_RegisteredProfile identifies the *Diagnostics Profile* in order for a client to determine whether an  
 1443 instance of CIM\_DiagnosticService is conformant with this profile. The CIM\_RegisteredProfile class is  
 1444 defined by the *Profile Registration Profile*. With the exception of the mandatory values specified in Table  
 1445 60, the behavior of the CIM\_RegisteredProfile instance is in accordance with the *Profile Registration*  
 1446 *Profile*.

1447 **Table 60 – Class: CIM\_RegisteredProfile**

| Properties             | Requirement | Notes                                              |
|------------------------|-------------|----------------------------------------------------|
| RegisteredName         | Mandatory   | This property shall have a value of "Diagnostics". |
| RegisteredVersion      | Mandatory   | This property shall have a value of "1.0.0".       |
| RegisteredOrganization | Mandatory   | This property shall have a value of 2 (DMTF).      |

1448

1449

1450  
1451  
1452  
1453

## **ANNEX A** (informative)

### **Change Log**

| <b>Version</b> | <b>Date</b> | <b>Description</b> |
|----------------|-------------|--------------------|
| 1.0.0a         | 07/12/07    | Preliminary        |
| 1.0.0          | 02/13/09    | Standard           |
| 1.0.1          | 09/23/09    | Errata             |

1454 **ANNEX B**  
1455 (informative)

1456  
1457  
1458 **Acknowledgements**

1459 The authors wish to acknowledge the following people.

1460 Editors:

- 1461 • Andre Asselin – IBM Corporation
- 1462 • Mateus Baur – Hewlett-Packard Company
- 1463 • Barbara Craig – Hewlett-Packard Company
- 1464 • Carl Chan – WBEM Solutions, Inc.

1465 Contributors:

- 1466 • Jim Davis – WBEM Solutions, Inc.
- 1467 • Members of the DMTF Diagnostics SIG (sub-Working Group of CIM Core)

## Bibliography

1468

1469 DMTF DSP1004, [Base Server Profile 1.0.0](#),  
1470 [http://www.dmtf.org/standards/published\\_documents/DSP1004.pdf](http://www.dmtf.org/standards/published_documents/DSP1004.pdf)

1471 DMTF DSP2000, [CIM Diagnostic Model White Paper 1.0.0](#),  
1472 [http://www.dmtf.org/standards/published\\_documents/DSP2000.pdf](http://www.dmtf.org/standards/published_documents/DSP2000.pdf)  
1473