



1
2
3
4

Document Number: DSP1119

Date: 2012-09-12

Version: 1.0.0a

5 **Diagnostic Job Control Profile**

Information for Work-in-Progress version:

IMPORTANT: This specification is not a standard. It does not necessarily reflect the views of the DMTF or all of its members. Because this document is a Work in Progress, this specification may still change, perhaps profoundly. This document is available for public review and comment until the stated expiration date.

It expires on: 2013-02-01

Target version for DMTF Standard: 2.34.0

Provide any comments through the DMTF Feedback Portal:

<http://www.dmtf.org/standards/feedback>

6 **Document Type: Specification**
7 **Document Status: Work In Progress**
8 **Document Language: en-US**

9 Copyright notice

10 Copyright © 2012 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25 withdrawn or modified after publication, and shall be indemnified and held harmless by any party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

| | | |
|----|---|----|
| 32 | Introduction..... | 6 |
| 33 | 1 Scope | 7 |
| 34 | 2 Normative references | 7 |
| 35 | 3 Terms and definitions | 8 |
| 36 | 4 Symbols and abbreviated terms..... | 8 |
| 37 | 5 Synopsis | 9 |
| 38 | 6 Description | 10 |
| 39 | 7 Implementation..... | 12 |
| 40 | 7.1 CIM_AffectedJobElement | 12 |
| 41 | 7.2 CIM_ConcreteJob | 12 |
| 42 | 7.3 CIM_DiagnosticServiceJobCapabilities | 14 |
| 43 | 7.4 CIM_JobSettingData (Client) | 16 |
| 44 | 7.5 CIM_JobSettingData (Default) | 18 |
| 45 | 7.6 Interactive options..... | 20 |
| 46 | 7.7 Job deletion options | 23 |
| 47 | 8 Methods..... | 25 |
| 48 | 8.1 Profile conventions for operations | 25 |
| 49 | 8.2 CIM_ConcreteJob | 26 |
| 50 | 8.3 CIM_DiagnosticServiceJobCapabilities | 28 |
| 51 | 8.4 CIM_MethodResult | 29 |
| 52 | 8.5 CIM_OwningJobElement | 29 |
| 53 | 8.6 CIM_AffectedJobElement | 29 |
| 54 | 8.7 CIM_AssociatedJobMethodResult..... | 29 |
| 55 | 8.8 CIM_HostedDependency..... | 29 |
| 56 | 8.9 CIM_RegisteredProfile..... | 29 |
| 57 | 8.10 CIM_JobSettingData..... | 29 |
| 58 | 8.11 CIM_ElementSettingdData | 30 |
| 59 | 8.12 CIM_ElementCapabilities | 30 |
| 60 | 8.13 CIM_DiagnosticTest.RunDiagnosticService() | 30 |
| 61 | 9 Use cases (Informative) | 31 |
| 62 | 9.1 Use case summary | 31 |
| 63 | 9.2 User input required | 33 |
| 64 | 9.3 User action required..... | 36 |
| 65 | 9.4 Silent mode operation | 38 |
| 66 | 9.5 Finding diagnostic jobs | 38 |
| 67 | 9.6 Configuring a diagnostic job | 39 |
| 68 | 9.7 Execute and control a job for a diagnostic test | 39 |
| 69 | 9.8 Delete a job..... | 42 |
| 70 | 10 CIM elements | 43 |
| 71 | 10.1 CIM_AffectedJobElement | 44 |
| 72 | 10.2 CIM_ConcreteJob | 45 |
| 73 | 10.3 CIM_DiagnosticServiceJobCapabilities | 45 |
| 74 | 10.4 CIM_ElementCapabilities (Job) | 46 |
| 75 | 10.5 CIM_ElementSettingData (Default JobSettingData)..... | 46 |
| 76 | 10.6 CIM_HostedDependency..... | 47 |
| 77 | 10.7 CIM_JobSettingData (Client) | 47 |
| 78 | 10.8 CIM_JobSettingData (Default) | 47 |
| 79 | 10.9 CIM_OwningJobElement | 48 |
| 80 | 10.10 CIM_RegisteredProfile..... | 48 |
| 81 | Annex A (informative) Change log | 49 |
| 82 | | |

83 **Figures**

84 Figure 1 – Diagnostic Job Control Profile: Profile class diagram 11

85

86 **Tables**

| | | |
|-----|--|----|
| 87 | Table 1 – Referenced profiles | 9 |
| 88 | Table 2 – OperationalStatus to JobState mapping | 13 |
| 89 | Table 3 – Interactive options | 22 |
| 90 | Table 4 – Job deletion options | 25 |
| 91 | Table 5 – ResumeWithInput() method: Return code values | 27 |
| 92 | Table 6 – ResumeWithInput() method: Parameters | 27 |
| 93 | Table 7 – ResumeWithAction() method: Return code values | 28 |
| 94 | Table 8 – CreateGoalSettings() method: Return code values | 28 |
| 95 | Table 9 – CreateGoalSettings() method: Parameters | 29 |
| 96 | Table 10 – Operations: CIM_JobSettingData | 30 |
| 97 | Table 11 – Operations: CIM_ElementSettingData | 30 |
| 98 | Table 12 – Operations: CIM_ElementCapabilities | 30 |
| 99 | Table 13 – Job setting options | 31 |
| 100 | Table 14 – Diagnostic test use cases | 32 |
| 101 | Table 15 – CIM Elements: Diagnostic Job Control Profile | 43 |
| 102 | Table 16 – Class: CIM_AffectedJobElement | 45 |
| 103 | Table 17 – Class: CIM_ConcreteJob | 45 |
| 104 | Table 18 – Class: CIM_DiagnosticServiceJobCapabilities | 46 |
| 105 | Table 19 – Class: CIM_ElementCapabilities | 46 |
| 106 | Table 20 – Class: CIM_ElementSettingData | 46 |
| 107 | Table 21 – Class: CIM_HostedDependency | 47 |
| 108 | Table 22 – Class: CIM_JobSettingData (Client) | 47 |
| 109 | Table 23 – Class: CIM_JobSettingData (Default) | 47 |
| 110 | Table 24 – Class: CIM_OwningJobElement | 48 |
| 111 | Table 25 – Class: CIM_RegisteredProfile | 48 |

112

113

114

Foreword

115 The *Diagnostic Job Control Profile* (DSP1119) was prepared by the Diagnostics Working Group of the
116 DMTF.

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

119 Acknowledgments

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

- 121 • Dave Barrett – Emulex Corporation
- 122 • Rodney Brown – IBM Corporation
- 123 • Carl Chan – WBEM Solutions, Inc.
- 124 • Peter Lamanna – EMC Corporation
- 125 • Mike Walker – Storage Networking Industry Association

126

Introduction

127 A *profile* is a collection of Common Information Model (CIM) elements and behavior rules that represents
128 a specific area of management. The purpose of the profile is to ensure interoperability of Web-Based
129 Enterprise Management (WBEM) services for a specific subset of the CIM schema — in this case,
130 Diagnostic Job Control.

131 The goal of the *Diagnostic Job Control Profile* is to define industry-standard building blocks that enable
132 management diagnostic tests running in a standard job infrastructure. The *Diagnostic Job Control Profile*
133 extends the *Job Control Profile* ([DSP1103](#)) by identifying a set of job control functions that should be
134 included in provider implementations.

135 Document conventions

136 Typographical conventions

137 The following typographical conventions are used in this document:

- 138 • Document titles are marked in *italics*.
- 139 • Important terms that are used for the first time are marked in *italics*.

140 ABNF usage conventions

141 Format definitions in this document are specified using ABNF (see [RFC5234](#)), with the following
142 deviations:

- 143 • Literal strings are to be interpreted as case-sensitive Unicode characters, as opposed to the
144 definition in [RFC5234](#) that interprets literal strings as case-insensitive US-ASCII characters.

145 Experimental material

146 Experimental material has yet to receive sufficient review to satisfy the adoption requirements set forth by
147 the DMTF. Experimental material is included in this document as an aid to implementers who are
148 interested in likely future developments. Experimental material may change as implementation
149 experience is gained. It is likely that experimental material will be included in an upcoming revision of the
150 specification. Until that time, experimental material is purely informational.

151 The following typographical convention indicates experimental material:

152 **EXPERIMENTAL**

153 Experimental material appears here.

154 **EXPERIMENTAL**

155 In places where this typographical convention cannot be used (for example, tables or figures), the
156 "EXPERIMENTAL" label is used alone.

157

Diagnostic Job Control Profile

158 1 Scope

159 The *Diagnostic Job Control Profile* specializes the *Job Control Profile* ([DSP1103](#)) by defining the job
160 control functions used to monitor and interact with diagnostic tests.

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

163 2 Normative references

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

168 DMTF DSP0004, *CIM Infrastructure Specification 2.6*,
169 http://dmf.org/sites/default/files/standards/documents/DSP0004_2.6.pdf

170 DMTF DSP0200, *CIM Operations over HTTP 1.3*,
171 http://dmf.org/sites/default/files/standards/documents/DSP0200_1.3.pdf

172 DMTF DSP1001, *Management Profile Specification Usage Guide 1.0*,
173 http://dmf.org/sites/default/files/standards/documents/DSP1001_1.0.pdf

174 DMTF DSP1002, *Diagnostics Profile Specification 2.0*,
175 http://dmf.org/sites/default/files/standards/documents/DSP1002_2.0.pdf

176 DMTF DSP1033, *Profile Registration Profile 1.0*,
177 http://dmf.org/sites/default/files/standards/documents/DSP1033_1.0.pdf

178 DMTF DSP1054, *Indications Profile 1.2*,
179 http://dmf.org/sites/default/files/standards/documents/DSP1054_1.2.pdf

180 DMTF DSP1103, *Job Control Profile 1.0.0*,
181 http://dmf.org/sites/default/files/standards/documents/DSP1103_1.0.pdf

182 DMTF DSP1105, *CPU Diagnostics Profile 1.0*
183 http://dmf.org/sites/default/files/standards/documents/DSP1105_1.0.pdf

184 DMTF DSP8055, *Diagnostics Message Registry 1.0.0a*,
185 http://dmf.org/sites/default/files/standards/documents/DSP8055_1.0a.xml

186 IETF RFC5234, *ABNF: Augmented BNF for Syntax Specifications, January 2008*,
187 <http://tools.ietf.org/html/rfc5234>

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

190 3 Terms and definitions

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

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

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

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

204 The terms defined in [DSP0004](#), [DSP0200](#), and [DSP1001](#) apply to this document.

205 3.1

206 **job**

207 a task that takes some time to execute

208 3.2

209 **organization**

210 consortium, standards group, or company creating a DMTF profile specification

211 4 Symbols and abbreviated terms

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

213 4.1

214 **CDM**

215 Common Diagnostic Model

216 4.2

217 **CIM**

218 Common Information Model

219 4.3

220 **CIMOM**

221 CIM Object Manager

222 4.4

223 **CQL**

224 CIM Query Language

225 4.5

226 **ME**

227 Managed Element

228 4.6

- 229 **MOF**
- 230 Managed Object Format
- 231 **4.7**
- 232 **OS**
- 233 Operating System
- 234 **4.8**
- 235 **QoS**
- 236 Quality of Service
- 237 **4.9**
- 238 **URI**
- 239 Uniform Resource Identifier
- 240 **4.10**
- 241 **WBEM**
- 242 Web-Based Enterprise Management

243 **5 Synopsis**

244 **Profile Name:** Diagnostics Job Control

245 **Version:** 1.0.0a

246 **Organization:** DMTF

247 **CIM schema version:** 2.34

248 **Central Class:** CIM_ConcreteJob

249 **Scoping Class:** CIM_System

250 **Specializes:** Job Control Profile 1.0.0

251 The *Diagnostic Job Control Profile* extends the management capability of referencing profiles by adding
 252 common methods for managing the jobs associated with diagnostic tests that are run on a managed
 253 system

254 The Central Instance of this profile shall be an instance of CIM_ConcreteJob. The Scoping Instance shall
 255 be the instance of CIM_System (the central instance of the referencing profile) with which the Central
 256 Instance (the instance of CIM_ConcreteJob) is associated through CIM_HostedDependency. The
 257 CIM_System is the system running the CIM_DiagnosticTest and its associated CIM_ConcreteJob.

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

259 **Table 1 – Referenced profiles**

| Profile Name | Organization | Version | Description |
|----------------------|--------------|---------|-------------|
| Job Control | DMTF | 1.0 | Specializes |
| Indications | DMTF | 1.2 | Mandatory |
| Diagnostics | DMTF | 2.0 | Mandatory |
| Profile Registration | DMTF | 1.0 | Mandatory |

260 6 Description

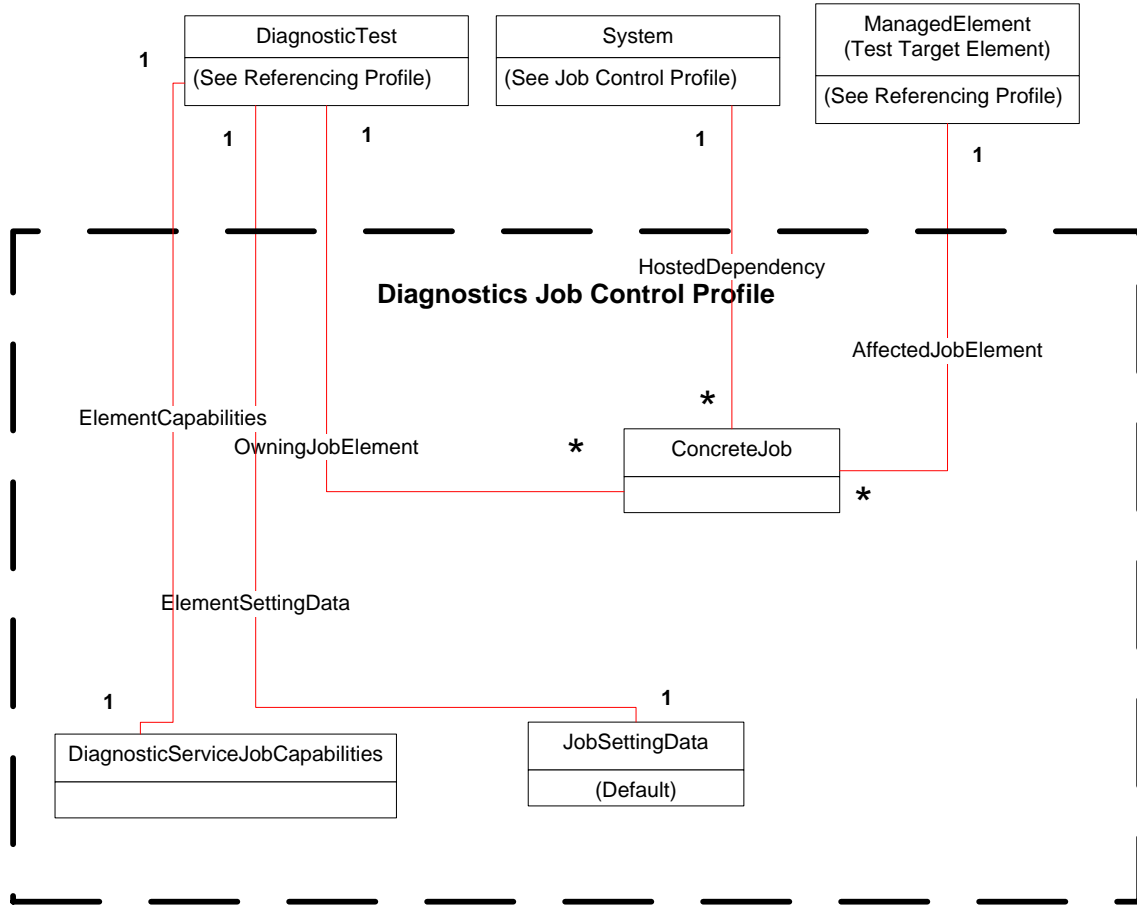
261 The *Diagnostics Profile* ([DSP1002](#)) defines the behavior and interfaces to be used for running and
262 monitoring diagnostic tests and reviewing their results. [DSP1103](#) defines the behavior and interfaces to
263 be used for running and monitoring jobs associated with those executing tests. This profile extends and
264 constrains the elements of the [DSP1103](#) elements that have diagnostic test-specific behavior and
265 interfaces.

266 Specifically, a user invokes `CIM_DiagnosticTest.RunDiagnosticService()` to start a diagnostic test. A user
267 may optionally pass an embedded instance of `CIM_JobSettingData` as an input parameter to specify the
268 behavior of the associated `CIM_ConcreteJob` instance. A `CIM_ConcreteJob` instance is created when a
269 diagnostic test starts. When the `CIM_ConcreteJob` instance is deleted is controlled by the values of the
270 properties in `CIM_ConcreteJob` and `CIM_JobSettingData`.

271 To start a diagnostic test, the client calls `CIM_DiagnosticTest.RunDiagnosticService()` which returns 0
272 (Success) and the objectpath of the `CIM_ConcreteJob` instance. Thereafter, `CIM_ConcreteJob` manages
273 the diagnostic test execution. Additionally, a client monitors and interacts with the diagnostic test
274 execution or workflow via the returned `CIM_ConcreteJob` instance.

275 Some diagnostic tests may launch other diagnostic tests. Others may require user interaction. The
276 `CIM_JobSettingData` instance contains properties that define the behavior for interactive diagnostic tests.

277 When a diagnostic test starts, only one `CIM_ConcreteJob` instance is created. When the diagnostic test
278 completes, its `CIM_ConcreteJob` instance will persist for a predetermined length of time before deletion.
279 The same diagnostic test could start again, creating another `CIM_ConcreteJob` instance before the
280 previous `CIM_ConcreteJob` instance is deleted. For this reason, the cardinality of
281 `CIM_OwningJobElement` is one-to-many.
282



283

284

Figure 1 – Diagnostic Job Control Profile: Profile class diagram

285 The Referencing Profile for DiagnosticTest is a component profile. For example, if the Referencing Profile
 286 is the *CPU Diagnostics Profile* ([DSP1105](#)) DiagnosticTest is a subclass called CPUDiagnosticTest.
 287 Similarly, if the Referencing Profile for ManagedElement is the same component profile as for
 288 DiagnosticTest. Thus, the [DSP1105](#) ManagedElement is CIM_Processor, CIM_ProcessorCore or
 289 CIM_HardwareThread.

290

291 7 Implementation

292 This clause details the requirements related to the arrangement of instances and their properties for
293 implementations of this profile.

294 7.1 CIM_AffectedJobElement

295 This subclause defines the properties of the CIM_AffectedJobElement class.

296 7.1.1 CIM_AffectedJobElement.ElementEffects

297 This optional property shall include only the following values: 1 (Other), 2 (Exclusive Use), 3
298 (Performance Impact), 4 (Element Integrity), and 5 (Create). If 1 (Other) is specified, the
299 OtherElementEffectsDescriptions shall have a value.

300 7.2 CIM_ConcreteJob

301 This subclause indicates the properties of the CIM_ConcreteJob class. Each execution of a test will
302 create an instance of CIM_ConcreteJob so that a client can track the progress and control the execution
303 of the diagnostic. To quickly and directly find the CIM_ConcreteJob instance of a executing test, a client
304 should retain the value of the Job output parameter returned by
305 CIM_DiagnosticTest.RunDiagnosticService() when the test is started.

306 7.2.1 CIM_ConcreteJob.InstanceID

307 This string property is the key property for this class. It should be constructed using the following
308 preferred algorithm:

309 <OrgID>:<LocalID>

310 where <OrgID> identifies the business entity (e.g. ACME) and <LocalID> is a value that uniquely
311 identifies each ConcreteJob instance that is launched on a system when a test is executed. See the MOF
312 file description for further information.

313 The purpose for <LocalID> is to provide some form of uniqueness within the context of running separate
314 diagnostic tests over a period of time for the domain of the test execution (whether just the local system
315 or several remote systems). In practice, <LocalID> could be an incremented counter or a timestamp in
316 combination with other test identifiers or factors.

317 A unique <LocalID> allows a user to easily retrieve test results from the diagnostic log for a specific test
318 execution because the InstanceID values of CIM_ConcreteJob, and the subclasses of
319 CIM_DiagnosticRecord are closely related. See Figure 5 in [DSP1002](#) and use case 9.8.4
320 GetDiagnosticResults for further information.

321 Specifically, CIM_DiagnosticRecord.InstanceID has the same value as its related
322 CIM_ConcreteJob.InstanceID with an appended record number identifier. As an example, if
323 CIM_ConcreteJob.InstanceID has the form "Widget:<StartTime>", for the third record,
324 CIM_DiagnosticRecord.InstanceID has the form "Widget:<StartTime>:3", where <StartTime> is the value
325 of CIM_ConcreteJob.StartTime.

326 7.2.2 CIM_ConcreteJob.Name

327 The value of this string property shall correspond to the value of the Name property of its associated
328 CIM_DiagnosticTest instance.

329 **7.2.3 CIM_ConcreteJob.JobState**

330 As defined in [DSP1103](#), this enumerated integer may have the values of 2 (New), 3 (Starting), 4
 331 (Running), 5 (Suspended), 6 (Shutting Down), 7 (Completed), 8 (Terminated), 9 (Killed), 10 (Exception).
 332 See Table 3 in [DSP1103](#) for further information. For this profile, 12 (Query Pending) is also permitted to
 333 provide the ability for a client to interact with a diagnostic test. The job changes the value JobState to 12
 334 (Query Pending) when it sends an AlertIndication to the client requesting input or action. The job changes
 335 the JobState from 12 (Query Pending) when it successfully receives a ResumeWithInput() or
 336 ResumeWithAction() request, or the client fails to respond within the
 337 CIM_JobSettingData.InteractiveTimeout period.

338 On a successful ResumeWithInput() or ResumeWithAction invocation(), the job changes JobState to 4
 339 (Running). If the extrinsic method fails, then the job may wait for a client retry. If the job waits for a client
 340 retry, it would stay in the 12 (Query Pending) state. If the client exceeds the number of retries (see
 341 CIM_JobSettingData.ClientRetries) or the CIM_ConcreteJob.InteractiveTimeout expires the job may
 342 terminate and set JobState to 8 (Terminated).

343 Table 4 in [DSP1103](#) defines the mapping of values between OperationalStatus and JobState. Table 2
 344 defines the additional mapping for this profile.

345 **Table 2 – OperationalStatus to JobState mapping**

| Operational Status | JobState | Description |
|--------------------|--------------------|---|
| 2 (OK) | 4 (Running) | Client has responded to the prompt |
| 10 (Stopped) | 12 (Query Pending) | Waiting for the client to respond to the prompt |

346 **7.2.4 CIM_ConcreteJob.DeleteOnCompletion**

347 This profile extends [DSP1103](#) to define that the default value shall be TRUE. This boolean property
 348 indicates whether the CIM_ConcreteJob instance associated to a diagnostic test execution is
 349 automatically deleted when test execution completes. See [DSP1103](#) for further implementation details.

350 **7.2.5 CIM_ConcreteJob.TimeBeforeRemoval**

351 This profile extends [DSP1103](#) to define the time to wait before removing a job after the job completes, is
 352 terminated or killed. The value supplied must be a datetime offset.

353 See [DSP1103](#) for further implementation details.

354 [DSP1103](#) recommends a value of five or more minutes. For diagnostic test environments, clients should
 355 consider using a scaled approach. For example, start with a value ten times the typical elapsed time
 356 experienced in a normally loaded environment and increase to possibly twenty times for a highly loaded
 357 environment.

358 **7.2.6 CIM_ConcreteJob.StartTime**

359 For this profile, the value of this timestamp datetime property represents the start time for the diagnostic
 360 test. Such information should also be written to a CIM_DiagnosticLog associated to the diagnostic test
 361 using a CIM_DiagnosticServiceRecord entry.

362 **7.2.7 CIM_ConcreteJob.ElapsedTime**

363 For this profile, the value of this interval datetime property shall be updated at a vendor-defined interval. A
 364 client can monitor this property at a client-defined interval. When the property changes its value, the client
 365 knows that the test is still making progress.

366 7.2.8 CIM_ConcreteJob.PercentComplete

367 In addition to the requirements specified in [DSP1103](#), this profile uses PercentComplete to show the
368 amount of testing done in terms of actual percent complete. Service implementations should update this
369 property within a reasonable time of becoming aware of a progress change. If progress can not be
370 determined with that reasonable amount of time it should be set to 50 percent. It shall be set to 100
371 percent only when the test is complete. It shall not be set to 100 percent if the test stops for any other
372 reason (for example, the test stopped or was killed by user, the test exited due to a critical failure, or the
373 test found an error and HaltOnError is TRUE) because the actual percent complete is not 100 percent.

374 7.2.9 CIM_ConcreteJob.ElementName

375 The value of this string property shall be initialized to the value of the ElementName property of its
376 associated CIM_DiagnosticTest instance. Clients can modify this value to further identify the test
377 instance.

378 EXPERIMENTAL

379 7.3 CIM_DiagnosticServiceJobCapabilities

380 This subclause indicates the properties of the optional CIM_DiagnosticServiceJobCapabilities class. This
381 class should be implemented for the convenience of clients. However, a client cannot modify any
382 properties in CIM_DiagnosticServiceJobCapabilities.

383 To start a diagnostic test, a client invokes the CIM_DiagnosticTest.RunDiagnosticService() extrinsic
384 method. The Setting input parameter is an instance of CIM_JobSettingData (Client). When implemented,
385 the property values of CIM_JobSettingData (Client) instance shall be consistent with the values of the
386 CIM_DiagnosticServiceJobCapabilities instance.

387 If CIM_DiagnosticServiceJobCapabilities is implemented, a client can create a CIM_JobSettingData
388 (Client) instance containing the default values specified in the CIM_DiagnosticServiceJobCapabilities
389 instance by invoking the CIM_DiagnosticServiceJobCapabilities.CreateJobSettings() extrinsic method.
390 See subclause 8.3.1 for further information.

391 For interactive diagnostic tests that will wait for a client response, the CIM_DiagnosticTest.Characteristics
392 property shall contain the value 3 (Is Interactive). An interactive test may define the time interval it shall
393 wait for a client to respond. While waiting for the client to respond, the CIM_ConcreteJob.JobState
394 property associated to the diagnostic test has the value of 12 (Query Pending). If the client fails to
395 respond within the specified time interval, the diagnostic test may terminate, resume using default
396 responses, or wait another time interval. The CIM_DiagnosticServiceJobCapabilities properties
397 InteractiveTimeoutMax, DefaultValuesSupported, and ClientRetriesMax control such behavior. Thus,
398 when the CIM_DiagnosticTest.Characteristics property contains the value 3 (Is Interactive), these
399 properties shall have a value. Otherwise, they are ignored.

400 7.3.1 CIM_DiagnosticServiceJobCapabilities.InstanceID

401 CIM_DiagnosticServiceJobCapabilities.InstanceID should be constructed using the following preferred
402 algorithm:

403 <OrgID>:<LocalID>

404 where <OrgID> identifies the business entity (for example, ACME) and <LocalID> is a value that uniquely
405 identifies each DiagnosticServiceJobCapabilities instance that is instantiated on a system.

406 The purpose for <LocalID> is to provide some form of uniqueness within the context of different
407 DiagnosticServiceJobCapabilities instances within the system. In practice, since there would be only one
408 CIM_DiagnosticServiceJobCapabilities for an instance of the CIM_DiagnosticTest.

409 **7.3.2 CIM_DiagnosticServiceJobCapabilities.ElementName**

410 The value of this string property shall correspond to the value of the ElementName property of its
411 associated CIM_DiagnosticTest instance.

412 **7.3.3 CIM_DiagnosticServiceJobCapabilities.DeleteJobSupported**

413 This boolean property indicates whether the diagnostic test implementation allows a client to perform a
414 DeleteInstance operation on a CIM_ConcreteJob instance. It also defines whether a client can set the
415 value of CIM_JobSettingData.DeleteOnCompletion when it passes an instance of CIM_JobSettingData
416 as the JobSetting parameter to the CIM_DiagnosticTest.RunDiagnosticService() extrinsic method.

417 If the value of DeleteJobSupported is FALSE, a client cannot perform a DeleteInstance operation on a
418 CIM_ConcreteJob instance associated to its CIM_DiagnosticTest instance. Furthermore, the value of the
419 CIM_JobSettingData.DeleteOnCompletion property shall be TRUE. In addition, a client shall not set the
420 value of the DeleteOnCompletion property in the CIM_JobSettingData instance that it passes as the
421 JobSetting parameter to the CIM_DiagnosticTest.RunDiagnosticService() extrinsic method.

422 If the value of DeleteJobSupported is TRUE, a client can perform a DeleteInstance operation on a
423 CIM_ConcreteJob instance associated to its CIM_DiagnosticTest instance. Furthermore, the value of the
424 CIM_JobSettingData.DeleteOnCompletion property may be TRUE or FALSE. In addition, a client may set
425 the value of the DeleteOnCompletion property in the CIM_JobSettingData instance that it passes as the
426 JobSetting parameter to the CIM_DiagnosticTest.RunDiagnosticService() extrinsic method.

427 However, a client cannot perform the DeleteInstance operation when CIM_ConcreteJob.JobState has the
428 value 2 (New), 3 (Starting), 4 (Running), 5 (Suspended) or 12 (Query Pending) even if
429 DeleteJobSupported is TRUE and CIM_ConcreteJob.DeleteOnCompletion is FALSE.

430 To delete a non-completed job, a client can terminate the job by changing its state to 8 (Terminated) or 9
431 (Killed) by invoking the CIM_ConcreteJob.RequestedStateChange() extrinsic method.

432 **7.3.4 CIM_DiagnosticServiceJobCapabilities.RequestedStatesSupported**

433 This array property indicates the permitted values that a client may pass as the RequestedState
434 parameter to the CIM_ConcreteJob.RequestStateChange() extrinsic method. The permitted values are 2
435 (Start), 3 (Suspend), 4 (Terminate), and 5 (Kill).

436 A client specifies 3 (Suspend) to suspend a diagnostic test and specifies 2 (Start) to resume a suspended
437 diagnostic test.

438 **7.3.5 CIM_DiagnosticServiceJobCapabilities.InteractiveTimeoutMax**

439 This interval datetime offset property shall have a value if the CIM_DiagnosticTest.Characteristics
440 property contains the value 3 (Is Interactive).

441 For an interactive diagnostic test that prompts a client for a response, this property defines the maximum
442 time interval a test shall wait for a client to respond. If a diagnostic test prompts a client multiple times, the
443 specified maximum time interval applies to each prompt.

444 If a client passes an instance of CIM_JobSettingData as the JobSetting parameter to the
445 CIM_DiagnosticTest.RunDiagnosticService() extrinsic method, and the value of
446 CIM_JobSettingData.InteractiveTimeout exceeds the value of
447 CIM_DiagnosticServiceJobCapabilities.InteractiveTimeoutMax, the associated CIM_ConcreteJob
448 instance shall use the value of CIM_DiagnosticServiceJobCapabilities.InteractiveTimeoutMax

449 NOTE If a default for InteractiveTimeout is supported, then this would be specified in the Default
450 CIM_JobSettingData.InteractiveTimeout property.

451 **7.3.6 CIM_DiagnosticsServiceJobCapabilities.DefaultValuesSupported**

452 This boolean property indicates whether an interactive diagnostic test will supply default input values
453 when the test prompts a client for a response but the client fails to respond.. This property shall have a
454 value if the CIM_DiagnosticTest.Characteristics property contains the value of 3 (Is Interactive).
455 Otherwise, it is ignored.

456 When the value is TRUE, the values of CIM_JobSettingData.DefaultInputValues and
457 JobSettingData.DefaultInputNames are used as needed by the CIM_ConcreteJob instance. Otherwise,
458 the value shall be FALSE.

459 **7.3.7 CIM_DiagnosticsServiceJobCapabilities.ClientRetriesMax**

460 This property shall have a value if the CIM_DiagnosticTest.Characteristics property contains the value of
461 3 (Is Interactive). Otherwise, it is ignored.

462 When an interactive diagnostic test prompts a client for a response, the test may define the time interval it
463 shall wait for a client to respond. This property indicates the maximum number of times a diagnostic test
464 shall wait for another time interval.

465 **7.3.8 CIM_DiagnosticsServiceJobCapabilities.CleanupInterval**

466 As described in [DSP1103](#), if the value of the CIM_ConcreteJob.DeleteOnCompletion property is FALSE,
467 the job associated to the diagnostic test execution shall remain until it is explicitly deleted. When the value
468 of the CIM_ConcreteJob.DeleteOnCompletion property is FALSE, the CleanupInterval datetime property
469 defines the time interval before the job is removed.

470 **7.3.9 CIM_DiagnosticServiceJobCapabilities.SilentModeSupported**

471 If the value of the property is TRUE, the interactive diagnostic test is capable of running without prompting
472 the client for responses. Instead, the test uses the default input argument values defined in
473 CIM_JobSettingdData. If the value of the property is FALSE, the interactive diagnostic test shall prompt
474 the client for responses.

475 **EXPERIMENTAL**

476 **7.4 CIM_JobSettingData (Client)**

477 This subclause indicates the properties of the CIM_JobSettingData class that may be used by a client as
478 the JobSetting parameter when invoking the CIM_DiagnosticTest.RunDiagnosticService() intrinsic
479 method to start a diagnostic test. An instance of this class controls the execution of CIM_ConcreteJob
480 instance related to the executing diagnostic test.

481 If CIM_DiagnosticServiceJobCapabilities is implemented, a client can create a CIM_JobSettingData
482 (Client) instance containing the default values specified in the CIM_DiagnosticServiceJobCapabilities
483 instance by invoking the CIM_DiagnosticServiceJobCapabilities.CreateJobSettings() intrinsic method.
484 See subclause 8.3.1 for further information.

485 CIM_JobSettingData is specified by a client as an embedded instance input parameter. The class
486 CIM_JobSettingData (Client) is defined in the CIM Elements tables to define what the client may include
487 in the embedded instance. In addition, the client should refer to the
488 CIM_DiagnosticServiceJobCapabilities class to see what restrictions the implementation may impose on
489 the client providing the CIM_JobSettingData embedded instance.

490 For interactive diagnostic tests that will wait for a client response, the CIM_DiagnosticTest.Characteristics
491 property shall contain the value 3 (Is Interactive). An interactive test may define the time interval it shall
492 wait for a client to respond. While waiting for the client to respond, the CIM_ConcreteJob.JobState

493 property associated to the diagnostic test has the value of 12 (Query Pending). If the client fails to
494 respond within the specified time interval, the diagnostic test may terminate, resume using default
495 responses, or wait another time interval. The CIM_JobSettingData properties InteractiveTimeout,
496 TerminateOnTimeout, DefaultInputValues, and DefaultInputNames control such behavior. Thus, when the
497 CIM_DiagnosticTest.Characteristics property contains the value 3 (Is Interactive) and the value of
498 CIM_DiagnosticServiceJobCapabilities.DefaultValuesSupported is TRUE, these properties shall have a
499 value. Otherwise, they are ignored.

500 **7.4.1 CIM_JobSettingData.InstanceID**

501 CIM_JobSettingData.InstanceID should be constructed using the following preferred algorithm:

502 <OrgID>:<LocalID>

503 where <OrgID> identifies the business entity (for example, ACME) and <LocalID> is a value that uniquely
504 identifies each JobSettingData instance that is instantiated on a system.

505 The purpose for <LocalID> is to provide some form of uniqueness within the context of different
506 JobSettingData instances within the system. In practice, <LocalID> could be an incremented counter or a
507 timestamp in combination with other test identifiers or factors.

508 **7.4.2 CIM_JobSettingData.DeleteOnCompletion**

509 This boolean property indicates whether the job should be automatically deleted upon completion. The
510 property is mandatory. When the value is TRUE, the job shall be deleted after the
511 CIM_ConcreteJob.TimeBeforeRemoval time interval has elapsed. When the value is FALSE, the job must
512 be deleted by an DeleteInstance operation.

513 NOTE When the value of the CIM_DiagnosticServiceJobCapabilities.DeleteJobSupported property is FALSE, the
514 value of CIM_JobSettingData.DeleteOnCompletion shall be TRUE.

515 **EXPERIMENTAL**

516 **7.4.3 CIM_JobSettingData.InteractiveTimeout**

517 This interval datetime property shall have a value if the CIM_DiagnosticTest.Characteristics property
518 contains the value of 3 (Is Interactive). Otherwise, this property is ignored.

519 If the client fails to respond within the specified time interval, the test may terminate, resume using default
520 responses, or wait another time interval. The default value is 15 minutes (00000000001500.000000:000).

521 **7.4.4 CIM_JobSettingData.TerminateOnTimeout**

522 This boolean property shall have a value if the CIM_DiagnosticTest.Characteristics property contains the
523 value of 3 (Is Interactive). Otherwise, this property is ignored.

524 This property defines the behavior when a client fails to respond within the time interval defined by
525 CIM_JobSettingData.InteractiveTimeout. When this value is TRUE the job will terminate when the
526 InteractiveTimeout is exceeded on the last retry, if applicable. When this value is FALSE the job will use
527 DefaultInputValues and DefaultInputNames.

528 **7.4.5 CIM_JobSettingData.DefaultInputValues**

529 This string array property shall contain one or more values if the CIM_DiagnosticTest.Characteristics
530 property contains the value of 3 (Is Interactive) and the value of
531 CIM_DiagnosticServiceJobCapabilities.DefaultValuesSupported is TRUE. Otherwise, this property is
532 ignored.

533 This string array property contains the default values for a client response when the test resumes. The
534 name of each DefaultInputValues array element is defined at the same array index in the
535 CIM_JobSettingData.DefaultInputNames string array.

536 NOTE These values override any values that may be defined in the CIM_JobSettingData (Default) instance.

537 **7.4.6 CIM_JobSettingData.DefaultInputNames**

538 This string array property shall contain one or more values if the CIM_DiagnosticTest.Characteristics
539 property contains the value of 3 (Is Interactive) and the value of
540 CIM_DiagnosticServiceJobCapabilities.DefaultValuesSupported is TRUE. Otherwise, this property is
541 ignored.

542 This string array property contains the possible argument names requested by the diagnostic test. The
543 default value of each DefaultInputNames array element is defined at the same array index in the
544 CIM_JobSettingData.DefaultInputValues string array.

545 **7.4.7 CIM_JobSettingData.ClientRetries**

546 This integer property indicates the number of times the diagnostic test will prompt the client for the same
547 response after the client fails to invoke the CIM_ConcreteJob.ResumeWithInput() or
548 CIM_ConcreteJob.ResumeWithAction() extrinsic method within a specified period of time . A non-zero
549 value for this property indicates that the diagnostic test will issue another DIAG34 or DIAG35 message for
550 the same response.

551 This property is required if CIM_DiagnosticTest.Characteristics has the value of 3 (Is Interactive).
552 Otherwise, this property value is ignored.

553 **7.4.8 CIM_JobSettingData.RunInSilentMode**

554 This boolean property indicates whether the diagnostic test will not prompt the client for responses even
555 though CIM_DiagnosticTest.Characteristics contains the value of 3 (Is Interactive). When the value is
556 TRUE, no prompts are issued. Instead, the diagnostic test will execute using the default values defined in
557 CIM_JobSettingData. When the value is FALSE, the interactive diagnostic test will prompt the client for a
558 response.

559 If CIM_DiagnosticServiceJobCapabilities.SilentModeSupported has the value of FALSE, this property is
560 ignored.

561 **EXPERIMENTAL**

562 **7.5 CIM_JobSettingData (Default)**

563 This subclause indicates the properties of the default CIM_JobSettingData class. An instance of this class
564 controls the execution of a diagnostic test job. This class is optional. If it is implemented it shall represent
565 the CIM_JobSettingData (Default) for the CIM_DiagnosticTest. This instance is identified by
566 CIM_ElementSettingData with IsDefault="true" between the instance and the CIM_DiagnosticTest
567 instance. A different default CIM_JobSettingData instance may be defined for each CIM_DiagnosticTest
568 test type. For example, each of the different CPU diagnostic tests may define a different set of default
569 CIM_JobSettingData values.

570 A CIM_JobSettingData (Client) may be specified by a client as an embedded instance input to an
571 invocation of the CIM_DiagnosticTest.RunDiagnosticService() method. This embedded instance is not
572 instantiated as an instance of CIM_JobSettingData (Default), but the class CIM_JobSettingData (Client) is
573 defined in the CIM Elements tables to indicate what the client may include in the embedded instance. To
574 use all of the default values, the client can create an identical instance of CIM_JobSettingData except that
575 the InstanceID key property shall have a different value.

576 If CIM_DiagnosticServiceJobCapabilities is implemented, the client should refer to the
577 CIM_DiagnosticServiceJobCapabilities instance to see what restrictions the implementation may impose
578 on the client providing the CIM_JobSettingData embedded instance.

579 If CIM_DiagnosticServiceJobCapabilities is implemented, a client can create a CIM_JobSettingData
580 (Client) instance containing the default values specified in the CIM_DiagnosticServiceJobCapabilities
581 instance by invoking the CIM_DiagnosticServiceJobCapabilities.CreateJobSettings() extrinsic method.
582 See subclause 8.3.1 for further information.

583 **7.5.1 CIM_JobSettingData.InstanceID**

584 CIM_JobSettingData.InstanceID should be constructed using the following preferred algorithm:

585 <OrgID>:<LocalID>

586 where <OrgID> identifies the business entity (for example, ACME) and <LocalID> is a value that uniquely
587 identifies each JobSettingData instance that is instantiated on a system.

588 The purpose for <LocalID> is to provide some form of uniqueness within the context of different
589 JobSettingData instances within the system. In practice, <LocalID> could be an incremented counter or a
590 timestamp in combination with other test identifiers or factors.

591 **7.5.2 CIM_JobSettingData.DeleteOnCompletion**

592 This boolean property indicates whether the job should be automatically deleted upon completion. The
593 property is mandatory. When the value is TRUE, the job shall be deleted after the
594 CIM_ConcreteJob.TimeBeforeRemoval time interval. When the value is FALSE, the job shall be deleted
595 by an DeleteInstance operation.

596 NOTE When the value of the CIM_DiagnosticServiceJobCapabilities.DeleteJobSupported property is FALSE, the
597 value of CIM_JobSettingData.DeleteOnCompletion shall be TRUE.

598 **EXPERIMENTAL**

599 **7.5.3 CIM_JobSettingData.InteractiveTimeout**

600 This interval datetime property shall have a value if the CIM_DiagnosticTest.Characteristics property
601 contains the value of 3 (Is Interactive). Otherwise, this property is ignored.

602 When an interactive diagnostic test prompts a client for a response, the test may define the time interval it
603 shall wait for a client to respond. While waiting for the client to respond, the CIM_ConcreteJob.JobState
604 property associated to the diagnostic test has the value of 12 (Query Pending). This property indicates
605 the time interval that an interactive diagnostic test will wait for a client to respond.

606 If the client fails to respond within the specified time interval, the test may terminate, resume using default
607 responses, or wait another time interval. The default value is 15 minutes (00000000001500.000000:000).

608 **7.5.4 CIM_JobSettingData.TerminateOnTimeout**

609 This boolean property shall have a value if the CIM_DiagnosticTest.Characteristics property contains the
610 value of 3 (Is Interactive). Otherwise, this property is ignored.

611 This property defines the behavior when a client fails to respond within the time interval defined by
612 CIM_JobSettingData.InteractiveTimeout. If this value is TRUE the job will terminate when the
613 InteractiveTimeout is exceeded on the last retry, if applicable. If FALSE the job will use DefaultInputs.

614 **7.5.5 CIM_JobSettingData.DefaultInputValues**

615 This string array property shall contain one or more values if the CIM_DiagnosticTest.Characteristics
616 property contains the value of 3 (Is Interactive) and the value of
617 CIM_DiagnosticServiceJobCapabilities.DefaultValuesSupported is TRUE. Otherwise, this property is
618 ignored.

619 If the client fails to respond within the specified time interval, the diagnostic test may terminate, resume
620 using default responses, or wait another time interval.

621 This string array property contains the default values for client responses when the test resumes. The
622 name of each DefaultInputValues array element is defined at the same array index in the
623 CIM_JobSettingData.DefaultInputNames string array.

624 **7.5.6 CIM_JobSettingData.DefaultInputNames**

625 This string array property shall contain one or more values if the CIM_DiagnosticTest.Characteristics
626 property contains the value of 3 (Is Interactive) and the value of
627 CIM_DiagnosticServiceJobCapabilities.DefaultValuesSupported is TRUE. Otherwise, this property is
628 ignored.

629 This string array property contains the possible argument names requested by the diagnostic test. The
630 default value of each DefaultInputNames array element is defined at the same array index in the
631 CIM_JobSettingData.DefaultInputValues string array.

632 **7.5.7 CIM_JobSettingData.ClientRetries**

633 This integer property indicates the number of times the diagnostic test will prompt the client for the same
634 response after the client fails to invoke the CIM_ConcreteJob.ResumeWithInput() or
635 CIM_ConcreteJob.ResumeWithAction() extrinsic method within a specified period of time . A value of
636 zero for this property indicates that the diagnostic test will issue a second prompt for the same response.

637 This property is required if CIM_DiagnosticTest.Characteristics has the value of 3 (Is Interactive).
638 Otherwise, this property value is ignored.

639 **7.5.8 CIM_JobSettingData.RunInSilentMode**

640 This boolean property indicates whether the diagnostic test will not prompt the client for responses even
641 though CIM_DiagnosticTest.Characteristics contains the value of 3 (Is Interactive). When the value is
642 TRUE, no prompts are issued. Instead, the diagnostic test will execute using the default values defined in
643 CIM_JobSettingData. When the value is FALSE, the interactive diagnostic test will prompt the client for a
644 response.

645 If CIM_DiagnosticServiceJobCapabilities.SilentModeSupported has the value of FALSE, this property is
646 ignored.

647 **EXPERIMENTAL**

648 **7.6 Interactive options**

649 An interactive diagnostic test is controlled by properties in the optional
650 CIM_DiagnosticServiceJobCapabilities class, the properties in the mandatory CIM_JobSettingData
651 (Default) class, and the JobSetting input parameter used when the client invokes the
652 CIM_DiagnosticTest.RunDiagnosticService() extrinsic method.

653 When a diagnostic test prompts the client for a response, the time interval a test shall wait for a client to
654 respond is determined by the values in the following properties. Table 3 shows the behavior when the
655 following properties and JobSetting parameter have certain and possibly conflicting values.

- 656 • CIM_DiagnosticServiceJobCapabilities.InteractiveTimeoutMax
- 657 • CIM_DiagnosticServiceJobCapabilities.ClientRetriesMax
- 658 • CIM_JobSettingData.InteractiveTimeout
- 659 • CIM_JobSettingData.TerminateOnTimeout

660 When a timeout occurs or when running in silent mode, the diagnostic test may resume using default
661 values for the arguments that the client should have provided. How default values can be used is
662 determined by the combination of values in the following properties: Table 3 shows the behavior when the
663 following properties and JobSetting parameter have certain and possibly conflicting values.

- 664 • CIM_DiagnosticServiceJobCapabilities.DefaultValuesSupported
- 665 • CIM_JobSettingData.DefaultInputValues
- 666 • CIM_JobSettingData.DefaultInputNames

667 An interactive diagnostic test can be configured to run using default values shown above without
668 prompting the client for any responses using the values in the following properties. Table 3 shows the
669 behavior when the following properties have certain values.

- 670 • CIM_DiagnosticServiceJobCapabilities.SilentModeSupported
- 671 • CIM_JobSettingData.RunInSilentMode

672

Table 3 – Interactive options

| Capability | JobSettingData (Default) | JobSetting | Behavior |
|---|---|---|--|
| InteractiveTimeoutMax | InteractiveTimeout <= InteractiveTimeoutMax | InteractiveTimeout <= InteractiveTimeoutMax | Use JobSetting value |
| | | InteractiveTimeout >= InteractiveTimeoutMax | Use Capability value (2) |
| DefaultValuesSupported = TRUE | DefaultInputValues = non-NULL DefaultInputNames = non-NULL | DefaultInputValues = non-NULL DefaultInputNames = non-NULL | Use JobSetting |
| | | DefaultInputValues = NULL DefaultInputNames = NULL | Use JobSettingData.(2) |
| DefaultValuesSupported = FALSE | DefaultInputValues = NULL DefaultInputNames = NULL | DefaultInputValues = NULL DefaultInputNames = NULL | The client shall supply input values. |
| | | DefaultInputValues = non-NULL DefaultInputNames = non-NULL | JobSetting is ignored. (3) |
| DefaultValuesSupported = TRUE SilentModeSupported = TRUE | DefaultInputValues = non-NULL DefaultInputNames = non-NULL | DefaultInputValues = non-NULL DefaultInputNames = non-NULL RunInSilentMode = TRUE | Run in silent mode using JobSetting default values. |
| | | DefaultInputValues = non-NULL DefaultInputNames = non-NULL RunInSilentMode = TRUE | Run in silent mode using JobSettingData default values. |
| ClientRetriesMax = N DefaultValuesSupported = FALSE | TerminateOnTimeout = TRUE | TerminateOnTimeout = TRUE | The Job will terminate after N tries to solicit input. |
| | | TerminateOnTimeout = FALSE | Use JobSettingData (2) |
| ClientRetriesMax = N | ClientRetries = M where M <= N | ClientRetries = R where R <= N | After R retries, the job terminates or runs with defaults. (1) |
| | | ClientRetries = NULL | After M retries, the job terminates or runs with defaults. (1) (2) |

| Capability | JobSettingData (Default) | JobSetting | Behavior |
|--|--|---|--|
| ClientRetriesMax = N | ClientRetries = M where M <= N | ClientRetries = R where R > N | After M retries, the job terminates or runs with defaults. (1) (2) |
| InteractiveTimeoutMax | InteractiveTimeout > InteractiveTimeoutMax | | Undefined |
| DefaultValuesSupported = TRUE | DefaultInputValues = NULL DefaultInputNames = NULL | | Undefined |
| DefaultValuesSupported = FALSE | DefaultInputValues = non-NULL DefaultInputNames = non-NULL | | Undefined |
| DefaultValuesSupported = TRUE SilentModeSupported = TRUE | DefaultInputValues = NULL DefaultInputNames = NULL RunInSilentMode = TRUE or FALSE | DefaultInputValues = NULL DefaultInputNames = NULL RunInSilentMode = TRUE | Undefined |
| DefaultValuesSupported = FALSE SilentModeSupported = TRUE | | | Undefined |
| DefaultValuesSupported = FALSE ClientRetries = N | TerminateOnTimeout = FALSE | | Undefined |
| ClientRetries = N | ClientRetries = M where M > N | | Undefined |

673 (1) The job will wait one InteractiveTimeout for each response. After the timeout, another prompt is
674 issued.

675 (2) An alert indication (DIAG39) is sent indicating that the JobSetting was reset. The overridden
676 effective JobSetting should be logged.

677 (3) An alert indication (DIAG40) is sent indicating that the default values were not used. The
678 overridden effective JobSetting should be logged.

679 **7.7 Job deletion options**

680 To start a diagnostic test, the client invokes the CIM_DiagnosticTest.RunDiagnosticService extrinsic
681 method, which returns an instance of CIM_ConcreteJob. When a diagnostic test completes execution, the
682 CIM_ConcreteJob instance will be deleted. When and how the CIM_ConcreteJob instance is deleted is
683 controlled by properties in the optional CIM_DiagnosticServiceJobCapabilities class (the first column), the
684 properties in the mandatory CIM_JobSettingData (Default) class (the second column), and the optional

685 JobSetting input parameter (the third column), which is an embedded CIM_JobSettingData instance,
686 used when the client invokes the CIM_DiagnosticTest.RunDiagnosticService() extrinsic method. Table 4
687 shows the behavior (the fourth column) when these following properties have certain and possibly
688 conflicting values.

- 689 • CIM_DiagnosticServiceJobCapabilities.DeleteJobSupported
- 690 • CIM_DiagnosticServiceJobCapabilities.CleanupInterval
- 691 • CIM_JobSettingData.DeleteOnCompletion
- 692 • CIM_JobSettingData.TimeBeforeRemoval

693 Since the JobSetting input parameter is optional, its value may be NULL. In this case, the
694 CIM_DiagnosticTest.RunDiagnosticService extrinsic method shall use the values of the mandatory
695 CIM_JobSettingData (Default) instance.

696 An instance of CIM_DiagnosticServiceJobCapabilities may not exist because its implementation is
697 optional. In this case, ignore the first column when interpreting Table 4.

698

Table 4 – Job deletion options

| Capability | JobSettingData (Default) | JobSetting | Behavior |
|---|------------------------------------|----------------------------|--|
| DeleteJobSupported = TRUE AND CleanupInterval = non-NULL | DeleteOnCompletion = TRUE or FALSE | DeleteOnCompletion = TRUE | The provider deletes the instance TimeBeforeRemoval after the job completes. |
| | | DeleteOnCompletion = FALSE | The client should delete the instance. The provider may delete the instance CleanupInterval after the job completes. |
| DeleteJobSupported = TRUE AND CleanupInterval = NULL | DeleteOnCompletion = TRUE | DeleteOnCompletion = TRUE | The provider deletes the instance TimeBeforeRemoval after the job completes. |
| | | DeleteOnCompletion = FALSE | The client should delete the instance. CleanupInterval is ignored. |
| DeleteJobSupported = FALSE AND CleanupInterval = non-NULL | DeleteOnCompletion = TRUE | DeleteOnCompletion = TRUE | The provider deletes the instance TimeBeforeRemoval after the job completes |
| | | DeleteOnCompletion = FALSE | JobSetting.DeleteOnCompletion is reset to TRUE. The provider deletes the instance TimeBeforeRemoval after the job completes. (1) |
| DeleteJobSupported = FALSE AND CleanupInterval = NULL | DeleteOnCompletion = FALSE | | Undefined |
| DeleteJobSupported = FALSE AND CleanupInterval = non-NULL | DeleteOnCompletion = FALSE | | Undefined |

699 (1) An alert indication (DIAG39) is sent indicating that the JobSetting was reset. The overridden
 700 effective JobSetting should be logged.

701 **8 Methods**

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

704 **8.1 Profile conventions for operations**

705 The default list of operations shall be as mandated in [DSP1103](#), subclauses 8.1.

706 Support for operations for each profile class (including associations) shall be as mandated in [DSP1103](#),
 707 subclauses 8.3 through 8.7.

708 8.2 CIM_ConcreteJob

709 All operations are supported as for CIM_ConcreteJob in [DSP1103](#), subclause 8.2. Additionally, the
710 DeleteInstance operation shall be supported when the CIM_JobSettingData.DeleteOnCompletion
711 property has the value of FALSE.

712 8.2.1 CIM_ConcreteJob.RequestStateChange()

713 All CIM_DiagnosticService.RunDiagnosticService() calls will return a reference to a CIM_ConcreteJob
714 instance, which represents the diagnostic execution. The CIM_ConcreteJob.RequestStateChange()
715 method is invoked to control the diagnostic program execution. The RequestedState input parameter
716 specifies the new desired state (Start, Suspend, Kill, Terminate).

717 Otherwise, the behavior of this extrinsic method shall be as mandated in [DSP1103](#), subclause 8.2.

718 Before invoking this method, a client examines
719 CIM_DiagnosticServiceJobCapabilities.RequestedStatesSupported to determine the values to use for the
720 RequestedState input parameter. The RequestStateChange() extrinsic method shall change the JobState
721 value if the transition is successfully performed.

722 EXPERIMENTAL

723 8.2.2 CIM_ConcreteJob.ResumeWithInput()

724 The CIM_ConcreteJob.ResumeWithInput() extrinsic method is invoked to resume the diagnostic program
725 execution when it has a JobState of 12 (Query Pending).

726 The return values are specified in Table 5. The input parameter is specified in Table 6. No output
727 parameters are defined. No standard messages are defined.

728 For an interactive test, the CIM_ConcreteJob provider prompts the client to respond with DIAG34
729 message, which is a comma separated string of argument names. The client calls
730 CIM_ConcreteJob.ResumeWithInput() to respond with values in the Inputs string array. The first value in
731 the Inputs string array corresponds to the first argument in the DIAG34 message, and so on

732 The Inputs string array shall have a value for each requested argument. A NULL value shall not be used.
733 To use a default value for a requested argument, the client looks for a matching name in
734 CIM_JobSettingData.DefaultInputNames. If found, the client uses the corresponding value from
735 CIM_JobSettingData.DefaultInputValues; that is, at the same array index.

736 If the client invokes the CIM_ConcreteJob.ResumeWithInput() extrinsic method where the Inputs string
737 array argument has invalid values or not enough values, then the CIM_ConcreteJob provider returns 5
738 (Invalid Parameter) and sets CIM_ConcreteJob.JobState = 10 (Exception) provided no more retries
739 remain.

740 If CIM_ConcreteJob.ResumeWithInput() returns value=3 (Can NOT complete within Timeout Period), the
741 client can retry provided more retries remain.

742

Table 5 – ResumeWithInput() method: Return code values

| Value | Description |
|--------------|---|
| 0 | Completed with No Error |
| | The ResumeWithInput was accepted and the job has resumed. The JobState has changed from “12” (Query Pending) to “4” (Running) |
| 2 | Unknown/Unspecified Error |
| | The JobState was “12” (Query Pending) and the inputs were valid, but the request failed for other reasons. |
| 3 | Can NOT complete within Timeout Period |
| 4 | Failed |
| 5 | Invalid Parameter |
| 6 | JobState not Query Pending |
| 32768..65535 | Vendor specific |

743

Table 6 – ResumeWithInput() method: Parameters

| Qualifiers | Name | Type | Description/Values |
|------------|--------|----------|--|
| IN | Inputs | String[] | The client inputs being requested by the job when its state changed to 12 (Query Pending). |

744 **8.2.3 CIM_ConcreteJob.ResumeWithAction()**

745 The CIM_ConcreteJob.ResumeWithAction() extrinsic method is invoked to resume the diagnostic
 746 program execution when it has a JobState of 12 (Query Pending).

747 The return values are specified in Table 7. No input or output parameters are defined. No standard
 748 messages are defined.

749 For an interactive test, the CIM_ConcreteJob provider prompts the client to respond with DIAG35
 750 message. The client invokes the CIM_ConcreteJob.ResumeWithAction() extrinsic method when no
 751 arguments are requested. For example, the diagnostic test might prompt the user to attach the network
 752 cable before allowing the test to proceed.

753

Table 7 – ResumeWithAction() method: Return code values

| Value | Description |
|--------------|--|
| 0 | Completed with No Error |
| | The ResumeWithInput was accepted and the job has resumed. The JobState has changed from “12” (Query Pending) to “4” (Running). |
| 2 | Unknown/Unspecified Error |
| | The JobState was “12” (Query Pending) and the inputs were valid, but the request failed for other reasons. |
| 3 | Can NOT complete within Timeout Period |
| 4 | Failed |
| 6 | JobState not Query Pending |
| 32768..65535 | Vendor specific |

754 8.3 CIM_DiagnosticServiceJobCapabilities

755 8.3.1 CreateGoalSettings()

756 The CIM_DiagnosticServiceJobCapabilities.CreateGoalSettings() method, which is inherited from
757 CIM_Capabilities, is invoked in the context of a specific CIM_DiagnosticServiceJobCapabilities instance.

758 This method is used to create a CIM_JobSettingData instance using the
759 CIM_DiagnosticServiceJobCapabilities as a template. The purpose of this method is to create a
760 CIM_JobSettingData based on the CIM_DiagnosticServiceJobCapabilities on which this method is
761 invoked and has properties set in line with those CIM_DiagnosticServiceJobCapabilities.

762 The return values are specified in Table 8. The parameters are specified in Table 9. No standard
763 messages are defined.

764

Table 8 – CreateGoalSettings() method: Return code values

| Value | Description |
|--------------|-------------------------|
| 0 | Completed with No Error |
| 1 | Not supported |
| 2 | Unknown |
| 3 | Timeout |
| 4 | Failed |
| 5 | Invalid Parameter |
| 6 | Alternative Proposed |
| 32768..65535 | Vendor specific |

765

Table 9 – CreateGoalSettings() method: Parameters

| Qualifiers | Name | Type | Description/Values |
|------------|--------------------------|--------|--|
| IN | TemplateGoalSettings[] | String | An array of CIM_JobSettingData embedded instances that reflect what the client wants. This parameter may be NULL. If NULL, the method returns a setting that conforms to the CIM_DiagnosticServiceJobCapabilities. |
| IN/OUT | SupportedGoalSettings[] | String | An array of CIM_JobSettingData embedded instances that are consistent with the CIM_DiagnosticServiceJobCapabilities and are closest matches to the input TemplateGoalSettings |

766 **EXPERIMENTAL**

767 **8.4 CIM_MethodResult**

768 All operations are supported as for CIM_MethodResult in [DSP1103](#).

769 **8.5 CIM_OwningJobElement**

770 All operations are supported as for CIM_OwningJobElement in [DSP1103](#).

771 **8.6 CIM_AffectedJobElement**

772 All operations are supported as for CIM_AffectedJobElement in [DSP1103](#).

773 **8.7 CIM_AssociatedJobMethodResult**

774 All operations are supported as for CIM_AssociatedJobMethodResult in [DSP1103](#).

775 **8.8 CIM_HostedDependency**

776 All operations are supported as for CIM_HostedDependency in [DSP1103](#).

777 **8.9 CIM_RegisteredProfile**

778 All operations are supported as for CIM_RegisteredProfile in the *Profile Registration Profile* ([DSP1033](#)).

779 **8.10 CIM_JobSettingData**

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

782

Table 10 – Operations: CIM_JobSettingData

| Operation | Requirement | Messages |
|------------------------|-------------|----------|
| GetInstance | Mandatory | None |
| EnumerateInstances | Mandatory | None |
| EnumerateInstanceNames | Mandatory | None |
| ExecQuery | Optional | None |
| Associators | Mandatory | None |
| AssociatorNames | Mandatory | None |
| References | Optional | None |
| ReferenceNames | Optional | None |

783 8.11 CIM_ElementSettingdData

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

786

Table 11 – Operations: CIM_ElementSettingData

| Operation | Requirement | Messages |
|------------------------|-------------|----------|
| GetInstance | Mandatory | None |
| EnumerateInstances | Mandatory | None |
| EnumerateInstanceNames | Mandatory | None |

787 8.12 CIM_ElementCapabilities

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

790

Table 12 – Operations: CIM_ElementCapabilities

| Operation | Requirement | Messages |
|------------------------|-------------|----------|
| GetInstance | Mandatory | None |
| EnumerateInstances | Mandatory | None |
| EnumerateInstanceNames | Mandatory | None |

791 8.13 CIM_DiagnosticTest.RunDiagnosticService()

792 [DSP1002](#) describes this extrinsic method. This subclause describes how the
793 CIM_DiagnosticServiceJobCapabilites, the CIM_JobSettingData (Default), and the JobSetting input
794 parameter affects the execution of this extrinsic method.

795 The CIM_JobSettingData (Default) is mandatory. The CIM_DiagnosticServiceJobCapabilities and the
796 JobSetting parameter of the RunDiagnosticService method are optional. If the
797 CIM_DiagnosticServiceJobCapabilities is not implemented, the client application cannot alter the default
798 CIM_JobSettingData for the diagnostic test and the JobSetting parameter should be NULL or set to the
799 default CIM_JobSettingData. If the client application sets the JobSetting parameter to values that conflict
800 with the default CIM_JobSettingData, the test will not fail, but the JobSetting parameter will be reset to the

801 default values (the “effective” JobSetting) and a warning alert message will be issued. The effective
 802 JobSetting parameter values will also be logged in the associated CIM_DiagnosticLog.

803 If CIM_DiagnosticServiceJobCapabilities is implemented, the client application may specify values in the
 804 JobSetting parameter that conform to the corresponding capability. For example, the client application
 805 may specify an InteractiveTimeout that is equal or less than the InteractiveTimeoutMax..If the client
 806 application specifies a value that is in conflict with the options allowed by the
 807 CIM_DiagnosticServiceJobCapabilities for the diagnostic test, the conflicting value will be reset to one of
 808 two values: The value in the default JobSettingData or the maximum allowed by the
 809 CIM_DiagnosticServiceJobCapabilities. If the client invokes this extrinsic method and the JobSetting
 810 parameter has the value of NULL, the default CIM_JobSettingData will be used. In either case, if any
 811 value was changed, an alert message will be issued. Whether or not a value was changed, the effective
 812 Jobsetting used by the diagnostic test execution will be logged in the CIM_DiagnosticLog.

813 Table 13 shows the behavior for different combinations of CIM_DiagnosticServiceJobCapabilities,
 814 CIM_JobSettingData (Default), and the JobSetting parameter.

815 **Table 13 – Job setting options**

| Capabilities | JobSettingData (Default) | JobSetting | Behavior |
|--------------|--------------------------|-----------------------------|--|
| Absent | Present | NULL | Use JobSettingData (Default). |
| Absent | Present | No conflict in values exist | Use JobSetting. |
| | | Conflict in values exist | Use JobSettingData (Default). An alert is issued. |
| Present | Present | NULL | Use JobSettingData (Default). |
| Present | Present | No conflict in values exist | Use JobSetting. |
| | | Conflict in values exist | JobSetting is modified to conform to the capabilities. An alert is issued. |
| | Absent | | Undefined. |
| Present | Present but conflicting | | Undefined. |

816 The effective JobSetting used is logged.

817 **EXPERIMENTAL**

818 **9 Use cases (Informative)**

819 This clause contains use cases for the *Diagnostic Job Control Profile* that describes how a diagnostic test
 820 behaves and interacts with a client. An interactive diagnostic test is a CIM_DiagnosticTest instance where
 821 its Characteristics property contains the value 3 (Is Interactive).

822 **9.1 Use case summary**

823 Table 14 summarizes the use cases that are described in this clause. The use cases are categorized and
 824 named, and references are provided to the subclause that describes the use case.

825 **NOTE** Although use case names follow the convention for naming classes, properties, and methods in the schema,
 826 this naming was done for readability only and does not imply any functionality attached to the name.

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

Table 14 – Diagnostic test use cases

| Category | Scenarios | Description |
|------------------------------|--|--|
| User input required | The test requires a single response for a single value. The client responds with valid values. See 9.2.1. | Some interactive diagnostic tests require the user to respond with input values before the test can proceed. See 9.2. |
| | The test requires a single response for multiple values. The client responds with valid values. See 9.2.2. | |
| | The test requires multiple responses. After the client responds with valid values, the test runs to partial completion and then prompts for another response. The client responds to each prompt with valid values. See 9.2.3. | |
| | The client fails to respond to a prompt. See 9.2.4. | |
| | The client responds to a prompt with invalid values. See 9.2.5. | |
| | The client responds to a prompt with an insufficient number of values. See 9.2.6. | |
| User action required | The test requires a single response. The client responds. See 9.3.1. | Some interactive diagnostic tests require the user to perform an action before the test can proceed. See 9.3. |
| | The test requires multiple responses required before running the test. The client responds to each prompt. See 9.3.2. | |
| | The test requires multiple responses. After the client responds, the test runs to partial completion and then prompts for another response. The client responds to each prompt. See 9.3.3. | |
| | The client fails to respond to a prompt. See 9.3.4. | |
| Silent mode operation | | This profile defines the ability to run interactive diagnostic tests without user interaction by using predefined default values. See 9.4. |
| Finding a diagnostic job | Find all diagnostic jobs on a system | This profile defines the sequence of operations to perform this task. See 9.5.1. |
| | Find all diagnostic jobs for a ManagedElement | This profile defines the sequence of operations to perform this task. See 9.5.2. |
| Configuring a diagnostic job | Get default job settings | This profile defines the sequence of operations to perform this task. See 9.6.1. |

| Category | Scenarios | Description |
|-------------------------------------|---------------------------------------|--|
| | Create job settings | This profile defines the sequence of operations to perform this task. See 9.6.2. |
| Control a job for a diagnostic test | Suspend a job for a diagnostic test | See 9.7.1. |
| | Resume a job for a diagnostic test | See 9.7.2. |
| | Terminate a job for a diagnostic test | See 9.7.3. |
| | Kill a job for a diagnostic test | Abort a running diagnostic immediately, with no attempt to perform a clean termination. See 9.7.4. |
| Delete a job | Client deletes a job | See 9.8.1. |
| | Provider deletes a job | See 9.8.2. |

829 Before performing the use cases in this profile, it is assumed that a client has already utilized the use
 830 case methodology defined in [DSP1002](#) to discover the DiagnosticTest instance.

831 To start a test, the client invokes the DiagnosticTest.RunDiagnosticService() extrinsic method which
 832 returns 0 (Success) and the objectpath of a ConcreteJob instance with ConcreteJob.JobState = 4
 833 (Running) and CIM_ConcreteJob.PercentComplete = 0. Thereafter, ConcreteJob manages the diagnostic
 834 test execution. Additionally, a client monitors and interacts with the diagnostic test via the returned
 835 ConcreteJob instance.

836 NOTE An interactive diagnostic test may prompt a client more than once during test execution where some prompts
 837 require user input while others do not.

838 In the following examples, responses are enclosed in brackets. [Enter] indicates that the client pressed
 839 the Enter key, typically to select the default. <timeout> indicates that the client did not respond.

840 **9.2 User input required**

841 For an interactive test where user input is required, the ConcreteJob provider prompts the client to
 842 respond with DIAG34 message, which is a comma separated string of argument names. The client calls
 843 ConcreteJob.ResumeWithInput() to respond with values in the Inputs string array. The first value in the
 844 Inputs string array corresponds to the first argument in the DIAG34 message, and so on.

845 **9.2.1 Single prompt and response has a valid value**

846 - *How many minutes do you want the test to run? [20] (Test starts)*

- 847 1) Client calls DiagnosticTest.RunDiagnosticService() which returns 0 (Success) and the objectpath
 848 of the ConcreteJob instance and starts the diagnostic test. ConcreteJob.JobState = 4 (Running)
 849 and ConcreteJob.PercentComplete = 0.
- 850 2) ConcreteJob provider prompts client to respond with a DIAG34 message, which requests a value
 851 for the Minutes argument.
- 852 3) ConcreteJob provider sets ConcreteJob.JobState = 12 (Query Pending) to wait for client
 853 response.
- 854 4) Client calls ConcreteJob.ResumeWithInput() to respond with input argument Inputs[0]="20" for
 855 the Minutes argument requested by DIAG34 message.
- 856 5) ConcreteJob provider sets ConcreteJob.JobState = 4 (Running).
- 857 6) ConcreteJob.ResumeWithAction() returns 0 ("Completed with No Error") and test execution
 858 starts.

859 7) After the test completes successfully, the ConcreteJob provider sets ConcreteJob.JobState = 7
860 (Completed), ConcreteJob.OperationalStatus= 17 (Completed) and ConcreteJob
861 PercentComplete=100.

862 **9.2.2 Single prompt and response has multiple valid values**

863 - *Which CPU speeds in GHz to you want to test? [2.4,3.0] (Test starts)*

864 1) Client calls DiagnosticTest.RunDiagnosticService() which returns 0 (Success) and the objectpath
865 of the ConcreteJob instance and starts the diagnostic test. ConcreteJob.JobState = 4 (Running)
866 and ConcreteJob.PercentComplete = 0.

867 2) ConcreteJob provider prompts client to respond with a DIAG34 message, which requests a value
868 for the CPUSpeed1 and CPUSpeed2 arguments.

869 3) ConcreteJob provider sets ConcreteJob.JobState = 12 (Query Pending) to wait for client
870 response.

871 4) Client calls ConcreteJob.ResumeWithInput() to respond with input argument Inputs[0]="2.4" for
872 the CPUSpeed1 argument and Inputs[1]="3.0" for the CPUSpeed2 argument as requested by the
873 DIAG34 message.

874 5) ConcreteJob provider sets ConcreteJob.JobState = 4 (Running).

875 6) ConcreteJob.ResumeWithAction() returns 0 ("Completed with No Error") and test execution
876 starts.

877 7) After the test completes successfully, the ConcreteJob provider sets ConcreteJob.JobState = 7
878 (Completed), ConcreteJob.OperationalStatus= 17 (Completed) and ConcreteJob
879 PercentComplete=100.

880 **9.2.3 Multiple prompts and responses required with partial test execution after each**

881 - *Which network port do you want to test? [2] (Test execution starts)*

882 - *Which network port do you want to test next? [3] (Test execution resumes)*

883 1) Client calls DiagnosticTest.RunDiagnosticService() which returns 0 (Success) and the objectpath
884 of the ConcreteJob instance and starts the diagnostic test. ConcreteJob.JobState = 4 (Running)
885 and ConcreteJob.PercentComplete = 0.

886 2) ConcreteJob provider prompts client to respond with a DIAG34 message, which requests a value
887 for the Port1 argument.

888 3) ConcreteJob provider sets ConcreteJob.JobState = 12 (Query Pending) to wait for client
889 response.

890 4) Client calls ConcreteJob.ResumeWithInput() to respond with input argument Inputs[0]="2" for the
891 Port1 argument requested by DIAG34 message.

892 5) ConcreteJob provider sets ConcreteJob.JobState = 4 (Running) and
893 ConcreteJob.PercentComplete = 25 and test execution starts.

894 6) ConcreteJob.ResumeWithAction() returns 0 ("Completed with No Error") and test execution
895 resumes.

896 7) Test execution finishes with port 2. ConcreteJob provider sets ConcreteJob.PercentComplete =
897 50 and execution continues.

898 8) ConcreteJob provider prompts client to respond with a DIAG34 message, which requests a value
899 for the Port2 argument.

900 9) ConcreteJob provider sets ConcreteJob.JobState = 12 (Query Pending) to wait for client
901 response.

- 902 10) Client calls ConcreteJob.ResumeWithInput() to respond with input argument Inputs[0]="3" for the
903 Port2 argument requested by DIAG34 message.
- 904 11) ConcreteJob provider sets ConcreteJob.JobState = 4 (Running) and
905 ConcreteJob.PercentComplete = 50 and test execution starts.
- 906 12) After the test completes successfully, the ConcreteJob provider sets ConcreteJob.JobState = 7
907 (Completed) , ConcreteJob.OperationalStatus= 17 (Completed) and ConcreteJob
908 PercentComplete=100.

909 **9.2.4 Client does not respond to a prompt**

910 - Which network port do you want to test? <timeout>

- 911 1) Client calls DiagnosticTest.RunDiagnosticService() which returns 0 (Success) and the objectpath
912 of the ConcreteJob instance and starts the diagnostic test. ConcreteJob.JobState = 4 (Running)
913 and ConcreteJob.PercentComplete = 0.
- 914 2) ConcreteJob provider prompts client to respond with a DIAG34 message, which requests a value
915 for the Port argument.
- 916 3) ConcreteJob provider sets ConcreteJob.JobState = 12 (Query Pending) to wait for client
917 response.
- 918 4) When JobSettingData.InteractiveTimeout is exceeded, the ConcreteJob provider examines the
919 value of JobSettingData.ClientRetriesMax.
- 920 5) If the value of JobSettingData.ClientRetriesMax is zero, or if the number of retries has been
921 exceeded, the ConcreteJob provider sets ConcreteJob.JobState = 10 (Exception) provided
922 JobSettingData.TerminateOnTimeout=TRUE.
- 923 6) If the value of JobSettingData.ClientRetriesMax is non-zero, steps 2, 3, 4, and 5 are repeated.

924 NOTE If the value of JobSettingData.ClientRetriesMax is non-zero, the ConcreteJob provider monitors how many
925 timeouts that have occurred.

926 **9.2.5 Client responds with an invalid value**

927 - Which network port do you want to test? [-1]

- 928 1) Client calls DiagnosticTest.RunDiagnosticService() which returns 0 (Success) and the objectpath
929 of the ConcreteJob instance and starts the diagnostic test. ConcreteJob.JobState = 4 (Running)
930 and ConcreteJob.PercentComplete = 0.
- 931 2) ConcreteJob provider prompts client to respond with a DIAG34 message, which requests a value
932 for the Port argument.
- 933 3) Client calls ConcreteJob.ResumeWithInput() to respond with input argument Inputs[0]="-1" for
934 the Port1 argument requested by DIAG34 message.
- 935 4) The ConcretJob provider detects that Inputs[0]="-1" is invalid.
- 936 5) ConcreteJob provider responds to ConcreteJob.ResumeWithInput with 5 (Invalid Parameter).
- 937 6) ConcreteJob provider sets ConcreteJob.JobState= 10 (Exception).

938 **9.2.6 Client does not respond with enough valid values**

939 - Which CPU speeds in GHz to you want to test? [2.4]

- 940 1) Client calls DiagnosticTest.RunDiagnosticService() which returns 0 (Success) and the objectpath
941 of the ConcreteJob instance and starts the diagnostic test. ConcreteJob.JobState = 4 (Running)
942 and ConcreteJob.PercentComplete = 0.
- 943 2) ConcreteJob provider prompts client to respond with a DIAG34 message, which requests a value
944 for the CPUSpeed1 and CPUSpeed2 arguments.

- 945 3) Client calls ConcreteJob.ResumeWithInput() to respond with input argument Inputs[0]="2.4" for
 946 the CPUSpeed1 argument requested by DIAG34 message but does not provide a value for
 947 Inputs[1] for the CPUSpeed2 argument.
- 948 4) The ConcretJob provider detects that Inputs[1] has no value.
- 949 5) ConcreteJob provider responds to ConcreteJob.ResumeWithInput with 5 (Invalid Parameter).
- 950 6) ConcreteJob provider sets ConcreteJob.JobState= 10 (Exception).

951 9.3 User action required

952 For an interactive diagnostic test where user action is required, the ConcreteJob provider prompts client
 953 to respond with a DIAG35 message. The client responds by invoking the
 954 ConcreteJob.ResumeWithAction() extrinsic method.

955 9.3.1 Single prompt and response required

956 - *Press any key when the network cable has been attached. [Enter] (Test execution starts)*

- 957 1) Client calls DiagnosticTest.RunDiagnosticService() which returns 0 (Success) and the objectpath
 958 of the ConcreteJob instance and starts the diagnostic test. ConcreteJob.JobState = 4 (Running)
 959 and ConcreteJob.PercentComplete = 0.
- 960 2) ConcreteJob provider prompts client to respond with a DIAG35 message.
- 961 3) ConcreteJob provider sets ConcreteJob.JobState = 12 (Query Pending) to wait for client
 962 response.
- 963 4) Client calls ConcreteJob.ResumeWithAction() after attaching the cable.
- 964 5) ConcreteJob provider sets ConcreteJob.JobState = 4 (Running).
- 965 6) ConcreteJob.ResumeWithAction() returns 0 ("Completed with No Error") and test execution
 966 starts.
- 967 7) After the test completes successfully, the ConcreteJob provider sets ConcreteJob.JobState = 7
 968 (Completed), ConcreteJob.OperationalStatus= 17 (Completed), and ConcreteJob
 969 PercentComplete=100.

970 9.3.2 Multiple prompts and responses required before running the test

971 - *Press any key when the network cable has been attached. [Enter]*

972 - *Press any key after the CD has been inserted into the drive. [Enter] (Test execution starts)*

- 973 1) Client calls DiagnosticTest.RunDiagnosticService() which returns 0 (Success) and the objectpath
 974 of the ConcreteJob instance and starts the diagnostic test. ConcreteJob.JobState = 4 (Running)
 975 and ConcreteJob.PercentComplete = 0.
- 976 2) ConcreteJob provider prompts client to respond with a DIAG35 message.
- 977 3) ConcreteJob provider sets ConcreteJob.JobState = 12 (Query Pending) to wait for client
 978 response.
- 979 4) Client calls ConcreteJob.ResumeWithAction() after attaching the cable.
- 980 5) ConcreteJob.ResumeWithAction() returns 0 ("Completed with No Error").
- 981 6) ConcreteJob provider prompts client to respond with a DIAG35 message.
- 982 7) ConcreteJob provider sets ConcreteJob.JobState = 12 (Query Pending) to wait for client
 983 response.
- 984 8) Client calls ConcreteJob.ResumeWithAction() after inserting the CD into the drive.
- 985 9) ConcreteJob.ResumeWithAction() returns 0 ("Completed with No Error")

- 986 10) ConcreteJob provider sets ConcreteJob.JobState = 4 (Running) and test execution starts.
 987 11) After test completes successfully, ConcreteJob provider sets ConcreteJob.JobState = 7
 988 (Completed) and ConcreteJob.PercentComplete=100.

989 **9.3.3 Multiple prompts and responses required with partial test execution after each**

- 990 - Press any key when the network cable has been attached. [Enter]. (Test execution starts)
 991 - Press any key when the LED is on. [Enter] (Test execution resumes)
 992 - Press any key when the LED is off. [Enter] (Test execution resumes)
- 993 1) Client calls DiagnosticTest.RunDiagnosticService() which returns 0 (Success) and the objectpath
 994 of the ConcreteJob instance and starts the diagnostic test. ConcreteJob.JobState = 4 (Running)
 995 and ConcreteJob.PercentComplete = 0.
- 996 2) ConcreteJob provider prompts client to respond with a DIAG35 message.
- 997 3) ConcreteJob provider sets ConcreteJob.JobState = 12 (Query Pending) to wait for client
 998 response.
- 999 4) Client calls ConcreteJob.ResumeWithAction() after attaching the cable.
- 1000 5) ConcreteJob.ResumeWithAction() returns 0 ("Completed with No Error").
- 1001 6) ConcreteJob provider sets ConcreteJob.JobState = 4 (Running) and
 1002 ConcreteJob.PercentComplete = 25 and test execution resumes.
- 1003 7) ConcreteJob provider prompts client to respond with a DIAG35 message.
- 1004 8) ConcreteJob provider sets ConcreteJob.JobState = 12 (Query Pending) to wait for client
 1005 response.
- 1006 9) Client calls ConcreteJob.ResumeWithAction() after verifying the LED is ON.
- 1007 10) ConcreteJob.ResumeWithAction() returns 0 ("Completed with No Error")
- 1008 11) ConcreteJob provider sets ConcreteJob.JobState = 4 (Running) and
 1009 ConcreteJob.PercentComplete = 50 and test execution resumes.
- 1010 12) ConcreteJob provider prompts client to respond with a DIAG35 message.
- 1011 13) ConcreteJob provider sets ConcreteJob.JobState = 12 (Query Pending) to wait for client
 1012 response.
- 1013 14) Client calls ConcreteJob.ResumeWithAction() after verifying the LED is ON.
- 1014 15) ConcreteJob.ResumeWithAction() returns 0 ("Completed with No Error").
- 1015 16) ConcreteJob provider sets ConcreteJob.JobState = 4 (Running) and
 1016 ConcreteJob.PercentComplete = 75 and test execution resumes.
- 1017 17) After test completes successfully, ConcreteJob provider sets ConcreteJob.JobState = 7
 1018 (Completed) and ConcreteJob.PercentComplete=100.

1019 **9.3.4 Client does not respond to a prompt**

- 1020 - Press any key when the network cable has been attached. <timeout>
- 1021 1) Client calls DiagnosticTest.RunDiagnosticService() which returns 0 (Success) and the objectpath
 1022 of the ConcreteJob instance and starts the diagnostic test. ConcreteJob.JobState = 4 (Running)
 1023 and ConcreteJob.PercentComplete = 0.
- 1024 2) ConcreteJob provider prompts client to respond with a DIAG35 message.
- 1025 3) ConcreteJob provider sets ConcreteJob.JobState = 12 (Query Pending) to wait for client
 1026 response.

1027 4) When JobSettingData.InteractiveTimeout is exceeded, the ConcreteJob provider examines the
1028 value of JobSettingData.ClientRetriesMax.

1029 5) If the value of JobSettingData.ClientRetriesMax is zero, or if the number of retries has been
1030 exceeded, the ConcreteJob provider sets ConcreteJob.JobState = 10 (Exception) provided
1031 JobSettingData.TerminateOnTimeout=TRUE.

1032 6) If the value of JobSettingData.ClientRetriesMax is non-zero, steps 2, 3, 4, and 5 are repeated.

1033 NOTE If the value of JobSettingData.ClientRetriesMax is non-zero, the ConcreteJob provider monitors how many
1034 timeouts that have occurred.

1035 9.4 Silent mode operation

1036 An interactive test can be run as a non-interactive test; that is, the test does not prompt the client for
1037 responses. Instead, the ConcreteJob provider uses the default values from
1038 JobSettingData.DefaultInputNames and JobSettingData.DefaultInputValues to run the diagnostic test
1039 instead of sending a DIAG34 message to prompt the client.

1040 NOTE Silent mode works only if JobSettingData.DefaultInputNames and JobSettingData.DefaultInputValues have
1041 all the values needed to run the interactive test.

1042 An interactive test can only be run if the diagnostic test supports this capability; that is, if
1043 DiagnosticServiceJobCapabilities.SilentModeSupported = "true". To run in silent mode, the client sets
1044 JobSettingData.RunInSilentMode = "true" before invoking the DiagnosticTest.RunDiagnosticService()
1045 extrinsic method.

1046 9.5 Finding diagnostic jobs

1047 9.5.1 Finding all diagnostic jobs on a system

1048 A client can find all jobs for diagnostic tests on a System as follows. Assume that the client starts at a
1049 known System on which the diagnostic tests are run.

1050 1) From the System instance, the client calls the Associators operation using HostedDependency as
1051 the association class argument, and ConcreteJob is the result class argument, which returns all
1052 instances of ConcreteJob.

1053 2) For each ConcreteJob instance returned, the client calls the Associators operation using
1054 OwningJobElement as the association class argument, DiagnosticTest as the result class
1055 argument, and OwningElement as the result role argument. The operation returns the associated
1056 DiagnosticTest.

1057 9.5.2 Finding all diagnostic jobs for a ManagedElement

1058 A client can find all jobs for a ManagedElement as follows. Assume that the client starts at a known
1059 ManagedElement instance, which represents the device to be tested.

1060 1) From the ManagedElement instance, the client calls the Associators operation using
1061 AffectedJobElement as the association class argument, ConcreteJob as the result class
1062 argument, and AffectingElement as the result role argument.

1063 2) For each ConcreteJob instance returned, the client calls the Associators operation using
1064 OwningJobElement as the association class argument, DiagnosticTest as the result class
1065 argument, and OwningElement as the result role argument. The operation returns the associated
1066 DiagnosticTest.

1067 9.6 Configuring a diagnostic job

1068 To run a diagnostic test, the client invokes the `RunDiagnosticService()` extrinsic method of
 1069 `DiagnosticTest`. The `JobSetting` parameter may be an empty string, `NULL` or a string representing an
 1070 embedded instance of `JobSettingData`. When `JobSetting` is an empty string or `NULL`, then the job runs
 1071 using the default settings which may or may not have been published by the implementation.

1072 Note that the diagnostic default job settings are represented by a `JobSettingData` subclass that may have
 1073 extensions. If the client is aware of the extensions, they may be modified as well. If the client is unaware,
 1074 the default values should be used.

1075 9.6.1 Getting the default job settings

1076 The client can obtain the default job settings for a diagnostic service as follows. Assume that the client
 1077 starts at a known `DiagnosticTest` instance.

- 1078 1) From the `DiagnosticTest` instance, the client calls the `Associators` operation
 1079 using `ElementSettingData` as the association class argument, `JobSettingData` as the result
 1080 class argument, and `SettingData`. The operation returns `JobSettingData` instances.
- 1081 2) For each `JobSettingData` instance, the client calls the `References` operation using
 1082 `ElementSettingData` as the result class. The operation returns `ElementSettingData` instances.
- 1083 3) For each `ElementSettingData` instance, the client determines whether the value of the
 1084 `ElementSettingData.ManagedElement` property matches the `DiagnosticTest` name and the
 1085 value of the `ElementSettingData.IsDefault` property is 1 ("Is Default"). If so, the `JobSettingData`
 1086 instance represents the default job settings. The name of this `JobSettingData` instance may also
 1087 be retrieved from `ElementSettingData.SettingData` property.

1088 9.6.2 Creating the job settings

1089 A client may create their own job settings to pass as an argument to the
 1090 `DiagnosticTest.RunDiagnosticService()` extrinsic method as follows. Assume that the client starts at a
 1091 known `DiagnosticTest` instance.

- 1092 4) The client can attempt to discover the default job settings of the `DiagnosticTest` instance. The
 1093 `GetDefaultJobSettings` use case (subclause 9.6.1) describes the necessary steps.
- 1094 5) If the client wants to not use the default job settings, the client can attempt to find the associated
 1095 `DiagnosticServiceJobCapabilities` instance by calling the `Associators` operation using
 1096 `ElementCapabilities` as the association class argument, `DiagnosticServiceJobCapabilities` as
 1097 the result class, and `Capabilities` as the result role.
- 1098 6) If Step 2 returns an instance, the client calls the `CreateGoalSettings()` extrinsic method of the
 1099 returned `DiagnosticServiceJobCapabilities` instance. This operation returns an instance of
 1100 `JobSettingData` containing default values. The client can modify any property values as desired.
 1101 If a range of values is permitted for a property, the client should use only those values indicated
 1102 in the `DiagnosticServiceJobCapabilities` instance.
- 1103 7) If Step 2 does not return an instance because the implementation of
 1104 `DiagnosticServiceJobCapabilities` is optional, the client should use the default `JobSettingData`.

1105 9.7 Execute and control a job for a diagnostic test

1106 The `DiagnosticServiceJobCapabilities.RequestedStatesSupported` property indicates the permitted
 1107 values of the `RequestedState` input parameter for the `ConcreteJob.RequestStateChange()` extrinsic
 1108 method. Since `DiagnosticServiceJobCapabilities` is an optional class, a client may not be able to examine
 1109 an instance to determine which values of `RequestedState` to use. If a client invokes
 1110 `ConcreteJob.RequestStateChange()` to change to an unsupported state, the extrinsic method shall
 1111 return 4097 ("Invalid State Transition").

1112 9.7.1 Suspend a job for a diagnostic test

1113 The client can suspend the execution of a test by invoking the `ConcreteJob.RequestStateChange()`
1114 extrinsic method on the `ConcreteJob` instance that is returned from the
1115 `DiagnosticTest.RunDiagnosticService()` extrinsic method. Assume that the client starts at a known
1116 `ConcreteJob` instance and that a `DiagnosticServiceJobCapabilities` instance exists.

- 1117 1) From the `ConcreteJob` instance, the client calls the `Associators` operation using
1118 `OwningJobElement` as the association class argument, `DiagnosticTest` as the result class
1119 argument, and `OwningElement` as the result role. The associated `DiagnosticTest` instance is
1120 returned.
- 1121 2) From the `ConcreteJob` instance, the client calls the `Associators` operation using
1122 `ElementCapabilities` as the association class argument, `DiagnosticServiceJobCapabilities` as
1123 the result class argument, and `Capabilities` as the result role. The associated
1124 `DiagnosticServiceJobCapabilities` instance is returned.
- 1125 3) The client examines the `DiagnosticServiceJobCapabilities.RequestedStatesSupported` property.
1126 If it contains the value of 3 ("Suspend"), the `ConcreteJob` can be suspended.
- 1127 4) The client invokes the `ConcreteJob.RequestStateChange()` extrinsic method where input
1128 parameter `RequestedState` has the value of 3 ("Suspend").
- 1129 5) When the transition completes successfully, the `ConcreteJob.JobState` property shall have the
1130 value of 5 ("Suspended") and `ConcreteJob.TimeOfLastStateChange` property shall be set to the
1131 current time.

1132 9.7.2 Resume a job for a diagnostic

1133 The client can resume the execution of a test that was previously suspended by invoking the
1134 `ConcreteJob.RequestStateChange()` extrinsic method on the `ConcreteJob` instance that is returned from
1135 the `DiagnosticTest.RunDiagnosticService()` extrinsic method. Assume that the client starts at a known
1136 `ConcreteJob` instance and that a `DiagnosticServiceJobCapabilities` instance exists.

- 1137 1) From the `ConcreteJob` instance, the client calls the `Associators` operation using
1138 `OwningJobElement` as the association class argument, `DiagnosticTest` as the result class
1139 argument, and `OwningElement` as the result role. The associated `DiagnosticTest` instance is
1140 returned.
- 1141 2) From the `ConcreteJob` instance, the client calls the `Associators` operation using
1142 `ElementCapabilities` as the association class argument, `DiagnosticServiceJobCapabilities` as the
1143 result class argument, and `Capabilities` as the result role. The associated
1144 `DiagnosticServiceJobCapabilities` instance is returned.
- 1145 3) The client examines the `DiagnosticServiceJobCapabilities.RequestedStatesSupported` property.
1146 If it contains the value of 2 ("Start"), the `ConcreteJob` can be resumed.
- 1147 4) The client invokes the `ConcreteJob.RequestStateChange()` extrinsic method where input
1148 parameter `RequestedState` has the value of 2 ("Start").
- 1149 5) When the transition completes successfully, the `ConcreteJob.JobState` property shall have the
1150 value of property to 4 ("Running") and `ConcreteJob.TimeOfLastStateChange` property shall be set
1151 to the current time.

1152 NOTE The `JobState` property may transition from the value 3 ("Starting") before the final transition to the value of 4
1153 ("Running").

1154 9.7.3 Terminate a job for a diagnostic

1155 The client can cleanly terminate the execution of a test by invoking the
 1156 `ConcreteJob.RequestStateChange()` extrinsic method on the `ConcreteJob` instance that is returned from
 1157 the `DiagnosticTest.RunDiagnosticService()` extrinsic method. Assume that the client starts at a known
 1158 `ConcreteJob` instance and that a `DiagnosticServiceJobCapabilities` instance exists.

- 1159 1) From the `ConcreteJob` instance, the client calls the `Associators` operation using
 1160 `OwningJobElement` as the association class argument, `DiagnosticTest` as the result class
 1161 argument, and `OwningElement` as the result role. The associated `DiagnosticTest` instance is
 1162 returned.
- 1163 2) From the `ConcreteJob` instance, the client calls the `Associators` operation using
 1164 `ElementCapabilities` as the association class argument, `DiagnosticServiceJobCapabilities` as the
 1165 result class argument, and `Capabilities` as the result role. The associated
 1166 `DiagnosticServiceJobCapabilities` instance is returned.
- 1167 3) The client examines the `DiagnosticServiceJobCapabilities`. `RequestedStatesSupported` property.
 1168 If it contains the value of 4 (“Terminate”), the `ConcreteJob` can be terminated.
- 1169 4) The client invokes the `ConcreteJob.RequestStateChange()` extrinsic method where input
 1170 parameter `RequestedState` has the value of 4 (“Terminate”).
- 1171 5) When the transition completes successfully, the `ConcreteJob.JobState` property shall have the
 1172 value of property to 8 (“Terminated”) and `ConcreteJob.TimeOfLastStateChange` property shall be
 1173 set to the current time.

1174 NOTE The `JobState` property may transition to 7 (“Shutting Down”) before the final transition to 8 (“Terminated”).

1175 9.7.4 Kill a job for a diagnostic

1176 The client can immediately abort the execution of a test, with no attempt to perform a clean shutdown, by
 1177 invoking the `ConcreteJob.RequestStateChange()` extrinsic method on the `ConcreteJob` instance that is
 1178 returned from the `DiagnosticTest.RunDiagnosticService()` extrinsic method. Assume that the client
 1179 starts at a known `ConcreteJob` instance and that a `DiagnosticServiceJobCapabilities` instance exists.

- 1180 1) From the `ConcreteJob` instance, the client calls the `Associators` operation using
 1181 `OwningJobElement` as the association class argument, `DiagnosticTest` as the result class
 1182 argument, and `OwningElement` as the result role. The associated `DiagnosticTest` instance is
 1183 returned.
- 1184 2) From the `ConcreteJob` instance, the client calls the `Associators` operation using
 1185 `ElementCapabilities` as the association class argument, `DiagnosticServiceJobCapabilities` as the
 1186 result class argument, and `Capabilities` as the result role. The associated
 1187 `DiagnosticServiceJobCapabilities` instance is returned.
- 1188 3) The client examines the `DiagnosticServiceJobCapabilities`. `RequestedStatesSupported` property.
 1189 If it contains the value of 5 (“Kill”), the `ConcreteJob` can be aborted.
- 1190 4) The client invokes the `ConcreteJob.RequestStateChange()` extrinsic method where input
 1191 parameter `RequestedState` has the value of 5 (“Kill”).
- 1192 5) When the transition completes successfully, the `ConcreteJob.JobState` property shall have the
 1193 value of property to 9 (“Killed”) and `ConcreteJob.TimeOfLastStateChange` property shall be set to
 1194 the current time.

1195 9.8 Delete a job

1196 When the optional DiagnosticServiceJobCapabilities instance exists and its DeleteJobSupported property
1197 has the value of TRUE, a client can control how and when the ConcreteJob instance associated to a
1198 specific diagnostic test execution is deleted. Assume that the client has verified that
1199 DiagnosticServiceJobCapabilities.DeleteJobSupported has the value of TRUE. Also assume that the
1200 client starts at a known DiagnosticTest instance. See Table 4 for further information.

1201 NOTE A client may remove a job by invoking the ConcreteJob.RequestStateChange extrinsic method with input
1202 parameter having a value of 5 ("Kill") or 4 ("Terminate"). Also, a client may perform the DeleteInstance operation on
1203 the ConcreteJob instance when JobSettingData.DeleteOnCompletion property has the value of FALSE.

1204 9.8.1 Client deletes a job

1205 To configure the ConcreteJob instance to be deleted by the client rather than by the provider

- 1206 1) The client creates an embedded instance of JobSettingData where its DeleteOnCompletion
1207 property has the value of FALSE.
- 1208 2) The client invokes the DiagnosticTest.RunDiagnosticService() extrinsic method where the
1209 JobSetting input parameter has the value of the embedded instance of JobSettingData created in
1210 the previous step.
- 1211 3) After the diagnostic test completes or otherwise terminates, the ConcreteJob instance shall
1212 remain indefinitely until the client performs the DeleteInstance operation on it.

1213 NOTE Assume that DiagnosticServiceJobCapabilities.CleanupInterval is NULL. Otherwise, the provider deletes the
1214 ConcreteJob instance..

1215 9.8.2 Provider deletes a job

1216 To configure the ConcreteJob instance to be deleted by the provider rather than by the client

- 1217 1) The client creates an embedded instance of JobSettingData where its DeleteOnCompletion
1218 property has the value of TRUE.
- 1219 2) The client may set the TimeBeforeRemoval property of the embedded JobSettingData instance. If
1220 set, the ConcreteJob instance is deleted based on the value of TimeBeforeRemoval after the job
1221 completes.
- 1222 3) The client invokes the DiagnosticTest.RunDiagnosticService() extrinsic method where the
1223 JobSetting input parameter has the value of the embedded instance of JobSettingData created in
1224 the previous steps.
- 1225 4) The provider removes the ConcreteJob instance at the time indicated by the value of the
1226 TimeBeforeRemoval property.

1227 EXPERIMENTAL

1228 **10 CIM elements**

1229 Table 15 shows the instances of CIM elements for this profile. Instances of the CIM elements shall be
 1230 implemented as described in Table 15. Clause 7 (“Implementation”) and Clause 8 (“Methods”) may
 1231 impose additional requirements on these elements. See [DSP1103](#) for other mandatory elements that
 1232 must be implemented.

1233 **Table 15 – CIM Elements: Diagnostic Job Control Profile**

| Element Name | Requirement | Description |
|---|-------------|--|
| Classes | | |
| CIM_AffectedJobElement | Optional | See 10.1. |
| CIM_ConcreteJob | Mandatory | See 10.2. |
| CIM_DiagnosticServiceJobCapabilities | Optional | See 10.3. |
| CIM_ElementCapabilities (Job) | Optional | See 10.4. |
| CIM_ElementSettingData (Default Job) | Optional | See 10.5. |
| CIM_HostedDependency | Mandatory | See 10.6. |
| CIM_JobSettingData (Client) | Optional | See 10.7. |
| CIM_JobSettingData (Default) | Mandatory | See 10.8. |
| CIM_OwningJobElement | Mandatory | See 10.9. |
| CIM_RegisteredProfile | Mandatory | See 10.10. |
| Indications | | |
| SELECT * FROM CIM_AlertIndication WHERE OwningEntity="DMTF" and MessageID="DIAG9" | Conditional | This indication is mandatory if CIM_DiagnosticTest.Characteristics has the value of 3 (Is Interactive). Query Language:DMTF:CQL The test continued execution using a default response because a query timeout occurred. |
| SELECT * FROM CIM_AlertIndication WHERE OwningEntity="DMTF" and MessageID="DIAG12" | Optional | Query Language: DMTF:CQL The test did not run because the job could not be started. |
| SELECT * FROM CIM_AlertIndication WHERE OwningEntity="DMTF" and MessageID="DIAG19" | Mandatory | Query Language: DMTF:CQL The test was killed by the client |
| SELECT * FROM CIM_AlertIndication WHERE OwningEntity="DMTF" and MessageID="DIAG20" | Mandatory | Query Language: DMTF:CQL The test was terminated by the client |
| SELECT * FROM CIM_AlertIndication WHERE OwningEntity="DMTF" and MessageID="DIAG21" | Optional | Query Language: DMTF:CQL The test was suspended by the client |

| Element Name | Requirement | Description |
|--|-------------|--|
| SELECT * FROM CIM_AlertIndication WHERE OwningEntity="DMTF" and MessageID="DIAG34" | Optional | Query Language: DMTF:CQL The test solicits input from a client. |
| SELECT * FROM CIM_AlertIndication WHERE OwningEntity="DMTF" and MessageID="DIAG35" | Optional | Query Language: DMTF:CQL The test solicits an action from a user. |
| SELECT * FROM CIM_AlertIndication WHERE OwningEntity="DMTF" and MessageID="DIAG36" | Optional | Query Language: DMTF:CQL The test was killed by the test (provider). |
| SELECT * FROM CIM_AlertIndication WHERE OwningEntity="DMTF" and MessageID="DIAG37" | Optional | Query Language: DMTF:CQL The test was terminated by the test (provider). |
| SELECT * FROM CIM_AlertIndication WHERE OwningEntity="DMTF" and MessageID="DIAG38" | Optional | Query Language: DMTF:CQL The test was resumed by the client |
| SELECT * FROM CIM_AlertIndication WHERE OwningEntity="DMTF" and MessageID="DIAG39" | Conditional | This indication is mandatory if CIM_DiagnosticTest.Characteristics has the value of 3 (Is Interactive). Query Language:DMTF:CQL A JobSetting parameter was reset by the job. |
| SELECT * FROM CIM_AlertIndication WHERE OwningEntity="DMTF" and MessageID="DIAG40" | Conditional | This indication is mandatory if CIM_DiagnosticTest.Characteristics has the value of 3 (Is Interactive). Query Language:DMTF:CQL JobSetting parameter was ignored by the job. |

1234 10.1 CIM_AffectedJobElement

1235 Although defined in [DSP1103](#), the CIM_AffectedJobElement class is listed here because the
1236 AffectedElement reference is scoped down to CIM_DiagnosticTest. The constraints listed in Table 16 are
1237 in addition to those specified in [DSP1103](#). See [DSP1103](#) for other mandatory elements that must be
1238 implemented.

1239

Table 16 – Class: CIM_AffectedJobElement

| Properties | Requirement | Notes |
|---------------------------------|-------------|--|
| AffectedElement (overridden) | Mandatory | Key. The property shall be a reference to an instance of the CIM_ManagedElement being tested.. |
| AffectingElement | Mandatory | Key. The property shall be a reference to an instance of CIM_ConcreteJob. |
| ElementEffects | Optional | See 7.1.1. |
| OtherElementEffectsDescriptions | Optional | If ElementEffects contains the value 1, this property is Mandatory. |

1240 **10.2 CIM_ConcreteJob**

1241 Each successful invocation of the CIM_DiagnosticTest.RunDiagnosticService() extrinsic method returns
 1242 a CIM_ConcreteJob instance. Each CIM_ConcreteJob instance represents a diagnostic test execution.
 1243 This class specializes CIM_ConcreteJob as defined in [DSP1103](#). The constraints listed in Table 17 are in
 1244 addition to those specified in [DSP1103](#). See [DSP1103](#) for other mandatory elements that must be
 1245 implemented.

1246

Table 17 – Class: CIM_ConcreteJob

| Properties | Requirement | Notes |
|-----------------------|-------------|-----------------|
| InstanceID | Mandatory | Key. See 7.2.1. |
| Name | Mandatory | See 7.2.2. |
| JobState | Mandatory | See 7.2.3. |
| DeleteOnCompletion | Mandatory | See 7.2.4. |
| TimeBeforeRemoval | Mandatory | See 7.2.5. |
| StartTime | Mandatory | See 7.2.6. |
| ElapsedTime | Mandatory | See 7.2.7. |
| RequestedState | Mandatory | |
| PercentComplete | Mandatory | See 7.2.8. |
| ElementName | Mandatory | See 7.2.9. |
| RequestStateChange() | Mandatory | See 8.2.1. |
| ResumeWithInput() | Conditional | See 8.2.2. |
| ResumeWithAction() | Conditional | See 8.2.3. |

1247 **10.3 CIM_DiagnosticServiceJobCapabilities**

1248 The CIM_DiagnosticServiceJobCapabilities is used to provide information about on the capabilities of the
 1249 job that is used to run the diagnostic test.

1250

Table 18 – Class: CIM_DiagnosticServiceJobCapabilities

| Properties | Requirement | Notes |
|--------------------------|-------------|-----------------|
| InstanceID | Mandatory | Key. See 7.3.1. |
| ElementName | Mandatory | See 7.3.2. |
| DeleteJobSupported | Mandatory | See 7.3.3. |
| RequestedStatesSupported | Mandatory | See 7.3.4. |
| InteractiveTimeoutMax | Conditional | See 7.3.5. |
| DefaultValuesSupported | Conditional | See 7.3.6. |
| ClientRetriesMax | Conditional | See 7.3.7. |
| CleanupInterval | Optional | See 7.3.8. |
| SilentModeSupported | Conditional | See 7.3.9. |
| CreateGoalSettings() | Mandatory | See 8.3.1. |

1251 **10.4 CIM_ElementCapabilities (Job)**

1252 CIM_ElementCapabilities represents an association between a test and its capabilities to use a job to run
1253 the test. Table 19 contains the requirements for elements of this class.

1254

Table 19 – Class: CIM_ElementCapabilities

| Properties | Requirement | Notes |
|----------------|-------------|--|
| ManagedElement | Mandatory | Key. The property shall be a reference to an instance of CIM_DiagnosticTest. |
| Capabilities | Mandatory | Key. The property shall be a reference to an instance of CIM_DiagnosticServiceJobCapabilities. |

1255 **10.5 CIM_ElementSettingData (Default JobSettingData)**

1256 CIM_ElementSettingData represents an association between a test and the default setting data for the
1257 job used to run the test. ManagedElement is responsible for the creation of the job. Table 20 contains the
1258 requirements for elements of this class. If CIM_DiagnosticTest (or a subclass) has more than one type of
1259 test, a separate default job setting may be defined for each. However, all instances of the same type of
1260 test shall reference the same default job setting.

1261

Table 20 – Class: CIM_ElementSettingData

| Properties | Requirement | Notes |
|----------------|-------------|--|
| ManagedElement | Mandatory | Key. The property shall be a reference to an instance of CIM_DiagnosticTest. |
| SettingData | Mandatory | Key. The property shall be a reference to an instance of CIM_JobSettingData (Default). |
| IsDefault | Mandatory | The value of the property shall be TRUE. |

1262 **10.6 CIM_HostedDependency**

1263 CIM_HostedDependency represents an association between the system on which a test is run and the
 1264 CIM_ConcreteJob that is used to run the test. Table 21 contains the requirements for elements of this
 1265 class.

1266 **Table 21 – Class: CIM_HostedDependency**

| Properties | Requirement | Notes |
|------------|-------------|---|
| Antecedent | Mandatory | Key. The property shall be a reference to an instance of CIM_System. |
| Dependent | Mandatory | Key. The property shall be a reference to an instance of CIM_ConcreteJob. |

1267 **10.7 CIM_JobSettingData (Client)**

1268 This CIM_JobSettingData definition represents the settings the client uses for the job used to run the
 1269 diagnostic test. Table 22 contains the requirements for elements of this class.

1270 **Table 22 – Class: CIM_JobSettingData (Client)**

| Properties | Requirement | Notes |
|---------------------|-------------|--|
| InstanceID | Mandatory | Key. See 7.4.1. |
| DeleteOnCompletion | Conditional | See 7.4.2. |
| InteractiveTimeout | Conditional | See 7.4.3. |
| TerminateOnTimeout | Conditional | See 7.4.4. |
| DefaultInputsValues | Conditional | See 7.4.5. |
| DefaultInputNames | Conditional | See 7.4.6. |
| ClientRetries | Conditional | See 7.4.7. |
| RunInSilentMode | Conditional | See 7.4.8. |
| ElementName | Mandatory | This property is a free-form string of variable length. (pattern ".*") |

1271 **10.8 CIM_JobSettingData (Default)**

1272 This CIM_JobSettingData definition represents the default settings for the job used to run the diagnostic
 1273 test. Table 23 contains the requirements for elements of this class. If CIM_DiagnosticTest (or a subclass)
 1274 has more than one type of test, a separate default job setting may be defined for each.

1275 **Table 23 – Class: CIM_JobSettingData (Default)**

| Properties | Requirement | Notes |
|--------------------|-------------|-----------------|
| InstanceID | Mandatory | Key. See 7.5.1. |
| DeleteOnCompletion | Conditional | See 7.5.2. |
| InteractiveTimeout | Conditional | See 7.5.3. |
| TerminateOnTimeout | Conditional | See 7.5.4. |
| DefaultInputValues | Conditional | See 7.5.5. |

| Properties | Requirement | Notes |
|-------------------|-------------|--|
| DefaultInputNames | Conditional | See 7.5.6. |
| ClientRetries | Conditional | See 7.5.7. |
| RunInSilentMode | Conditional | See 7.5.8. |
| ElementName | Mandatory | This property is a free-form string of variable length. (pattern ".*") |

1276 10.9 CIM_OwningJobElement

1277 Although defined in [DSP1103](#), the CIM_OwningJobElement class is listed here because the
 1278 OwningElement reference is scoped down to CIM_DiagnosticTest, which is a subclass of
 1279 CIM_ManagedElement. The constraints listed in Table 24 are in addition to those specified in [DSP1103](#).
 1280 See [DSP1103](#) for other mandatory properties of CIM_HostedService that must be implemented.

1281 **Table 24 – Class: CIM_OwningJobElement**

| Properties | Requirement | Notes |
|---------------|-------------|--|
| OwningElement | Mandatory | Key. The property shall be a reference to an instance of CIM_DiagnosticTest. |
| OwnedElement | Mandatory | Key. The property shall be a reference to an instance of CIM_ConcreteJob. |

1282 10.10 CIM_RegisteredProfile

1283 The CIM_RegisteredProfile class is defined in [DSP1033](#). The constraints listed in Table 25 are in addition
 1284 to those specified in [DSP1033](#). See [DSP1033](#) for other mandatory properties of CIM_RegisteredProfile
 1285 that must be implemented.

1286 **Table 25 – Class: CIM_RegisteredProfile**

| Properties | Requirement | Notes |
|------------------------|-------------|---|
| RegisteredName | Mandatory | The value of the property shall be "Diagnostic Job Control" |
| RegisteredVersion | Mandatory | The value of the property shall be "1.0.0" |
| RegisteredOrganization | Mandatory | The value of the property shall be 2 (DMTF) |

1287
1288
1289
1290

Annex A (informative)

Change log

| Version | Date | Description |
|---------|------------|--------------------------|
| 0.1 | 2010-12-05 | Initial Version |
| 1.0.0a | 2012-09-12 | Work In Progress version |

1291